



UNIVERSIDADE ESTADUAL DO NORTE DO PARANÁ

CAMPUS LUIZ MENEGHEL



JONAS PÉGOLI COLLELA

**ESPECIFICAÇÃO DE UM NÚCLEO DE ARTEFATOS
PARA LINHA DE PRODUTO DE SOFTWARE**

Bandeirantes

2010

JONAS PÉGOLI COLLELA

**ESPECIFICAÇÃO DE UM NÚCLEO DE
ARTEFATOS PARA LINHA DE PRODUTO DE
SOFTWARE**

Trabalho de Conclusão de Curso apresentado ao Curso de Sistemas de Informação da Universidade Estadual do Norte do Paraná como pré-requisito para o desenvolvimento do TCC, orientado pela Prof.^a Me. Cristiane Yanase Hirabara de Castro.

Bandeirantes

2010

JONAS PÉGOLI COLLELA

**ESPECIFICAÇÃO DE UM NÚCLEO DE
ARTEFATOS PARA LINHA DE PRODUTO DE
SOFTWARE**

Trabalho de Conclusão de Curso apresentado ao Curso de Sistemas de Informação da Universidade Estadual do Norte do Paraná - campus Luiz Meneghel, como requisito para obtenção do grau de Bacharel em Sistemas de Informação, orientado pela Prof.^a Me. Cristiane Yanase Hirabara de Castro

COMISSÃO EXAMINADORA

Prof.^a Me. Cristiane Yanase Hirabara de Castro.

Universidade Estadual do Norte do Paraná – Campus Luiz Meneghel

Prof.^o Me. Bruno Miguel Nogueira de Souza.

Universidade Estadual do Norte do Paraná – Campus Luiz Meneghel

Prof.^o Me. André Luis Andrade Menolli.

Universidade Estadual do Norte do Paraná – Campus Luiz Meneghel

Bandeirantes, __ de _____ de 2010.

DEDICATÓRIA

Dedico este trabalho as pessoas mais importantes e especiais da minha vida, meu pai Silvio Roberto Collela, minha mãe Sandra Silvia Pégoli Collela, meu irmão Tiago Pégoli Collela, meu irmão Lucas Pégoli Collela, minha namorada Fernanda Geralda Panchoni Vilas Boas e todas as pessoas que estiveram ao meu lado, que sempre me apoiaram, me incentivaram no desenvolvimento de todo o processo e lutaram para me manter longe de casa e continuar meus estudos.

AGRADECIMENTOS

Agradeço à minha mãe, **Sandra Silvia Pégoli Collela**, fonte de toda a minha inspiração, ao meu pai, **Silvio Roberto Collela**, pelo amor e paciência demonstrados, meus irmãos, **Tiago Pégoli Collela** e **Lucas Pégoli Collela**, pelo amor, carinho e constante incentivo, e a minha namorada, **Fernanda G. Vilas Boas**, pelo carinho, apoio e incentivo durante o desenvolvimento do meu trabalho.

Agradeço à minha orientadora **Cristiane Yanase Hirabara de Castro**, que depositou sua confiança em meu trabalho, pelos seus conselhos profissionais e pelo conhecimento compartilhado comigo.

Agradeço aos amigos “irmãos” da República Califórnia **Jaime Villa Jr.**, **Thiago R. Kichiro**, **Ricardo Cecílio**, **Milton Barbosa**, **Vitor de Rezende**, **Sergio de Oliveira**, **Ayrton Pappi** e **Túlio Marques** pela amizade, respeito e contribuições.

Agradeço aos amigos **Felipe Tic**, **Guilherme Gonzaga**, **Pedro Viola**, **Zé Borelli** e todos que conviveram comigo em Bandeirantes.

Agradeço aos meus familiares, **Rodrigo Pégoli**, **Amarildo Collela**, **Silvia Pizza**, **Marcia Collela**, **Iracy Nogueira**, **Neiva Pégoli** pelo apoio e confiança depositada em meu sucesso.

Um agradecimento especial aos meus avôs **Ilidio Collela** e **Germinal Pégoli**, que apesar de não participarem mais fisicamente das minhas conquistas aqui, sei que sempre torceram e continuam torcendo pelo meu sucesso lá do céu.

*“Nem tudo que se enfrenta
pode ser modificado mas nada pode
ser modificado até que seja
enfrentado”.*
(Albert Einstein)

RESUMO

Cada vez mais as empresas dependem de produção em maior escala em um menor tempo de desenvolvimento para realização de seus negócios, portanto, torna-se fundamental garantir a qualidade do produto desenvolvido. Um dos principais fatores de sucesso no desenvolvimento de Software é a reutilização. No qual pode ser utilizada no desenvolvimento para melhorar o processo de desenvolvimento e também otimizar este processo, esta abordagem de Linha de Produto de Software tem o seu foco direcionado ao menor tempo de produção, maior escala de produção e menor custo. Este trabalho analisa os impactos na configuração de um produto de software, mais especificamente, durante o processo de desenvolvimento é analisado os impactos nas atualizações ocorridas em suas documentações. Durante as configurações efetuadas no núcleo de artefatos pode se perceber contribuições e maturidade do núcleo de artefatos. Na configuração do novo produto o gerenciamento de variabilidades auxilia na identificação dos pontos os quais um produto se diferencia em uma Linha de Produto de Software.

Palavras-chave: Linha de Produto de Software, Reutilização, Variabilidades.

ABSTRACT

Increasingly, companies rely on larger-scale production in a shorter development time to conduct their business, therefore, becomes crucial to ensure the quality of the product developed. One of the key success factors in software development is reuse. Which can be used in development to improve the development process and to optimize this process, this approach of Software Product Line has its focus directed to the minimum time of production, larger scale production and lower costs. This paper analyzes the impacts on the configuration of a software product, more specifically, during the development process is analyzed the impact on updates that have occurred in their documentation. During the settings made in the core of artifacts can be perceived contributions and maturity core artifacts. In the configuration of the new product management variability helps to identify the points which a product is differentiated in a Software Product Line.

Keywords: Software Product Line, Reuse, variability.

LISTA DE FIGURAS

FIGURA 1 ATIVIDADES ESSENCIAIS DA LINHA DE PRODUTO DE SOFTWARE.....	20
FIGURA 2: DESENVOLVIMENTO DO NÚCLEO DE ARTEFATOS	21
FIGURA 3: DESENVOLVIMENTO DO PRODUTO	22
FIGURA 4: ATIVIDADES PARA O PROCESSO DE GERENCIAMENTO DE VARIABILIDADE PARA LP.	24
FIGURA 5: MODELO DE CARACTERÍSTICAS DE UM PROCESSO DE COMPRA VIA INTERNET.....	29
FIGURA 6: MODELO DE CASOS DE USO DE UM PROCESSO DE COMPRA VIA INTERNET. 33	
FIGURA 7: DIAGRAMA DE CASO DE USO DE UM MICROONDAS EM LPS.	34
FIGURA 8: ETAPAS PARA MODELAR O DIAGRAMA DE CLASSES.	36
FIGURA 9: REPRESENTAÇÃO DE VARIABILIDADES EM DIAGRAMA DE CLASSES.....	37
FIGURA 10: DIAGRAMA DE CLASSES DO SOLIGEE.....	43
FIGURA 11: DIAGRAMA DE CLASSES DO CTEC.	44
FIGURA 12: CAMADAS VIEWER E CONTROLLER - DIAGRAMA DE CLASSES SOLIGEE. 45	
FIGURA 13: CAMADAS VIEWER E CONTROLLER - DIAGRAMA DE CLASSES CTEC.....	46
FIGURA 14: CAMADA MODEL - DIAGRAMA DO SOLIGEE.	47
FIGURA 15: CAMADA MODEL - PACOTE CENTRO TECNOLÓGICO DE CARNES.....	48
FIGURA 16: MODELO DE CARACTERÍSTICAS DO SOLIGEE.	50
FIGURA 17: MODELO DE CARACTERÍSTICAS DO CTEC.....	51
FIGURA 18: TELA DE MENU DO SOFTWARE CETEC.....	53
FIGURA 19: TELA DE CADASTRO DE ANIMAL CORTES BRUTOS.....	54
FIGURA 20: POP-UP DE CONFIRMAÇÃO DE CADASTRO.....	55
FIGURA 21: BOTÃO VOLTAR E POP-UP DE AVISO AO USUÁRIO.	56
FIGURA 22: TELA DE CADASTRO DE CORTES E DERIVADOS.	57
FIGURA 23: GRÁFICO DE VARIABILIDADES – OPCIONAIS.	58
FIGURA 24: GRÁFICO DE VARIABILIDADES – ALTERNATIVO OU.	58

LISTA DE SIGLAS

ACIAB	Associação Comercial, Industrial e Agrícola de Bandeirantes
CLM	Campus Luiz Meneghel
CTEC	Centro Tecnológico de Carnes
FODA	Feature Oriented Domain Analysis
FAST	Family-oriented Abstraction, Specification and Translation
HV	Hospital Veterinário
INES	Instituto Nacional de Ciência e Tecnologia para Engenharia de Software
LCD	Liquid Cristal Display
LPS	Linha de Produto de Software
LP	Linha de Produto
PLP	Product Line Practice
SOLIGEE	Software Livre de Gestão Empresarial
SEI	Software Engineering Institute
UENP	Universidade Estadual do Norte do Paraná
UML	Unified Modeling Language

SUMÁRIO

1	INTRODUÇÃO	12
1.1	OBJETIVO GERAL.....	13
1.2	OBJETIVO ESPECÍFICO	13
1.3	MOTIVAÇÃO.....	13
1.4	ORGANIZAÇÃO.....	14
2	LINHA DE PRODUTO DE SOFTWARE	15
2.1	CONCEITOS E DEFINIÇÕES.....	18
2.2	ABORDAGENS DE LINHA DE PRODUTO DE SOFTWARE	24
3	VARIABILIDADES E O MODELO DE CARACTERÍSTICAS	27
3.1	CONCEITOS E DEFINIÇÕES.....	27
3.2	CONCEITOS DA NOTAÇÃO.....	30
3.2.1	Notação para o Modelo de Características	30
3.2.2	Notação para o Diagrama de Casos de Uso	31
3.2.3	Notação para o Diagrama de Classes	34
4	DESENVOLVIMENTO	38
4.1	MÉTODOS	38
4.2	SOFTWARE LIVRE PARA GESTÃO EMPRESARIAL	38
4.3	CONFIGURAÇÃO DE UM SISTEMA ESPECÍFICO	40
4.4	DIAGRAMAS DE CLASSES.....	41
4.5	MODELOS DE CARACTERÍSTICAS.....	49
5	RESULTADOS.....	52
5.1	TELAS.....	52
5.2	GRÁFICOS.....	57
6	CONCLUSÃO	60
	REFERÊNCIAS BIBLIOGRÁFICAS	61

1 INTRODUÇÃO

As empresas e organizações na área de desenvolvimento de software, impulsionadas pela concorrência, procuram diversas formas para obter êxito no desenvolvimento e em seus produtos. Sendo assim e percebendo que entre softwares diferentes existem características comuns para um determinado domínio de desenvolvimento, observou-se a possibilidade de tornar o processo de desenvolvimento das empresas mais organizado e ágil.

A linha de produto de software (LPS) é um conjunto de sistemas de softwares, que compartilham entre eles as características comuns, visando atender um segmento de mercado ou missão.

Esta abordagem proporciona uma maneira sistemática, planejada e prática de reutilização de software. O objetivo é obter um processo de instanciação de produtos no qual não esteja envolvendo possibilidade a erros e que seja rápida, a instanciação de produtos deve ser ágil, podendo também utilizar geradores de aplicação, no qual podem gerar os artefatos utilizando uma especificação das variantes que existem em cada produto.

O crescimento da utilização da abordagem LPS foi benéfico para empresas que objetivam o trabalho direcionado a reutilização de outros produtos, pois é uma abordagem que faz com que a empresa esteja em busca de um maior aproveitamento do que já foi desenvolvido anteriormente.

A utilização deste recurso faz com que a empresa seja favorecida na redução de custo e tempo de produção, aumentando assim a sua produção. Porém, muito desse sucesso deve-se ao fato de a empresa enxergar no enfoque de linhas de produtos uma maneira de resolver problemas. A presente pesquisa foi realizada com intuito de analisar esta abordagem em diferentes áreas de atuação e verificar o seu impacto em cada uma delas.

O interesse das empresas ao adotar essa abordagem de LPS esta relacionada diretamente ao menor custo para o desenvolvimento de seus produtos e ao menor tempo para entrega desses produtos finalizados, pois o produto deve chegar ao mercado em menor tempo (*Time-to-Market*), é uma das vantagens da abordagem.

O que se percebe na utilização da abordagem de LPS é a dificuldade

no processo de gerenciamento de variabilidades, ou seja, dificuldade para identificar, representar, implementar e gerenciar as variabilidades durante o ciclo de vida de uma LP.

Para colocar em prática os conceitos da abordagem de LPS vai ser necessário estudar o Software Livre de Gestão Empresarial (SOLIGEE), gerenciar as variabilidades e entender suas características para que então possa colocar em desenvolvimento a documentação de um novo produto que é o desenvolvimento do software para o Centro Tecnológico de Carnes (CTEC).

1.1 OBJETIVO GERAL

O principal objetivo deste projeto é determinar os impactos verificados na configuração de um produto específico, em uma linha de produto de software.

1.2 OBJETIVO ESPECÍFICO

Os objetivos específicos deste trabalho, previamente definidos, foram:

1. Estudar conceitos necessários da abordagem de LPS e trabalhos recentes na área.
2. Buscar novas perspectivas sobre a abordagem de linha de produto de software e do modelo de característica com o propósito de trazer a tona os pontos mais fortes sobre a evolução do modelo para o diagrama de classes.
3. Após realizar os estudos, realizar uma análise do Modelo de Característica do SOLIGEE e do CTEC para identificar suas similaridades e suas especificidades.
4. Configurar o produto de acordo com os artefatos reutilizados e com os novos artefatos.
5. Analisar os resultados obtidos.

1.3 MOTIVAÇÃO

A pesquisa foi motivada pelo grande crescimento da utilização da abordagem de linha de produto de software por algumas empresas, podendo ser reutilizado o que já foi utilizado em outro projeto, visando diminuir a

complexidade do desenvolvimento podendo utilizar experiências já passadas. Essa abordagem objetiva a redução do custo e o seu tempo de produção e assim o aumento da sua produção.

Para automatizar o processo de desenvolvimento de software, a abordagem de Linha de Produto de Software é aplicável aos domínios em que existe uma demanda por produtos específicos, existindo um conjunto de características e pontos de variabilidades bem definidos. E o sistema em desenvolvimento no projeto SOLIGEE é altamente favorável a aplicação da abordagem de linha de produto, devido a sua arquitetura possibilita ser ajustada a maioria dos produtos de diferentes domínios no setor comercial.

Um das grandes motivações para a realização desse trabalho se deram pelo fato de que a abordagem de linha de produto se encaixa perfeitamente no desenvolvimento do CTEC.

1.4 ORGANIZAÇÃO

O trabalho está organizado da seguinte forma. O Capítulo 2 contextualiza linha de produto de software, sendo que na seção 2.1 são identificados os conceitos e definições, na seção 2.2 são identificadas as abordagens com ênfase em PLP. O Capítulo 3 contextualiza variabilidades e o modelo de características em linha de produto de software, sendo que na seção 3.1 mostra seus conceitos e definições, na seção 3.2 são expostos os conceitos da notação utilizada. O Capítulo 4 descreve como o trabalho será realizado, onde será realizado e como se pretende colher os dados, para compreender melhor a base do trabalho é preciso entender o que é o projeto, sendo que na seção 4.1 é contextualizado o Software Livre para Gestão Empresarial, na seção 4.2 demonstra o diagrama de classe, sendo que na seção 4.3 descreve a configuração de um sistema específico e na seção 4.4 o modelo de características demonstra as variabilidades. O Capítulo 5 apresenta os resultados obtidos no processo de desenvolvimento do trabalho de conclusão de curso, na seção 5.1 exibe as telas desenvolvidas, na seção 5.2 demonstra comparação em forma de gráficos das variabilidades e finalmente no Capítulo 6 apresenta-se a conclusão sobre o trabalho.

2 LINHA DE PRODUTO DE SOFTWARE

Algumas empresas pensam em aperfeiçoar seus processos de desenvolvimento de produto, buscando qualidade no produto e visando alguns objetivos, como: redução de custo, menor tempo de produção e produção com foco em vários produtos em desenvolvimento ao mesmo tempo.

As empresas ultimamente vêm trabalhando no desenvolvimento de produtos buscando uma melhor estratégia para que esse desenvolvimento seja finalizado contendo qualidade e que seu valor econômico esteja dentro das expectativas iniciais do processo de desenvolvimento. Uma abordagem inovadora e que pode atender essas expectativas dentro de uma empresa é à abordagem de Linha de Produto de Software.

Esta seção está contextualizando a Linha de Produto de Software, no qual o propósito é passar uma visão geral da abordagem de LPS e mostrar a importância de se utilizar uma linha de desenvolvimento dentro da empresa, demonstrar os motivos de estudar essa abordagem motivando as empresas a implantá-lo. O objetivo principal do capítulo é deixar claro para as empresas, que esta abordagem tem sido bastante aceita no mercado de desenvolvimento de software e muitas empresas estão obtendo resultados tanto na qualidade como benefícios econômicos com a adoção desta abordagem.

Segundo TRAVASSOS (2002) muitas abordagens que favorecem reutilização foram propostas nesses últimos anos. Algumas destas abordagens são: engenharia de domínio, frameworks, padrões, arquitetura de software e desenvolvimento baseado em componentes. Porém, o que vem sendo um problema é não ter uma maneira sistemática e previsível de utilizar a reutilização. Então, surge a proposta de linha de produto de software utilizando a construção sistemática de software baseada em uma família de produtos, ou seja, um conjunto de produtos de software com características suficientemente similares, para permitir a definição de uma infra-estrutura comum de estruturação dos itens que compõem os produtos e a parametrização das diferenças entre produtos.

O foco desta abordagem é reutilizar o máximo do que já se tem produzido, para que a empresa possua um grande aproveitamento do que se

fez assim aumentar a produtividade no desenvolvimento de novos produtos. O desenvolvimento de software é uma área promissora, porém, satisfazer os clientes pode ser uma dificuldade, os clientes exigem qualidade do produto e rapidez no desenvolvimento do produto.

Segundo PRESSMAN (2002) com o crescimento da procura e da complexidade dos softwares nas últimas décadas ocorreu uma contribuição para o surgimento de alguns problemas que estão ligados ao próprio desenvolvimento de software, problemas como: tempo estimado de finalização do software; conforme o aumento da complexidade maior será a dificuldade de desenvolvimento; as qualidades do processo e do produto não ficam de acordo com o que era esperado; dificuldades na manutenção e na sua evolução; e, de acordo com o mercado o software não evolui como deveria, isso ocorre quando a evolução não acompanha outras engenharias como a de hardware, sendo esse o principal problema.

Uma grande dificuldade encontrada no desenvolvimento de software é a modernização de outros componentes, como o *Hardware*, sempre as engenharias de *Hardware* está passos na frente da engenharia de software, com isso sempre aumenta as dificuldades no desenvolvimento do produto.

Além das dificuldades de hardware, existem outras dificuldades, como, qualidade do produto, ou seja, um produto de qualidade é aquele que o produto final esta de acordo com o esperado pelo cliente ou usuário e atinge todas as características definidas entre o desenvolvedor e o cliente, valor de desenvolvimento do produto, ou seja, o valor para se desenvolver um produto deve ser medido de uma forma que seja visto o quanto vai ser gasto para conseguir desenvolver determinado produto, para que assim possa analisar a viabilidade de seu desenvolvimento e tempo de entrega do produto é também um dos fatores de preocupação quando se trata de desenvolvimento de software, pois o produto deve ser entregue dentro do esperado pelo cliente e esse tempo é difícil de ser medido.

Em LPS, que visa o menor tempo de entrega e uma produtividade maior, esses fatores identificados podem prejudicar diretamente no desenvolvimento e em sua utilização dentro das empresas. Para isso é necessário que a empresa tome alguns cuidados para que esses problemas

não venham prejudicar o desenvolvimento dos produtos, pois um bom planejamento do processo pode facilitar no decorrer do desenvolvimento.

Segundo TRAVASSOS e GIMENES (2002) há muitos anos em engenharia de software vem se formando um pensamento de que para conseguir um produto de qualidade e que seu valor seja viável economicamente existe a necessidade de utilizar um conjunto de processos sistemático, abordagens e ferramentas. E dentro desse conjunto de abordagens a reutilização é vista como a mais relevante. Visando que com a reutilização de artefatos e partes específicas, desenvolvidas e testadas à construção do software pode ser finalizada em menor tempo de desenvolvimento e com uma confiabilidade ainda maior.

No desenvolvimento o grau de confiança que o software adquiriu, através de estudos pode perceber que o software além de ser confiável pode ser concluído em menor tempo de desenvolvimento. Para as empresas o que vem sendo utilizado é determinar um domínio em específico a ser trabalhado e assim analisar as qualidades idênticas no desenvolvimento para se obter um processo de desenvolvimento atendendo os princípios de Linha de Produto de Software.

Segundo RUBIRA (2009) com o grande crescimento na atividade de desenvolvimento de software vem ocorrendo alguns desafios como os termos de diminuição de custos, tempo de entrega relacionado ao esforço da equipe, aumento da dificuldade, complexidade e tamanhos dos produtos de software. Então novos enfoques vêm sendo utilizados para atender esses pedidos, ou seja, vem sendo proposto o reuso de artefatos de software, tais como o de Engenharia de Linha de Produto de Software. Essa engenharia tem o foco de projetar, implementar e evoluir produtos que possuem similaridades entre si, através de um conjunto de artefatos básicos. Desenvolvendo produtos de software em um determinado domínio a empresa pode ganhar significativamente na redução do esforço e custo adotando a linha de produto, desenvolvendo vários produtos similares ao mesmo tempo, ao contrario de desenvolver com foco em um único produto por vez.

As empresas podem ganhar tempo e qualidade desenvolvendo em uma vez somente para diversos produtos similares desde que seja um domínio

em específico. Algumas empresas podem adotar esta abordagem para que se possam concluir vários produtos de uma vez somente, pois o foco é vários produtos com um domínio, isso possibilita as empresas uma maior agilidade para manutenção e até mesmo no desenvolvimento dos produtos de software. O interesse das empresas ultimamente é produzir mais em menor tempo de desenvolvimento e lucrar mais, porém, quando se trata de LPS se produz mais em menor tempo, então o lucro é maior também.

Segundo RUBIRA (2009) muitas empresas hoje em dia, procuram desenvolver e manter seus produtos de software de forma que sejam similares no qual pertencem a um mesmo domínio de aplicação, ou seja, telecomunicações ou financeiro. Quando se trata de desenvolvimento de produtos de software com um grau de similaridade entre si, utilizar o enfoque de Linha de Produto de Software é bastante significativa e benéfico até mesmo economicamente e em termos de redução de custo e esforço e ainda com isso tem a melhoria da qualidade e produtividade do produto de software.

2.1 CONCEITOS E DEFINIÇÕES

Linha de Produto já era utilizada muitos anos atrás por empresas, mais era usada para produção de produtos manufaturados, hoje em dia o que tem visto é uma abordagem de LPS que é totalmente focada no desenvolvimento de software.

Para facilitar o entendimento dos conceitos de LPS e assim conhecer mais sobre esta abordagem, vão ser descrito os principais conceitos e definições de Linha de Produto de Software.

Linha de Produto de Software pode ser definida como a mesma forma de definição que é usada há anos atrás de quando se tratam de uma linhagem de produção de automóveis, roupas e etc. Linha de produto de Software é uma reutilização do que já foi produzido para produzir algo novo.

Segundo REINEHR (2004) Linha de produto é adaptar algo que está sendo usado internacionalmente como Software Product-Lines. Termo que é referência das linhas de produto de indústrias de manufatura, ou seja, foi introduzida uma revolução no processo de produção, no qual foi sugerido o

processo seqüencial de produção, que era baseado em repetir tarefas, sempre eram executadas pelas pessoas que tinham os recursos necessários.

Então dentro dessa definição, alguns conceitos e termos devem ser destacados, pois tem uma representação importante dentro do assunto de Linha de Produto de Software no qual é à base das principais vantagens.

Para o desenvolvimento de uma Linha de Produto de Software existe um processo de atividades para que se consiga desenvolver uma LPS com sucesso.

No desenvolvimento de uma LPS estão envolvidas basicamente três atividades distintas (CLEMENTS; NORTHROP, 2001; SEI, 2004):

- Desenvolvimento do núcleo de artefatos – também conhecida como Engenharia de Domínio;
- Desenvolvimento do produto – também conhecida como Engenharia da Aplicação;
- Gerenciamento de linha de produto.

As atividades vistas são iterativas, pois qualquer modificação em qualquer uma delas seria implicada na análise do impacto das outras atividades. Mais nem por isso a ordem das duas primeiras podem ser trocada, verificando-se o fato de que na atividade de desenvolvimento do produto, artefatos novos podem aparecer ou ser evoluído, ou até mesmo do núcleo de artefatos extraírem produtos já existentes. Na Figura 1, verifica-se a iteração das atividades.



Figura 1 Atividades essenciais da linha de produto de software

Fonte: SEI (2004).

Quando uma empresa pensa em implantar a abordagem de Linha de Produto de Software tem a necessidade de entender essas três atividades identificadas, é durante essas atividades que vão ser definido como o desenvolvimento do produto deve ser tratado.

Estas etapas devem ser bem definidas e elaboradas para que se obtenha sucesso e a empresa consiga seus objetivos estabelecidos, com isso é preciso entender cada uma dessas etapas.

A figura 2 exhibe uma das etapas, que é atividade de desenvolvimento do núcleo de artefatos, no qual, tem o objetivo de criar uma infra-estrutura de forma centralizada, pois será reutilizada pelos produtos gerados por uma Linha de Produto.

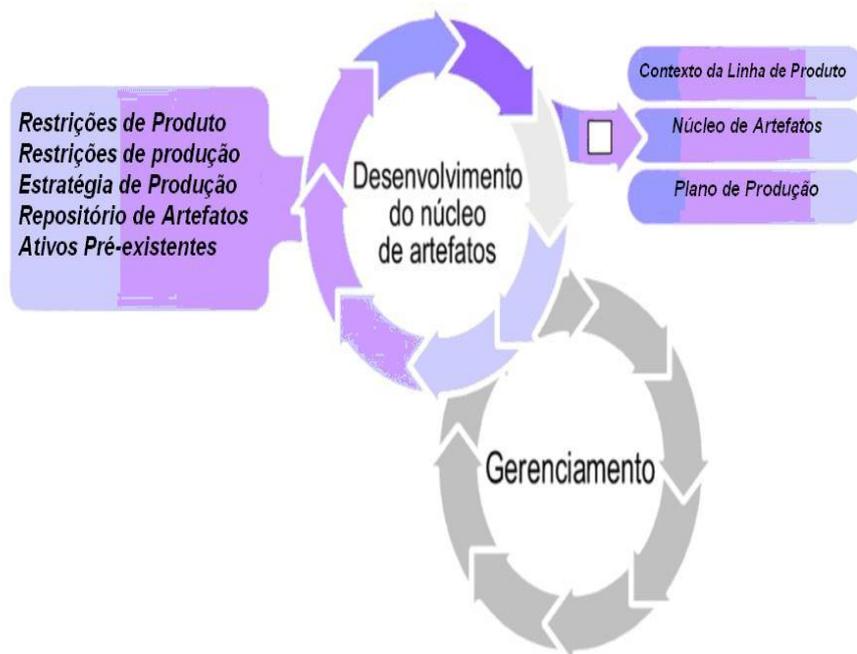


Figura 2: Desenvolvimento do núcleo de artefatos

Fonte: Adaptado de SEI (2008).

Segundo SEI (2009) o objetivo principal da atividade de desenvolvimento do núcleo de artefatos é o estabelecimento de capacidade de produção para os produtos. Esse desenvolvimento não ocorre em um vazio, mais sim, em um contexto situacional das suas restrições e recursos. Isso tem influencia na forma como o núcleo de desenvolvimento é realizado e a natureza dos resultados obtidos. Dentre eles, quatro dos mais importantes fatores são: restrição do produto, restrições de produção, estratégia de produção e ativos pré-existentis.

Seguindo o fluxo de atividades existentes pode identificar que uma atividade é dependente da outra, ou seja, atividade de desenvolvimento de produto é importante para que continue o ciclo das atividades.

Segundo SEI (2009) a atividade de desenvolvimento de produto depende das três saídas que são: campo de ação da linha de produtos, a base de ativos essenciais e o plano de produção. As atividades estão em um processo de iteração, como exemplo, quando um determinado produto por existir e estiver disponível ele puder afetar as qualidades de um produto no futuro.

A Figura 3 demonstra a atividade de desenvolvimento do produto na abordagem de LPS.

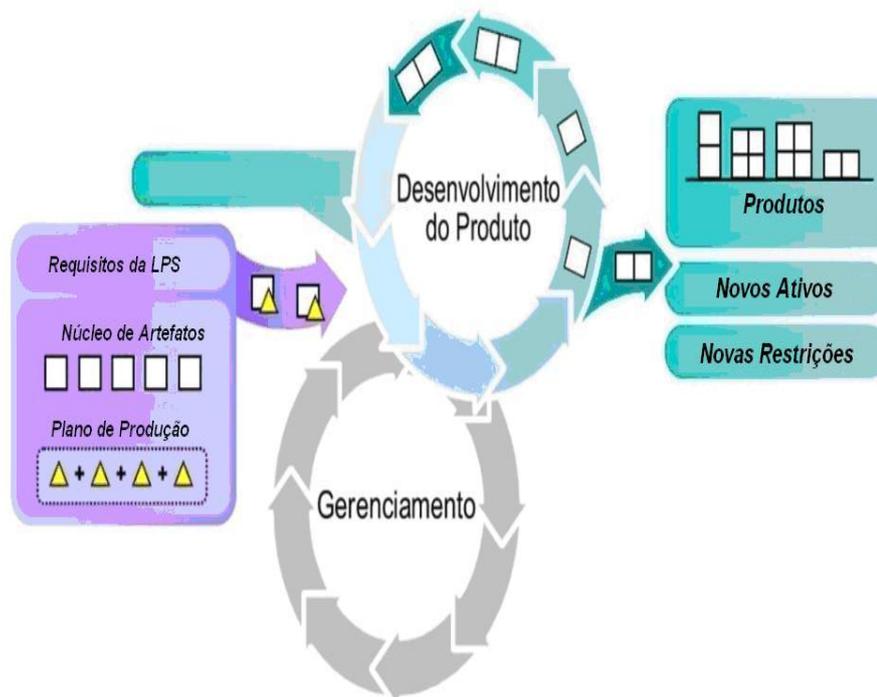


Figura 3: Desenvolvimento do Produto

Fonte: Adaptado de SEI (2008).

Para continuar no processo das atividades de desenvolvimento de LPS é necessário depois de desenvolver o núcleo de artefatos e desenvolver o produto é preciso saber como realizar o gerenciamento da Linha de Produto de Software.

Segundo SEI (2009) a gestão dentro das técnicas ou projetos e aos níveis da organização devem ser de forte comprometimento com o esforço da LPS. Pode ser manifestado de muitas formas esse comprometimento, pois alimentam o esforço da LPS.

Para chegar ao sucesso dentro dessas três atividades o ciclo de vida do processo deve ser saudável, ou seja, o processo não pode sofrer nenhuma falha, pois se não ocorrer pode ser vital.

Portanto, é possível observar, que a abordagem de LPS pode ser muito importante e pode trazer resultados gratificantes para a empresa, mais uma LPS, no qual, não foi bem formulada dentro das atividades pode trazer muito malefícios para a empresa, pois seu ciclo é vital.

LPS possui outros termos importantes que devem ser estudado para facilitar o entendimento do trabalho.

A variabilidade dentro da LPS é vista como variações que ocorrem dentro do desenvolvimento de novos produtos.

Segundo WEISS e CHI TAU (1999) variabilidade pode ser definido como a forma que os membros de uma família de produtos podem se diferenciar entre si.

Segundo SEI (2009) um ponto de vista dentro da variabilidade que tem a importância em um esforço na linha de produto é saber se as variações são necessárias ser identificadas no instante de definição da arquitetura dos produtos da linha ou se podem ser encontrados durante o desenvolvimento do produto em individual.

Uma LPS é composta por produtos com qualidades variantes e também qualidades similares que é o foco de LPS encontrar o que pode ser tratado como característica semelhante, pois encontrando as similaridades pode desenvolver novos produtos reutilizando suas semelhanças.

A figura 4 exhibe as atividades para o processo de gerenciamento de variabilidades na abordagem de LP.

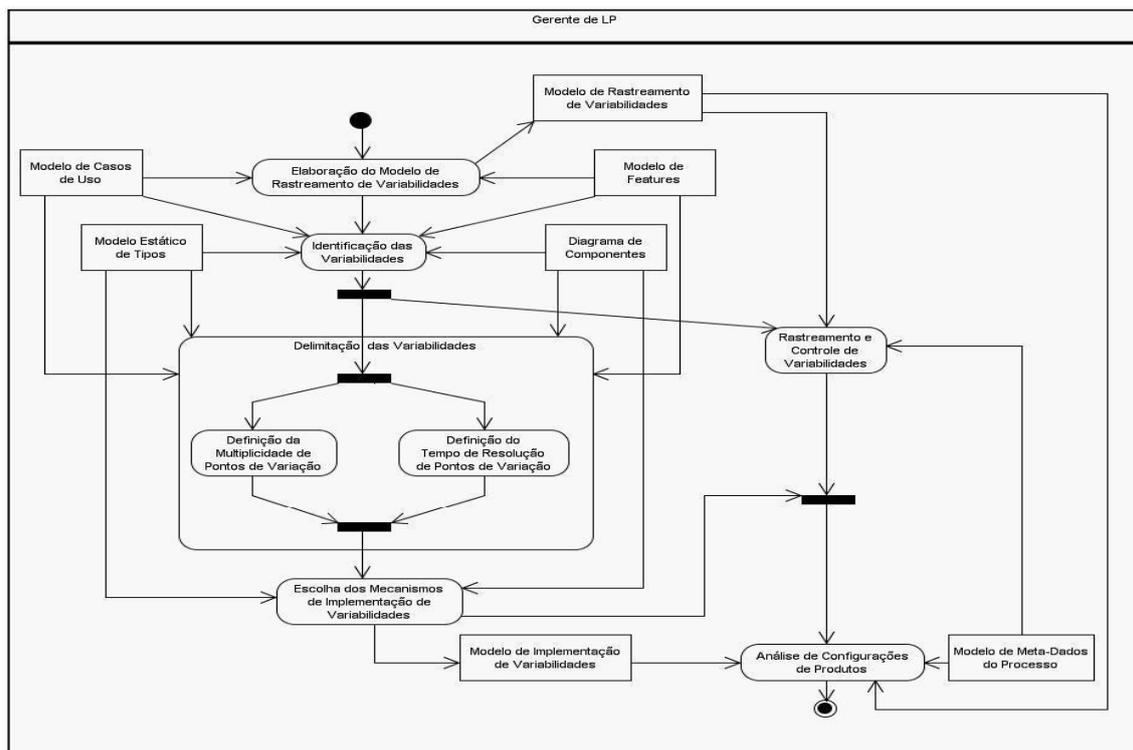


Figura 4: Atividades para o processo de gerenciamento de variabilidade para LP.

Fonte: Oliveira Junior (2005).

2.2 ABORDAGENS DE LINHA DE PRODUTO DE SOFTWARE

Como o crescimento da reutilização dentro do desenvolvimento de software tem aumentado, nesses últimos anos surgiram diferentes abordagens de Linha de Produto de Software, ou seja, quando pensa em Linha de Produto de Software aparece em destaque às seguintes abordagens: Product Line Practice (PLP), Feature Oriented Domain Analysis (FODA) e Family-oriented Abstraction, Specification and Translation (FAST), entre outras abordagens.

Durante essa seção será exibido algumas abordagens de LPS com uma ênfase maior na abordagem PLP, o objetivo dessa seção é demonstrar as abordagens de acordo com sua estrutura, assim facilitando identificar as diferenças entre elas.

Segundo OLIVEIRA JUNIOR (2005) a abordagem FAST trata de questões mais amplas de Linha de Produto. É uma abordagem precursora no qual serve de base possibilitando uma definição contextual mais geral para Linha de Produto.

Estas estratégias podem estar sendo utilizada como auxílio no momento de implementar a abordagem de LPS dentro das empresas.

Segundo TRAVASSOS e GIMENES (2002) a abordagem FAST durante a análise de características comuns é realizada através de uma série de reuniões com especialistas da área de domínio, com o auxílio de um moderador. A equipe de análise é quem produz o documento em grupo durante a reunião, definindo sobre o conteúdo com a guia do moderador. Normalmente, cada especialista tem o entendimento especializado de um determinado ou mais aspectos do domínio, porém o moderador deve possuir um conhecimento de experiência no processo de identificação de características comuns e possuir habilidade para identificar quando as definições estão precisas, claras e bem formadas, identificar características comuns, variabilidades, parâmetros de variação e questões úteis para composição do documento.

Existem diversas estratégias para a implementação da LPS e uma que é bastante usada é a PLP que foi desenvolvida pelo Software Engineering Institute (SEI), pois assim percebem-se diferenças da FAST.

Segundo TRAVASSOS e GIMENES (2002), um exemplo de estratégia de utilização para implementação de LP é a estrutura para prática de Linha de Produto de Software (PLP). O desenvolvimento de uma Linha de Produto (LP) está envolvido com atividades relacionadas ao desenvolvimento de artefatos centrais e desenvolvimento e gerenciamento do produto com o uso desses artefatos centralizados.

Segundo OLIVEIRA JUNIOR (2005) a iniciativa PLP não contém método de construção de Linha de Produto, mas ela tem um grande destaque por caracterizar e uniformizar os conceitos de LP, promovendo a sua utilização.

A utilização dessas abordagens ajuda na implantação, pois elas também fornecem um grau de dificuldade para o seu entendimento, para fazer um bom uso é necessário entender quais são as particularidades de cada abordagem.

Segundo TRAVASSOS e GIMENES (2002) para executar estas atividades tornam-se necessário definir qual é a área de trabalho que representa um conjunto menor de atividades, porém mais gerenciáveis. No qual uma área de trabalho possui um plano de trabalho e medidas relacionadas

que assim permitem acompanhar a sua execução e avaliar o sucesso dos trabalhos realizados. Estas áreas de trabalho usualmente produzem artefatos concretos, criando alguma forma, aos artefatos que estão centralizados que assim serão utilizados na LP.

Vendo como essa abordagem funciona em relação a Linha de Produto de Software é interessantes verificar quais são as áreas de trabalho que está definida dentro da PLP.

Segundo TRAVASSOS e GIMENES (2002) a estratégia PLP define três áreas de trabalhos.

1. **Engenharia de Software:** necessidade por ter que aplicar a tecnologia que é apropriada desde sua criação até sua evolução dos artefatos ou componentes centralizados e do produto.
2. **Gerenciamento Técnico:** relação com a criação e evolução dos artefatos ou componentes centrais.
3. **Gerenciamento Organizacional:** Utilizada para gerenciar os esforços com relação a LP como um todo.

Esta abordagem vem sendo evoluída de maneira continua de acordo com suas experiências adquiridas em diversos estudos de caso realizados com o apoio existente entre o SEI e as organizações que estudam Linha de Produto.

3 VARIABILIDADES E O MODELO DE CARACTERÍSTICAS

Adotando à abordagem de LPS utiliza-se o modelo de características para deixar visível aos desenvolvedores quais são as características similares e as suas variabilidades dentro desse modelo, pois para isso é necessário realizar a modelagem de características.

Esta seção está contextualizando as variabilidades e o modelo de características, no qual o objetivo é passar uma visão geral do contexto de modelo de características e suas variabilidades e demonstrar a importância de se utilizar o modelo dentro do desenvolvimento da LPS. O objetivo principal do capítulo é fazer com que o leitor compreenda que a modelagem do modelo de características é uma atividade fundamental dentro da abordagem de linha de produto de software.

Esta modelagem faz com que durante o processo de desenvolvimento do produto em uma LPS as características particulares de cada produto podem ser visível, formando a variabilidade da LPS. A modelagem pode facilitar para o desenvolvedor no entendimento do produto e durante o decorrer do processo, ou seja, no surgimento de novas características.

Segundo FERNANDES (2009) um fator principal no desenvolvimento de um sistema, no qual é utilizada à abordagem de LPS é que eles devem ser desenvolvidos através de componentes vindos de uma base de artefatos comuns, criados de acordo com regras definidas pela arquitetura.

A modelagem de características é uma atividade que através da complexidade dos requisitos do sistema, modela esses requisitos em forma de características para serem inseridas no modelo de características.

As características encontradas no sistema de software devem ser documentadas em um modelo de características, pois assim todos os requisitos vão estar relatados e descritos no modelo.

3.1 CONCEITOS E DEFINIÇÕES

Segundo RUBIRA e LOBO (2009) uma característica ou feature é propriedade do sistema que é importante para alguma parte do sistema e é

utilizada para adquirir funcionalidades comuns ou variáveis entre os sistemas de uma mesma família de produtos.

O modelo de característica deve ser elaborado e bem definido para que os desenvolvedores através dele, comecem a construir um produto utilizando a abordagem de LPS. E para que isso fique claro e essa construção do modelo de característica seja benéfica para o desenvolvimento é necessário estar atento aos conceitos e seus elementos.

Segundo RUBIRA e LOBO (2009) os elementos que são considerados como principais em um Modelo de Características são os seguintes: diagrama de características, regras de composição para as características e análise racional das características.

Diagrama de características: é responsável em representar as características que são consideradas padrões de uma família de sistemas em um mesmo domínio e o relacionamento entre famílias.

Regras de composição para as características: é nas regras de composição que deve ser definido qual vai ser a semântica entre as características que não devem ser inseridas no modelo de características. Como exemplo: para existir o teclado luminoso no computador requer que o computador seja mais valioso que mil reais.

Análise racional das características: é uma recomendação especificando de forma determinada uma característica de quando ela deve ou não ser utilizada, ou seja, no exemplo do computador, a recomendação é que a tela de Liquid Cristal Display (LCD) é mais econômica que a tela normal.

A utilização do modelo de características proporciona ao desenvolvedor uma maior visão sobre as variabilidades encontradas no desenvolvimento da linha de produto de software, pois o auxílio do modelo de características para encontrar novas variabilidades é um dos objetivos de se modular e desenvolver esse modelo.

Segundo LEITE, LAGUNA e BAIXAULI (2004) a variabilidade é definida como forma de alteração ou personalização de um sistema, pois existem dois grupos principais, no qual, os sistemas são separados, ou seja, variabilidade: sistema personalizado é devido principalmente para a seleção do usuário da peça que estiver interesse e famílias de produtos, pois grandes ou pequenos

números de produtos com similaridade são registrados para obter permissão da reutilização do comum.

Os sistemas que abordam linha de produto de software no seu desenvolvimento se diferenciam dos sistemas tradicionais, no qual, não é utilizada a abordagem de LPS, pois a principal diferença com os sistemas tradicionais é que os sistemas utilizando LPS devem prestar interesse na análise das características comuns e variáveis do sistema e com isso definir a relação entre eles.

A figura 5 ilustra o modelo de características de um processo de compra via Internet.

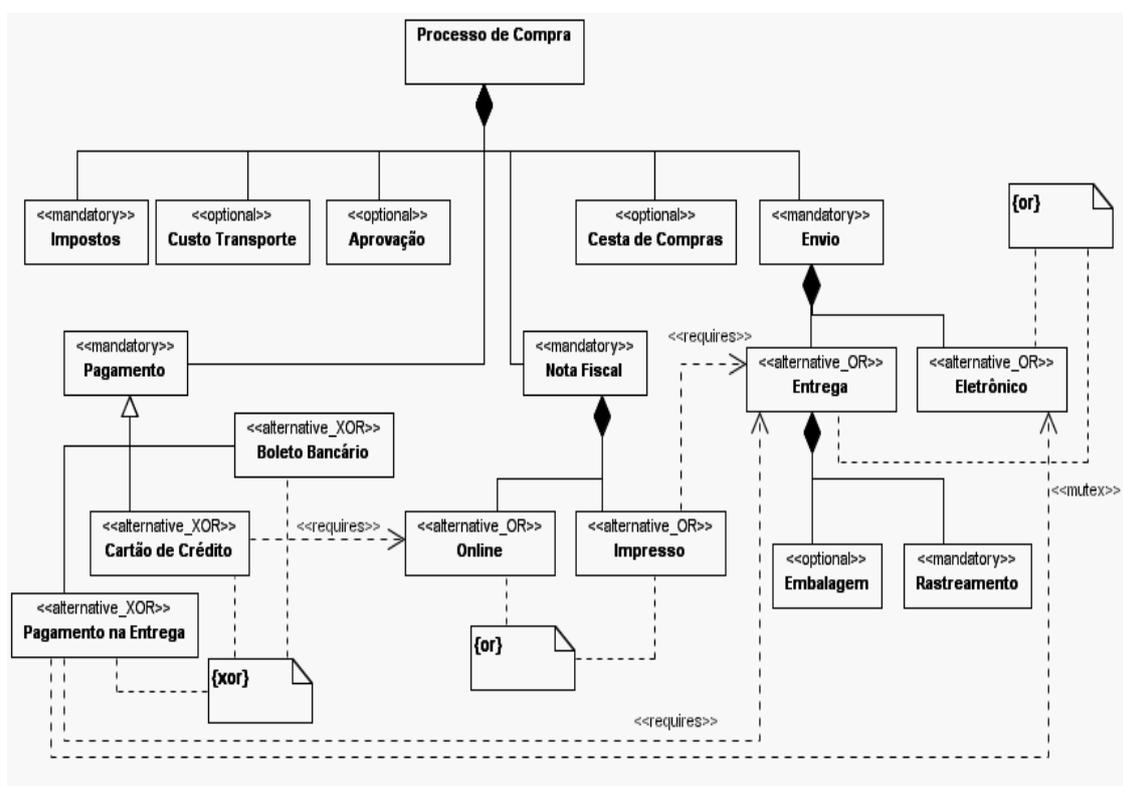


Figura 5: Modelo de Características de um processo de compra via Internet.

Fonte: OLIVEIRA JUNIOR (2005).

3.2 CONCEITOS DA NOTAÇÃO

A utilização de uma notação pode facilitar o entendimento do modelo e também auxiliar na padronização dos modelos desenvolvidos, pois isto é um dos motivos de adotar uma notação dentro do modelo de características.

Em uma notação é evidente a utilização de conceitos e características particulares, pois não é diferente na notação utilizada no decorrer do desenvolvimento do presente trabalho, ou seja, a notação GOMMA na qual está sendo utilizada para modelagem do modelo de características possui as suas particularidades e seus conceitos.

Segundo TEIXEIRA (2007) utilizando a notação de GOMAA na modelagem de características, as características e seus grupos de características obtêm a forma de representação como classes sem atributos e métodos, especificadas por modelo estabelecido como padrão.

Segundo TEIXEIRA (2008) a representação da notação de GOMAA é feita através do estereótipo da UML, este relacionamento gera uma subclassificação para as características.

Os diagramas modelados na abordagem de linha de produto de software utilizam-se os conceitos de GOMAA e são baseados nos conceitos da UML.

3.2.1 Notação para o Modelo de Características

O modelo de características é organizado da seguinte maneira, no topo da árvore é inserido o objetivo de todo o processo a ser desenvolvido, ou seja, uma funcionalidade completa. Para que essa atividade seja concluída com sucesso é fundamental que as outras ramificações da árvore estejam bem definidas.

As ramificações no modelo de características são as classes que devem estar representadas para um melhor entendimento do desenvolvimento do produto de Software. Em cada ramificação deve comparecer o seu relacionamento, pois essas relações são expressas por setas acompanhadas de um dos estereótipos ou padrão.

Segundo GOMAA (2004) através da opcionalidade é possível classificar as características da seguinte maneira: características comuns são as características que devem ser utilizadas por todo membro da LPS, ou características opcionais e características alternativas que representam o grau de variabilidade da LPS.

A divisão por variabilidade faz com que o modelo de características fique mais bem representado e modelado para o desenvolvimento do produto, pois essa representação deixa claro quais características são indispensáveis e quais são opcionais para o desenvolvimento do produto na abordagem de LPS.

Dentro do modelo de características é possível separar as classes de acordo com o seu grau de variabilidade, ou seja, é necessário identificar qual é o impacto da característica na linha de produto de software. Para realizar essa modelagem segue algumas características da notação utilizada.

- Características comuns: podem ser também denominadas como mandatórias ou <<*mandatory*>>, pois essa é a forma de representar as características obrigatórias.
- Características opcionais: é responsável por representar os requisitos que só são utilizados por alguns membros da LPS. A representação é através do padrão <<*optional feature*>>.
- Características alternativas: utilizadas quando características podem se tornar alternativas para seleção dentro de um grupo, Exemplo: o *Air-bag* pode ser alternativo para um carro.

O objetivo de se utilizar essa classificação entre as características é o agrupamento das características que possuam pré-requisitos em comum, pois os grupos representam restrições de como as características devem ser utilizadas por determinado membro de uma Linha de Produto de Software.

3.2.2 Notação para o Diagrama de Casos de Uso

A modelagem do diagrama de caso de uso é fundamental para que sejam exibidas as funcionalidades do sistema, ainda podendo demonstrar a qual tipo de função cada usuário do sistema vai poder realizar, ou seja, cada

ator do diagrama de casos de uso pode executar determinadas tarefas, Exemplo: Cliente → Efetuar Pagamento, Cliente → Incluir Produto na Cesta.

Segundo OLIVEIRA JUNIOR (2005) o modelo de caso de uso é fundamental no desenvolvimento de Linha de Produto, ou seja, a modelagem é o início para identificar as características e com isso encontrar as variabilidades existentes na LP.

Utilizando LPS no processo de desenvolvimento do produto devem-se analisar os requisitos necessários para a funcionalidade exata do software para o qual está sendo desenvolvido, pois desta forma verificar essas funções e interações para a modelagem do diagrama de caso de uso, a modelagem deve ser desenvolvida com base nas informações adquiridas, através de entrevistas, reuniões dentre outras formas.

Segundo BRITO (2001), para a modelagem deste tipo de diagrama segue algumas características da notação utilizada para o desenvolvimento do modelo.

- Ator: é alguém ou algo que deve realizar alguma interação com o sistema.
- Casos de Uso: são ações que devem ocorrer através de interações entre o ator e o sistema e assim permitir ao mesmo atingir o seu objetivo. Cada caso de uso é uma seqüência de possíveis interações realizada por um ator e o sistema em um determinado momento.

Existem vários tipos de relações entre os elementos que constituem o diagrama de casos de uso, entre eles estão:

- Associação: Uma relação considerada básica com o objetivo de representar a interação entre o ator e o caso de uso.
- Dependência: Uma particularidade de relação entre as classes, pois uma classe depende da outra.
- Generalização: É uma relação que pode ter duas definições, ou seja, Inclusão (<<include>>) é utilizada para evitar repetição ao dividir uma atividade comum a dois ou mais casos de uso, Herança (<<extends>>) utiliza-se herança quando é preciso representar uma variação/extensão do comportamento do caso de uso base.

Na figura 6 exibe o modelo de casos de uso para uma melhor visualização dos conceitos apresentados anteriormente.

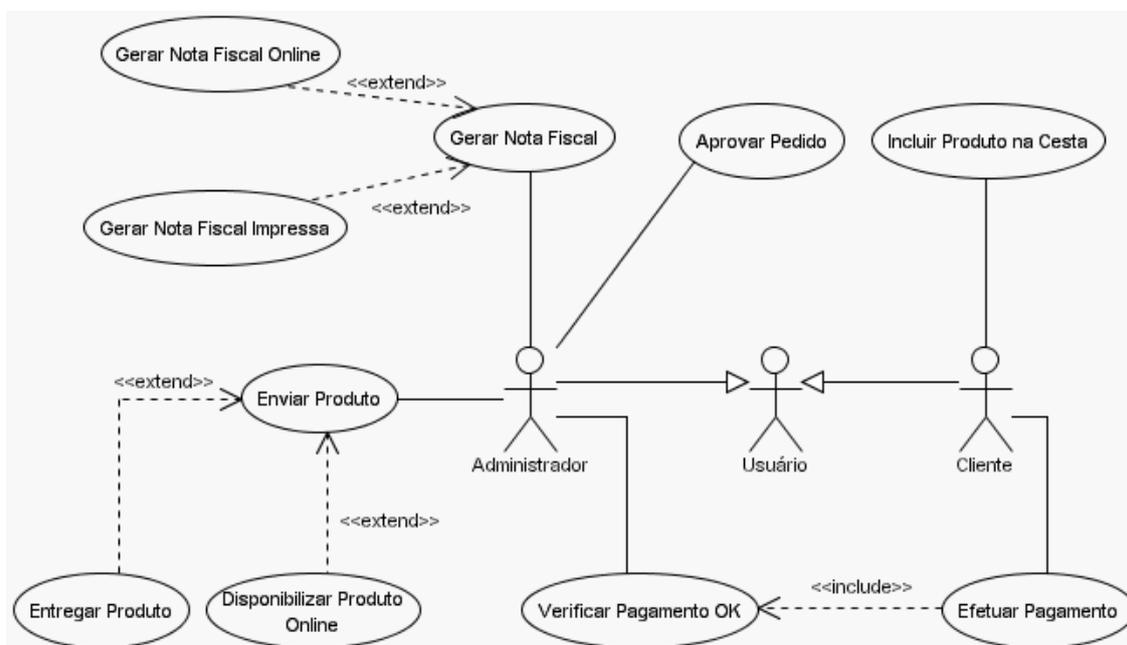


Figura 6: Modelo de casos de uso de um processo de compra via Internet.

Fonte: OLIVEIRA JUNIOR (2005).

Segundo GOMAA (2004) foi proposto três tipos de casos de uso para serem utilizados na abordagem de LPS: obrigatórios <<Kernel>>, no qual estão presentes em todos os membros da LP, opcionais <<optional>> são os que aparecem somente em alguns produtos e alternativos <<alternative>> o qual foi criado com intuito de distinguir esses elementos.

Descrevendo a figura 8: o cozinhar é o <<Kernel>>, pois em todos os microondas contém a função de cozinhar, porém exibir hora do dia, definir hora do dia e cozinhar com receita são <<Optional>>, pois não são todos os membros da LPS que podem fornecer a hora do dia e possuir a capacidade de cozinhar com receita.

A figura 7 exibe o diagrama de caso de uso utilizando a notação descrita anteriormente.

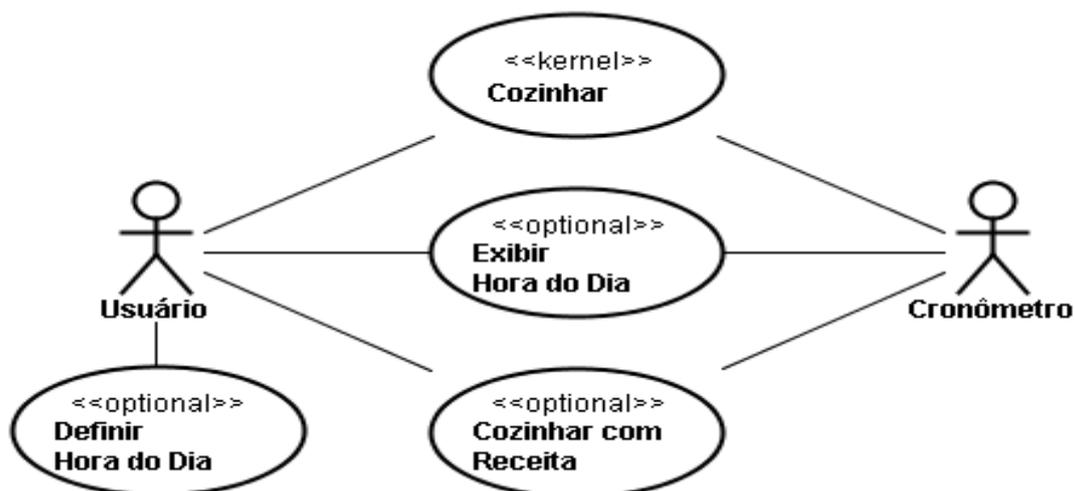


Figura 7: Diagrama de Caso de Uso de um microondas em LPS.

Fonte: Adaptado de (GOMAA 2004).

3.2.3 Notação para o Diagrama de Classes

O diagrama de classe objetiva descrever a visão estática do sistema em termos de classes e seus relacionamentos entre as classes, a modelagem é importante para definir a base para que outros diagramas apresentem outras visões do sistema.

Este diagrama representa a modelagem da parte estática do sistema, no qual é representado um conjunto de classes contendo seus atributos, operações e associações.

Uma boa modelagem necessita de um bom entendimento dos conceitos da notação utilizada, pois a seguir são demonstrados alguns conceitos de *Unified Modeling Language* (UML) utilizada para a modelagem do diagrama de classes.

Segundo TANAKA (2009) são descritos alguns conceitos utilizados para a modelagem do diagrama de classes, os quais são:

- Classe: as classes são descritas como uma abstração que enfatiza características relevantes e suprime outras características. Uma classe contém a seguinte estrutura, nome da classe, estrutura (atributos) e

comportamento (operações).

- Multiplicidade: a multiplicidade ou cardinalidade é responsável por especificar quantas instâncias de uma classe relacionam-se a uma única instância de uma classe associada.
- Classe associativa: essa classe é criada para conter atributos e/ou operações que fazem parte da relação e não a qualquer uma das classes relacionadas.
- Agregação: é a associação, porém utiliza um todo ou uma parte, no qual, os objetos que representam os componentes de algo são associados a um objeto que representa a estrutura inteira, Exemplo: um departamento agrega valores para uma empresa.
- Composição: um formulário de agregação com posse forte e tempos de vida e sua agregação não é compartilhada.
- Dependência: relacionamento entre dois objetos e possui a finalidade de determinar onde os relacionamentos da estrutura são requeridos. O seu relacionamento não é estrutural.
- Classe abstrata: uma classe que não pode ter instâncias diretas, já os seus descendentes sim, serve apenas como base para reutilização de código por outras classes.
- Classe concreta: é a classe que pode ser instanciada diretamente.
- Método abstrato: método de uma classe que não será implementado pelo sistema, ele deve ser implementado por todas as classes descendentes concretas.
- Método Concreto: é o método de uma classe que será implementado utilizando uma linguagem de programação orientado a objetos.

Na figura 8 segue os passos para a modelagem do diagrama de classes, pois cada círculo refere-se a uma atividade que deve ser executada para a conclusão do diagrama.

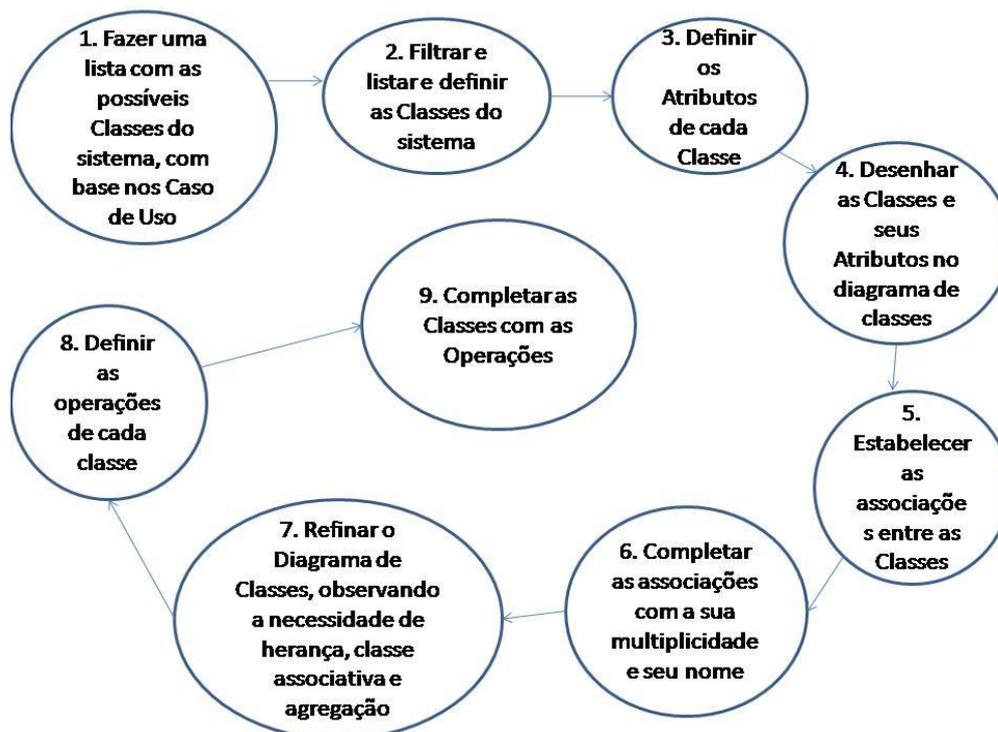


Figura 8: Etapas para modelar o Diagrama de Classes.

Fonte: Adaptado de TANAKA (2009).

Como é necessário em LPS utilizar a variabilidade tanto na modelagem do diagrama de caso de uso e modelo de características é fundamental também para a modelagem do Diagrama de Classes na abordagem de LPS utilizar o conceito de classificação entre as classes, pois assim ficam visíveis as variabilidades entre as classes.

Segundo TEIXEIRA (2007) as classes devem ser classificadas entre: obrigatórias, opcionais e alternativas na abordagem de LPS.

Obrigatórias: quando classificadas como obrigatórias podemos entender que estas classes estarão presentes em todos os produtos gerados pela Linha de Produto de Software, porém a escolha de seus atributos ou métodos dependerá da escolha das características.

Opcionais: as classes classificadas como opcionais são aquelas que estarão presentes em alguns produtos gerados pela LPS.

Alternativas: já as classes classificadas como alternativas servem para determinar que estas classes possuem pontos de variações.

A figura 9 demonstra como utilizar a classificação de variabilidades no diagrama de classes na abordagem de LPS.

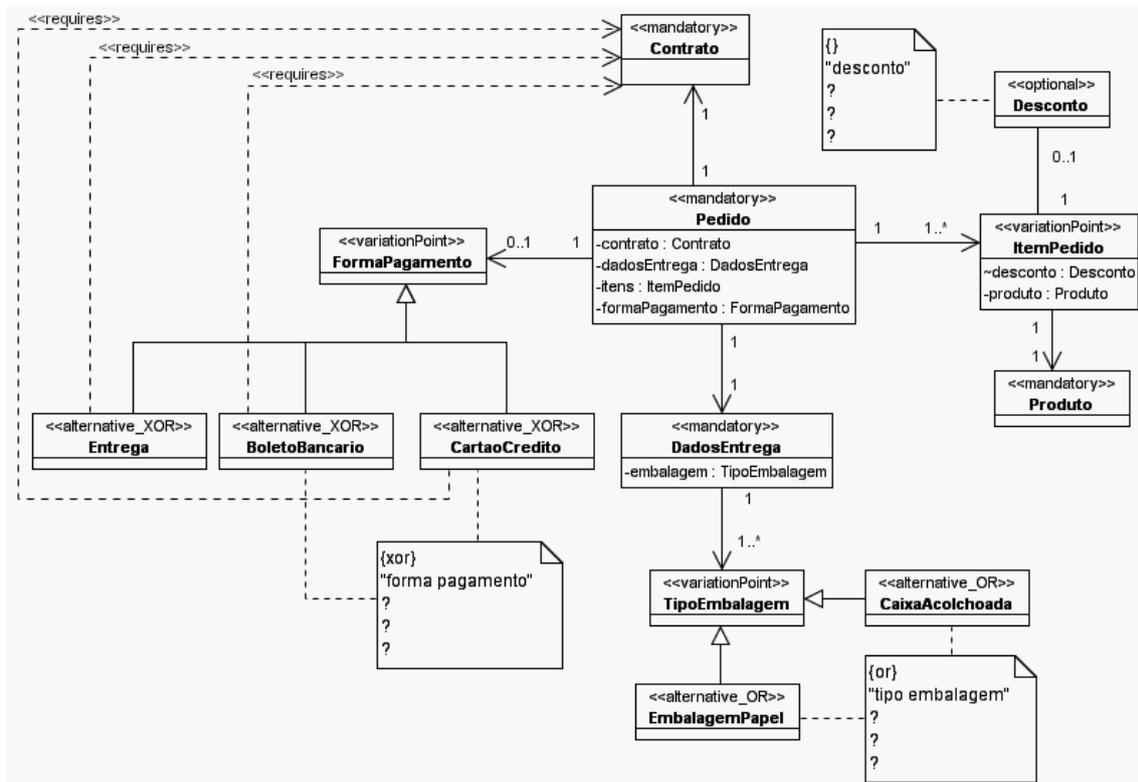


Figura 9: Representação de variabilidades em Diagrama de Classes.

Fonte: OLIVEIRA JUNIOR (2005).

4 DESENVOLVIMENTO

Este capítulo descreve quais métodos foram utilizados e como ocorreu o desenvolvimento do trabalho.

4.1 MÉTODOS

Para que os resultados dessa pesquisa fossem satisfatórios, a elaboração do projeto de pesquisa e o desenvolvimento da própria pesquisa foram baseados em planejamento cuidadoso, reflexões conceituais e conhecimentos já existentes.

Sendo assim, do ponto de vista de sua natureza e abordagem essa pesquisa configura-se como uma pesquisa aplicada e quantitativa. Aplicada, pois objetiva gerar conhecimentos para aplicação prática dirigidos à solução de problemas específicos. Envolve verdades e interesses locais por meio do desenvolvimento de um produto de software que atenda as demandas do Centro de Tecnologia de Carnes da UENP-CLM. Quantitativa, pois foram considerados dados quantificáveis para análise e comparação.

Desta forma, para o desenvolvimento da pesquisa a técnica utilizada foi um estudo de caso, para configuração de um novo produto de software utilizando o núcleo de artefatos de uma Linha de Produto de Software existente.

O objetivo do estudo de caso foi de coletar dados quantificáveis para comprovar a hipótese de que a inserção de novas características numa LPS impacta diretamente o projeto de software.

Na seção 4.1 demonstra o contexto que está inserido o estudo e a LPS existente, descrevendo o projeto SOLIGEE.

4.2 SOFTWARE LIVRE PARA GESTÃO EMPRESARIAL

O Projeto SOLIGEE (Software Livre para Gestão Empresarial) foi inicializado em dezembro de 2008, formado por interesse dos Professores da Universidade Estadual do Norte do Paraná - Campus Luiz Meneghel (UENP-CLM) e Diretores da Associação Comercial, Industrial e Agrícola de

Bandeirantes (ACIAB), o foco é disponibilizar recursos tecnológicos aos associados.

O objetivo do projeto é informatizar as empresas que tem relação ao comércio de Bandeirantes associadas à ACIAB, fornecendo um Sistema de Gerenciamento Comercial de Código Livre e Gratuito auxiliando as empresas no controle das informações recentes aos cadastros, estoque de mercadorias e movimentação financeira.

Esse sistema tem uma grande importância para esses empresários, pois o gerenciamento de informações (cadastros, consulta, cálculos, etc.) atualmente é manual, com isso gera atrasos, erros, inconsistências e outros fatores que acabam prejudicando a progressão da empresa e muitas vezes fazendo com que seu principal objetivo não seja alcançado: satisfazer o cliente.

O desenvolvimento do projeto foi dividido em várias etapas, segue abaixo uma lista das Metas realizadas.

- **Meta 1:** Análise de Requisitos – Levantamento dos requisitos mínimos para atender as necessidades dos comerciantes, através do acompanhamento das atividades cotidianas. Coleta de informações e processos em lojas que já possuíam algum sistema legado, realizando engenharia reversa desses sistemas;
- **Meta 2:** Estudar uma metodologia que melhor apóie o desenvolvimento da solução proposta;
- **Meta 3:** Estudar tecnologias e ferramentas para a implementação do software de gerenciamento comercial , priorizando obter como resultado um sistema simples, fácil de se utilizar e que atenda com eficiência as necessidades dos comerciante;
- **Meta 4:** Planejamento e documentação do sistema baseado na análise realizada;
- **Meta 5:** Estudo e implantação de Sistema online para controle de versão de código, gerenciamento de tarefas, controle de erros e correções. Os sistemas estão em uso atualmente no endereço eletrônico www.las.atualizado.com.br.

- **Meta 6:** Definição de Regras de Negócio e Modelagem do Banco de Dados;
- **Meta 7:** Desenvolvimento de módulos de Cadastros.
 - Cadastro de usuário do sistema;
 - Cadastro de clientes: Pessoa Física e Pessoa Jurídica;
 - Cadastro de fornecedores;
 - Cadastro de Produtos (categorias, subcategorias, marcas, etc.);
 - Módulo de Orçamentos;
 - Módulo de Pedidos;
 - Módulo de Venda;
 - Módulos de Notas de Entradas e Saídas;
- **Meta 8:** Desenvolvimento de módulos de Consultas e Edição dos dados.
 - Usuário do sistema;
 - Clientes: Pessoa Física e Pessoa Jurídica;
 - Fornecedores;
 - Produtos (categorias, subcategorias, marcas, etc.);
 - Orçamentos;
 - Pedidos de Compra;

4.3 CONFIGURAÇÃO DE UM SISTEMA ESPECÍFICO

Utilizando os conceitos de LPS foi estendido um método para desenvolvimento de uma aplicação específica, no caso um sistema para gerenciamento das principais atividades do Centro Tecnológico de Carnes (CTEC) da Universidade Estadual do Norte do Paraná - Campus Luiz Meneghel com o objetivo de promover a reutilização do núcleo de artefatos do sistema do Projeto SOLIGEE e assim melhorar a qualidade da aplicação já desenvolvida. Reutilizando partes bem especificadas, desenvolvidas e testadas do SOLIGEE é possível construir um sistema para o CTEC em menor tempo e com maior confiabilidade, mostrando os processos e técnicas para a obtenção de um produto de software de qualidade alta e com custo-benefício viável.

Primeiramente foi necessário analisar o núcleo de artefatos do projeto SOLIGEE, levando em conta os três aspectos que o desenvolvimento do

núcleo de artefatos contém que é a definição do contexto da linha de produto, o núcleo de artefatos e o plano de produção; também é importante frisar que artefato é um item reutilizável de software utilizado como bloco de construção de uma linha de produto, e nesse caso o item reutilizável será toda a documentação relativa à arquitetura de software do SOLIGEE bem como seus componentes já existentes. Estes artefatos incluem uma arquitetura base e um conjunto de componentes para preencher esta arquitetura. O projeto de uma arquitetura para uma família de produtos considera os aspectos comuns e as variabilidades entre os produtos, isso foi o que se levou em consideração para o desenvolvimento do sistema para o CTEC, bem como o que pode variar entre os produtos com a inclusão de novos requisitos. A partir dessa análise, foi extremamente necessário e importante verificar quais foram esses novos requisitos precisos para a implementação do sistema para gerenciamento específico das atividades do Centro Tecnológico de Carnes da UENP-CLM, a configuração do sistema foi iniciada, como já foi dito, reutilizando o núcleo de artefatos já existente, e atendendo aos requisitos específicos, pois as necessidades que não foram supridas pelo modelo do domínio do SOLIGEE, implicaram na necessidade de novas funcionalidades, o que são denominadas novos requisitos.

Esses novos requisitos implicaram em estender a arquitetura da linha de produto do SOLIGEE e adquirir novos componentes. E para os requisitos não satisfeitos pelo modelo já existente, será investigado o impacto de modificar a sua arquitetura ou seus componentes existentes, também será investigado qual será o impacto em desenvolver um sistema específico com a reutilização de uma linha de produto de outro sistema com domínio diferente, levantando as vantagens dessa reutilização.

4.4 DIAGRAMAS DE CLASSES

Na figura 10 mostra o diagrama de classes do SOLIGEE, exibindo todas as classes, desde as classes que já estão concluídas quanto às classes que estão em fase de desenvolvimento. Na fase inicial do projeto SOLIGEE foi desenvolvido um diagrama de classes, no qual, foi preciso então uma

atualização desse diagrama, pois foi necessário inserir novas características e algumas outras já não seriam mais necessárias. No entanto após a atualização do diagrama começou a fase de expansão do mesmo, pois foram inseridas as características do CTEC, realizando a junção do diagrama de classes do SOLIGEE juntamente com o CTEC.

No SOLIGEE o diagrama de classe é dividido em três pacotes que correspondem às camadas *viewer*, *controller* e *model*, pois na camada *viewer* contém um pacote com as classes de interface, na camada *controller* está um pacote com as classes ligadas à lógica de negócio e na camada *model* possui um pacote chamado VO que contém outros pacotes que são Cadastros e Lançamentos.

Como exibe a figura 11, no diagrama de classes do CTEC as classes são separadas seguindo o modelo do SOLIGEE, pois as classes são divididas em pacotes, no entanto foi necessário modelar novos pacotes nesse diagrama para melhor organização. Dentro do pacote *Viewer* pode perceber que foi mantido o diagrama e dentro dele foram colocados dois outros pacotes, um com o nome de “*viewer.Soligee*” contendo as classes de interface do SOLIGEE, como no primeiro diagrama, e o novo pacote foi chamado de “*viewer.Ctec*” contendo as classes de interface do sistema do CTEC. No pacote *Controller* também foi mantido e dentro dele foram colocados dois pacotes, um chamado “*controller.Soligee*” contendo o pacote contendo as classes referentes a lógica de negócios do SOLIGEE e o outro pacote chamado de “*controller.Ctec*” contendo a lógica de negócios do sistema para o CTEC. Dentro do pacote *model* os pacotes Cadastros e Lançamentos continuaram contendo suas respectivas classes, com a utilização da abordagem de Linha de Produto de Software para o desenvolvimento de um novo produto surgiu então a necessidade de inserir um novo pacote chamado “Centro Tecnológico de Carnes” para conter as classes do novo sistema, o pacote contém as classes *CortesVO*, *AnimalBrutoVO*, *ClientesCtecVO* e *TipoClienteVO*, demonstrando seus relacionamentos.

detalhada cada camada do modelo. A figura 12 corresponde as camadas Viewer e Controller do diagrama de classe do SOLIGEE.

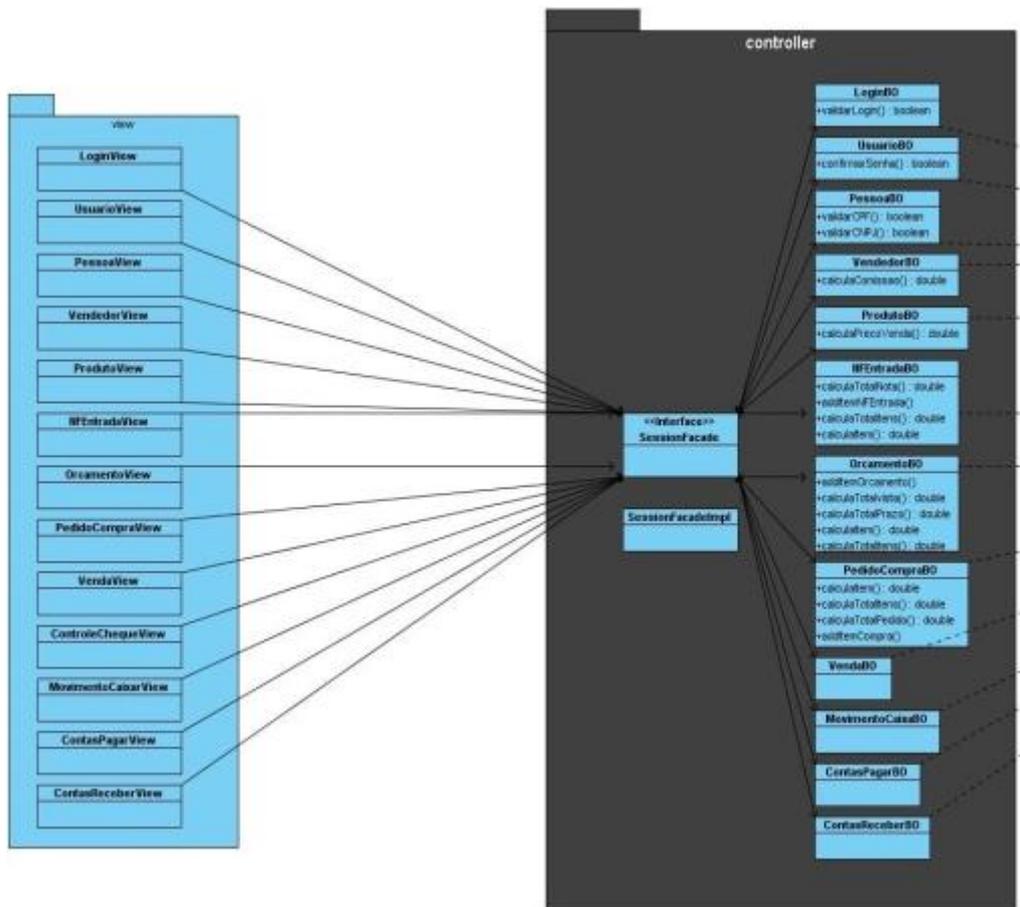


Figura 12: Camadas Viewer e Controller - Diagrama de classes SOLIGEE.

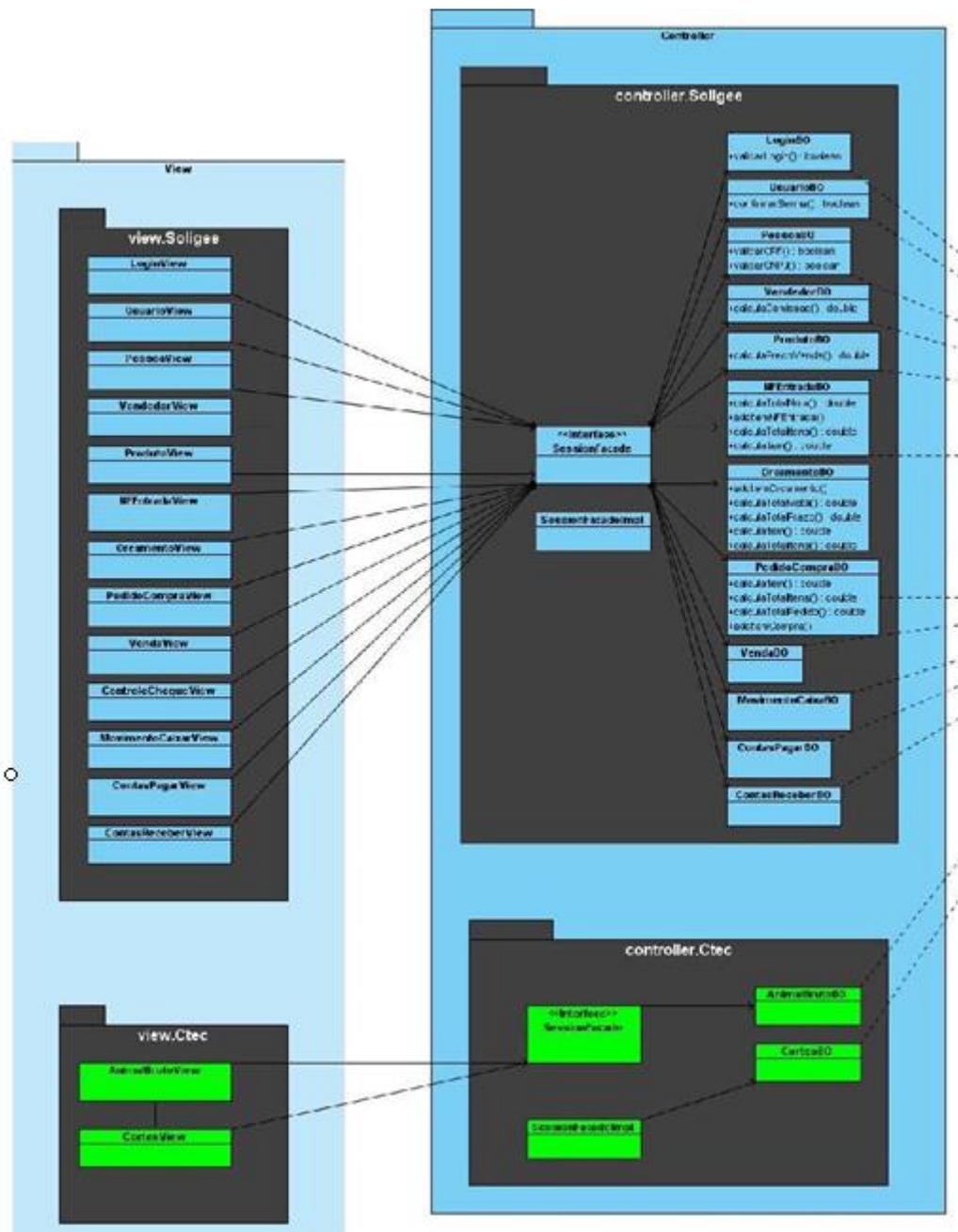


Figura 13: Camadas Viewer e Controller - Diagrama de classes CTEC.

A seguir mostra-se a camada Model dos dois diagramas de classes, a figura 14 é referente a camada no diagrama de classes do SOLIGEE e a figura 15 é referente a camada no diagrama de classes do HV.

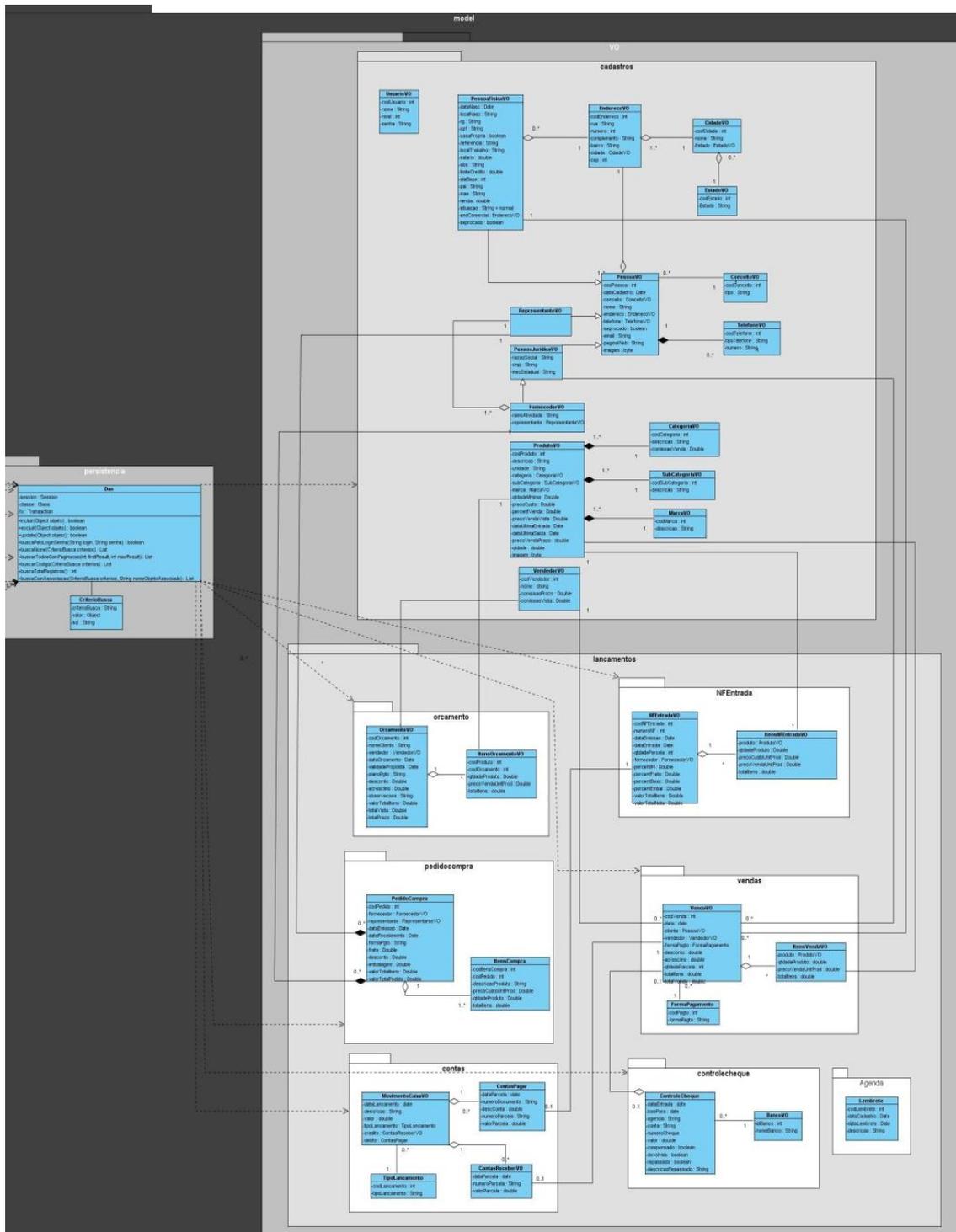


Figura 14: Camada Model - Diagrama do SOLIGEE.

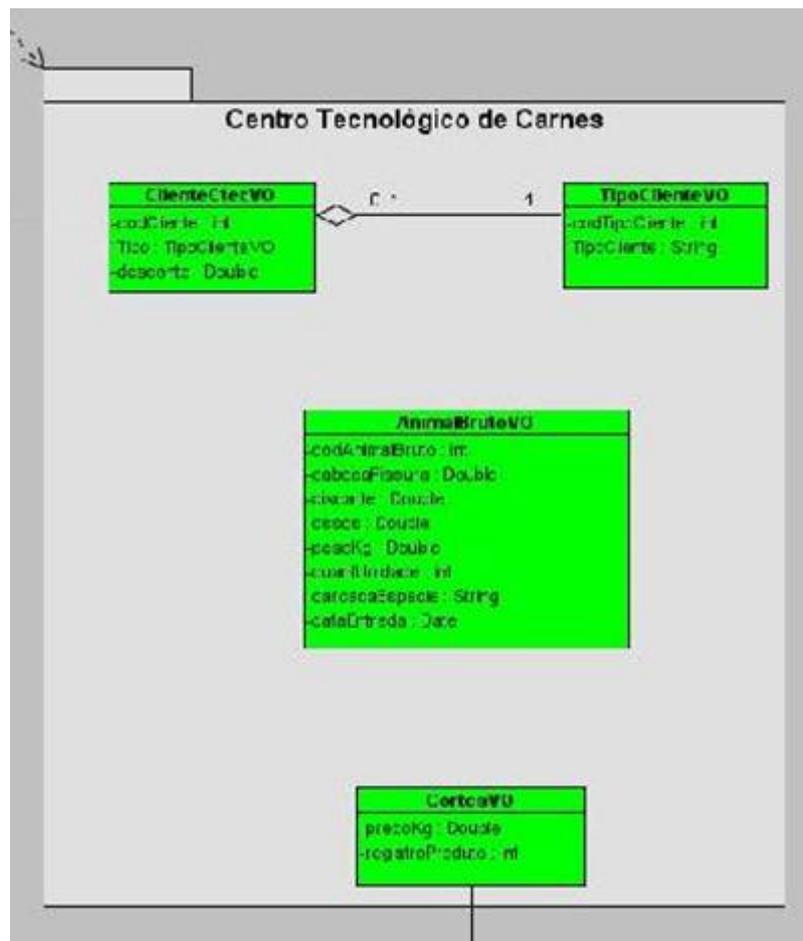


Figura 15: Camada Model - Paquete Centro Tecnológico de Carnes

4.5 MODELOS DE CARACTERÍSTICAS

A atividade de modelagem do modelo de características deve ser bem estudada e planejada, pois neste modelo constam as informações do novo produto a serem desenvolvidas (CTEC) como as características do núcleo (SOLIGEE), pois é nesta atividade que o desenvolvedor deve definir quais serão as variabilidades dentro do desenvolvimento do novo produto.

O modelo de características facilita o entendimento do produto que está sendo desenvolvido, ou seja, no modelo deve estar definido quais serão as características específicas e quais serão as características variáveis de cada produto.

A figura 16 exhibe as características do SOLIGEE enquanto na figura 17 são demonstradas as características do CTEC em azul após a configuração do produto.

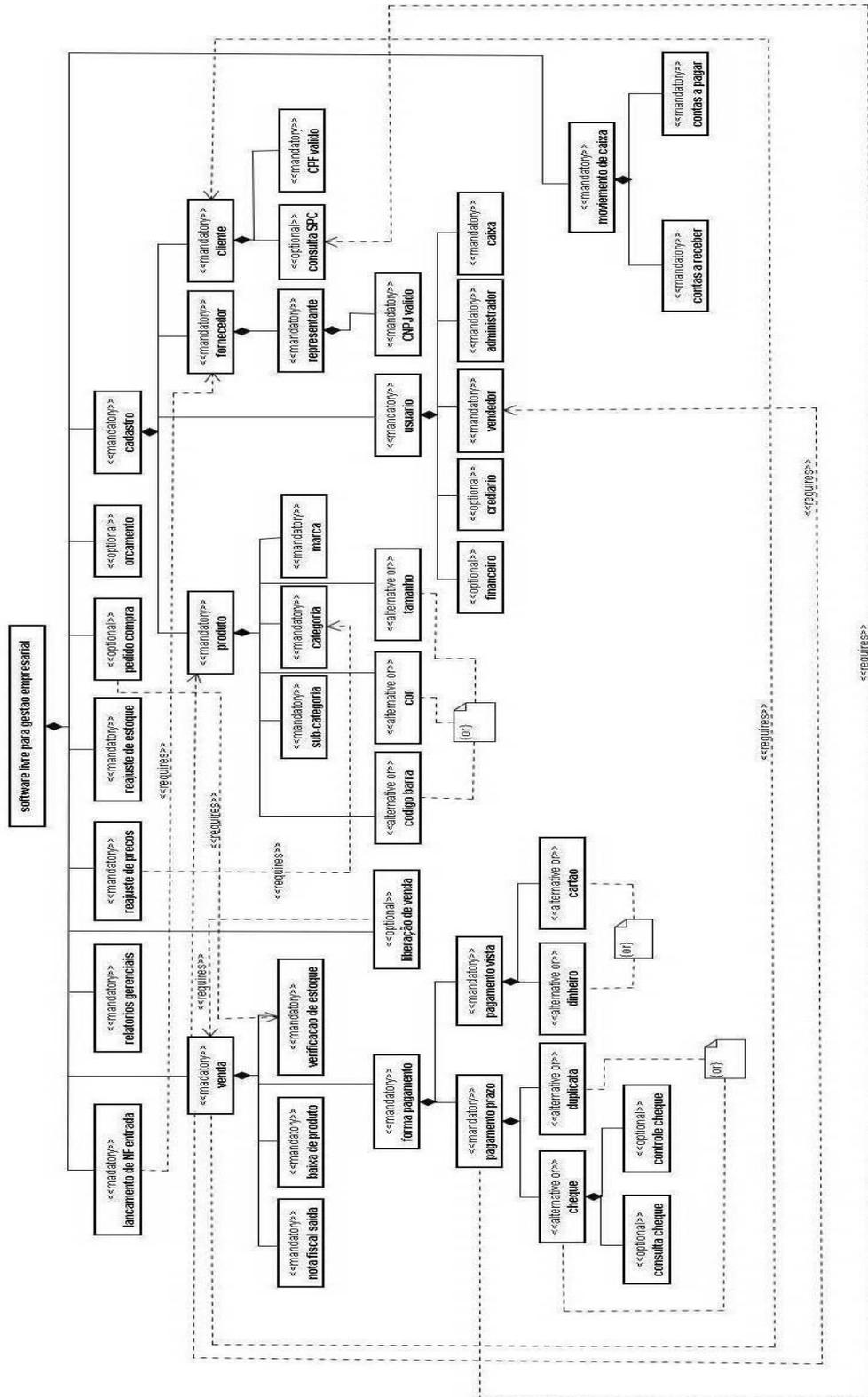


Figura 16: Modelo de Características do SOLIGEE.

5 RESULTADOS

Nesta seção o foco é demonstrar os resultados obtidos durante todo o processo de desenvolvimento do trabalho de conclusão de curso, visando evidenciar o que foi criado a partir da utilização das abordagens e conceitos em linha de produto de software e de forma gráfica representar os números encontrados.

5.1 TELAS

As telas foram desenvolvidas para suprir as necessidades do CTEC, para desvendar as dificuldades e o que deveria ser feito de forma que facilitasse a utilização, pois houve a necessidade de realizar uma entrevista com o usuário e com um profissional especializado no ambiente para o qual o Software estava sendo desenvolvido, no caso um Médico Veterinário.

Na figura 18 exibe a tela de Menu do Software CETEC, pois tem o objetivo de demonstrar a estrutura que foi definida no desenvolvimento desta tela.

Para o usuário efetuar um cadastro ou busca de Animal Bruto é necessário seguir os seguintes passos, CTEC – Animal Bruto – Cadastrar ou Buscar, pois seguindo estas etapas o usuário vai realizar o cadastro de Animal Bruto e para efetuar o cadastro ou uma busca de Cortes é necessário o usuário seguir as seguintes etapas, CTEC – Cortes – Cadastrar ou Buscar, com isto o usuário irá efetuar o cadastro e a busca de Cortes.



Figura 18: Tela de Menu do Software CETEC.

A figura 19 é representada a tela de cadastro do Animal Cortes Brutos, pois é a tela na qual o usuário vai inserir todos os dados referentes ao Animal Bruto, ou seja, estes dados são: Data do Cadastro, Carcaça Espécie (BOVINO ou SUINO), Quantidade/Unidade (número total de animas cadastrado), Peso do Animal, Ossos do Animal, Peso do Descarte do Animal, Peso de Cabeça e Fissura do Animal (tripa, fígado, vísceras).

SOLIGEE
Software Livre para Gestão Empresarial

Animal Cortes Brutos

Dados Animal Bruto

Cod.
6

Data de Cadastro (*):
01/10/2010

Carcaça Espécie (*):
Bovino

Quantidade/Unidade (*):

Peso do Animal em Kg (*):

Ossos do Animal em Kg (*):

Peso do Discarte do Animal em Kg (*):

Peso de Cabeça e Fissura do Animal em Kg (*):

Cadastrar

(*) = Campo obrigatório

LABS
Laboratório Avançado de Software

Figura 19: Tela de Cadastro de Animal Cortes Brutos.

A figura 20 ilustra como funciona o cadastro após o usuário realizar um cadastro, no qual o usuário necessita confirmar o cadastro clicando sobre o *OK* da *Pop-Up*.

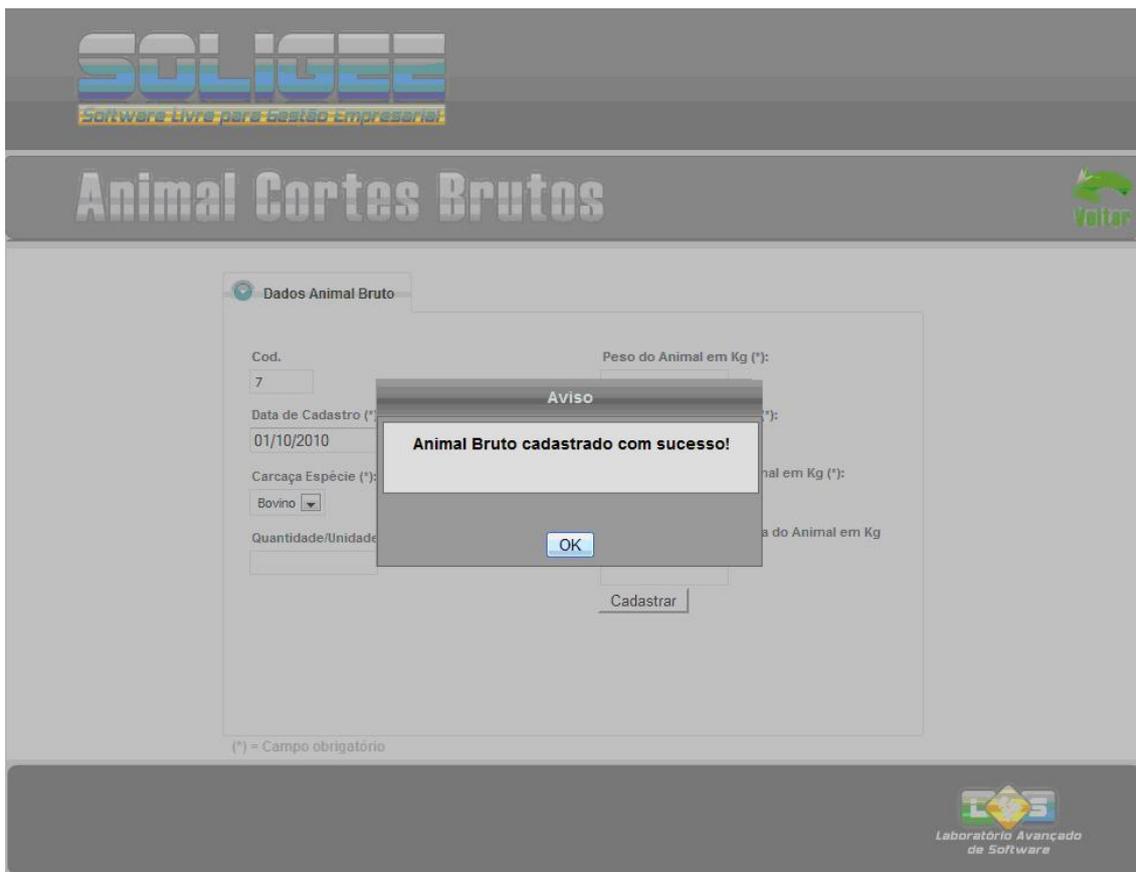


Figura 20: Pop-Up de confirmação de cadastro.

A figura 21 exibe o *Pop-Up* para que o usuário possa sair da tela de cadastro, pois é necessário o usuário clicar sobre o botão voltar para que essa *Pop-Up* de aviso seja ativada, após clicar sobre o voltar o usuário escolhe entre as opções sair ou não da tela de cadastro.

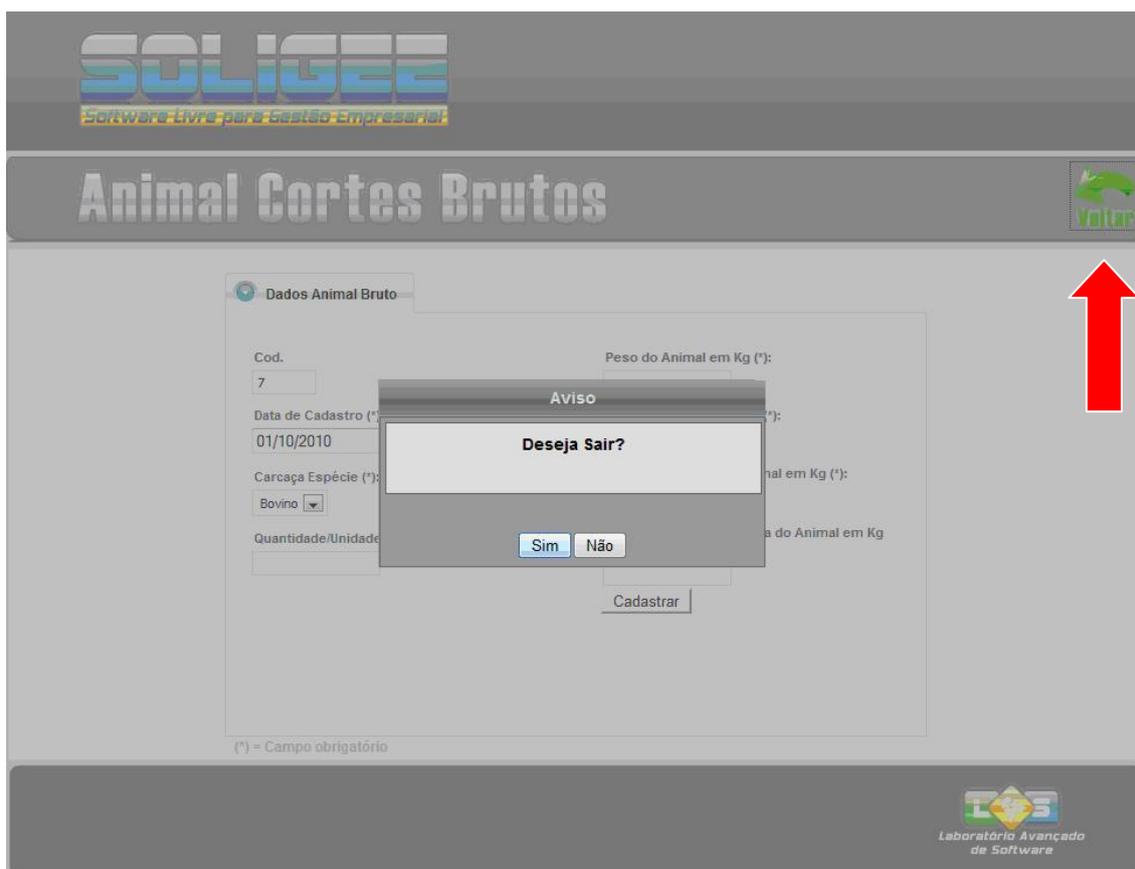


Figura 21: Botão Voltar e Pop-Up de aviso ao usuário.

A figura 22 demonstra a tela de cadastro de cortes, ou seja, Carne Processada e seus Derivados, pois carne processada é um corte do BOVINO ou SUINO (Picanha, Lombo, Costela) e os derivados são os produtos que podem ser gerados através dos cortes (Lingüiça Frescal, Bacon Defumado, Lingüiça Defumada).

SOLIGEE
Software Livre para Gestão Empresarial

Carne Processada e Derivados

Voltar

Cód.

Cadastro de Corte(*):

Total de Pacote:

Especie:

Tipo de Produto:

Cortes:

Preço do Kg:

Table_Column1	Table_Column2	Table_Column3
1	Column A	blue
2	Column B	red
3	Column C	green
4	Column D	yellow
5	Column E	white

LAS
Laboratório Avançado de Software

Figura 22: Tela de Cadastro de Cortes e Derivados.

5.2 GRÁFICOS

Nesta seção exibe comparações em forma de gráficos entre as variabilidades encontradas no desenvolvimento do CTEC e HV juntamente com o núcleo de artefatos SOLIGEE.

Como o núcleo de artefatos sofreu novas configurações o SOLIGEE, CTEC e HV possuem o mesmo número de características obrigatórias. Com isto vai ser analisado o número de características opcionais e alternativas que são características particulares de cada produto.

A figura 23 demonstra a quantidade de características que estão classificadas como opcionais, o resultado se obteve ao realizar uma análise das variabilidades no modelo de características do SOLIGEE, CTEC e HV.

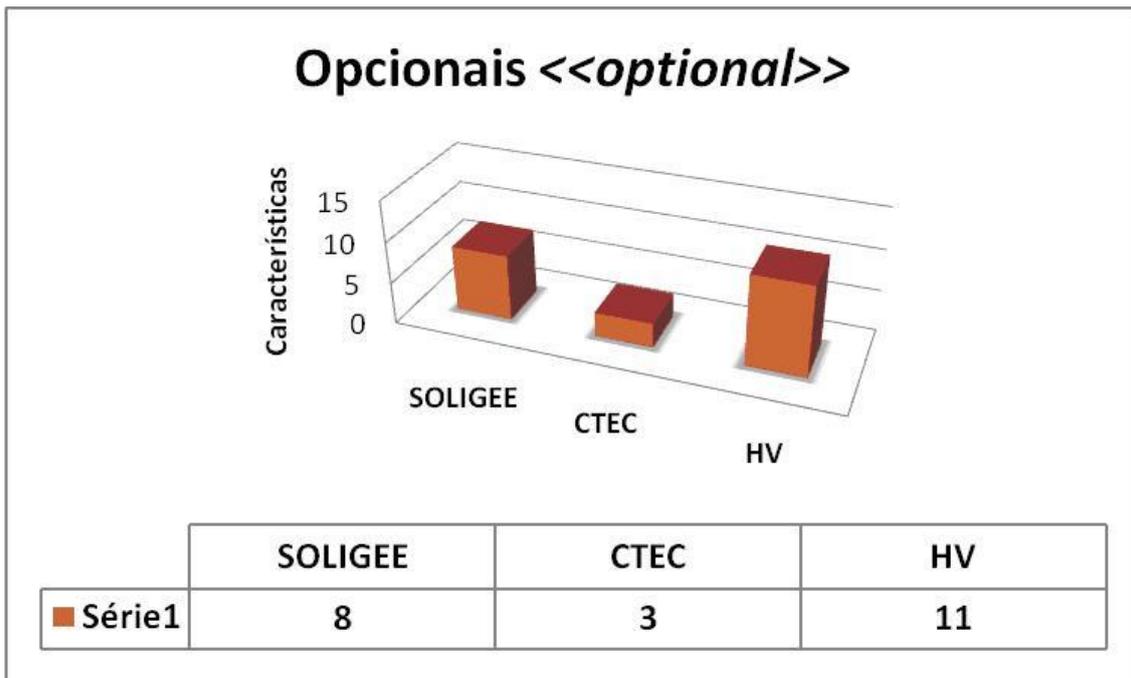


Figura 23: Gráfico de variabilidades – opcionais.

A figura 24 demonstra a quantidade de características que estão classificadas como alternativas, o resultado apresentado se obteve ao realizar uma análise das variabilidades no modelo de características do SOLIGEE, CTEC e HV.

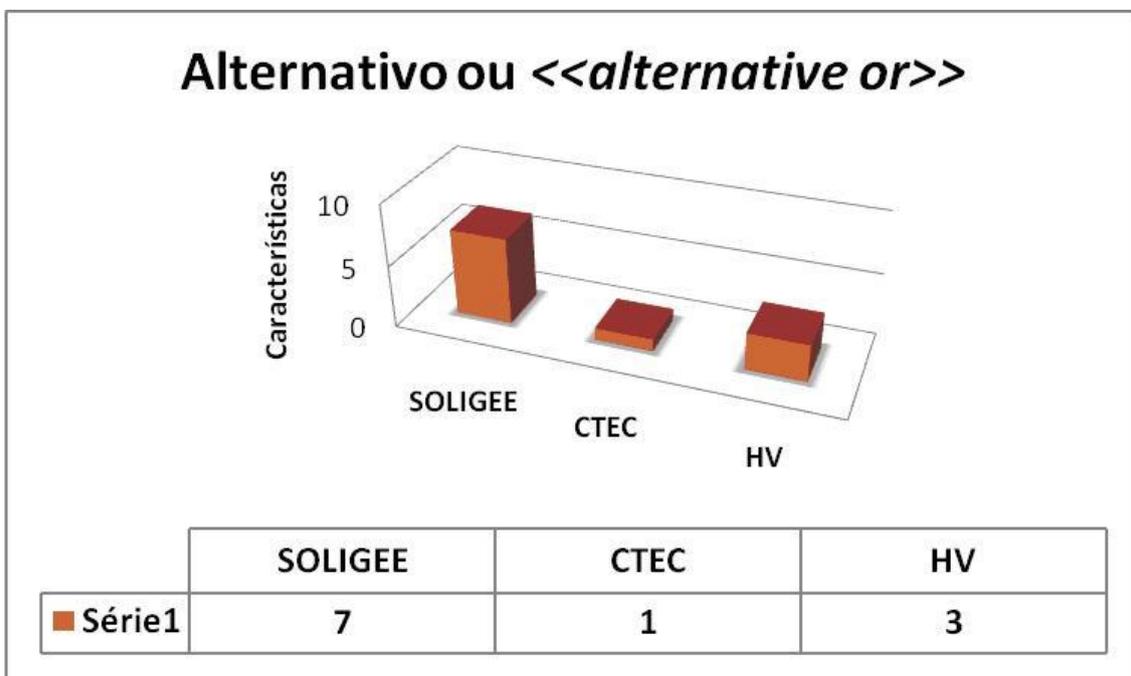


Figura 24: Gráfico de variabilidades – alternativo ou.

Após a análise das respectivas variabilidades pode-se perceber que após as configurações necessárias o núcleo de artefatos precisava de novas modificações, ou seja, a cada configuração o núcleo de artefatos se tornava ainda mais consolidado e maturo.

6 CONCLUSÃO

Linha de Produto de Software é vista como uma área muito promissora, pois oferece uma maneira sistemática, prática e planejada de reutilização de software. Muito desse sucesso deve-se ao fato da indústria enxergar no enfoque de LPS uma maneira de resolver problemas como *time-to-market*, redução de custo e tempo de produção, aumentando assim a sua produtividade.

O gerenciamento de variabilidades é uma das atividades mais importantes da LPS, pois através desse processo é possível a identificação dos pontos os quais um produto se diferencia em uma LPS.

Empresas que visam produzir em alta escala e menor tempo e ainda assim reduzir os custos, utilizando a abordagem de LPS em seu processo de desenvolvimento pode obter melhorias em seus processos de desenvolvimento.

Por meio deste estudo realizado sobre a abordagem de Linha de Produto de Software pode-se concluir que os resultados obtidos durante o processo de desenvolvimento de Software pode trazer melhorias no processo de desenvolvimento do produto. Foram constadas alterações necessárias no núcleo de artefatos, pois a partir da verificação de que ao inserir novas característica obrigatória <<*mandatory*>> eram necessárias alterações no núcleo de artefatos.

Embora esse estudo não esteja relacionado diretamente com o Projeto de Banco de Dados, o que podemos verificar é que com a inclusão de novas características, é fácil verificar que o Banco de Dados sofrerá impacto também, pois aumentou o número de características e atributos específicos no produto desenvolvido.

Como trabalhos futuros, realizar uma análise comparativa entre o produto do CTEC com o produto do Hospital Veterinário (HV), pois ambos foram gerados do núcleo de artefatos SOLIGEE e outro trabalho é utilizar o conceito de ontologias para engenharia de domínio.

REFERÊNCIAS BIBLIOGRÁFICAS

BRITO, S. I. Casos de uso <-<http://www.estig.ipbeja.pt/~issb/es/casosdeuso.pdf>
-> Acessado em 15/09/2010.

CASTRO, Y. C. Uma abordagem para o desenvolvimento de linha de produto orientado a aspectos. Dissertação de Mestrado. UEM-PR. Maringá. 2008.

CLEMENTS, P.; NORTHROP, L. Software product lines: practices and patterns. 1. ed. Boston: Addison-Wesley, 2001.

FERNANDES, C. P. UBIFEX: uma abordagem para modelagem de características de linha de produtos de software sensíveis ao contexto. Dissertação de Mestrado. UFRJ-RJ. Rio de Janeiro. 2009.

GOMAA, H. Designing Software Product Lines with UML: From Use Cases to Pattern-Based Software Architectures. Addison-Wesley, 2004.

INES – Instituto Nacional de Ciência e Tecnologia para Engenharia de Software. INES papers will be presented at DBKDA and ICN 2010 <-
<http://www.ines.org.br/?p=162>-> Acessado em 20/04/2010.

LEITE, P. J.; LAGUNA, A. M.; BAIXAULI, G, B. Análisis de Variabilidad con Modelos de Objetivos Universidad de Valladolid, PUC-RJ. Rio de Janeiro. 2004.

OLIVEIRA JUNIOR, E. A. de. Um processo de gerenciamento de variabilidade para linha de produto de software. Dissertação de Mestrado. UEM – PR. Maringá. 2005.

PRESSMAN, R. S. Engenharia de software. 5nd. ed. McGraw-Hill, 2002. SEI A framework for software product line practice. Versão 4.2. Software Engineering Institute, Carnegie Mellon University. <-<http://www.sei.cmu.edu/productlines/index.html>-> Acessado em 21/05/2010.

REINEHR, S. S. Linhas de Produto de Software: riscos e vantagens de sua Implantação. VI Simpósio Internacional de Melhoria de Processos de Software. São Paulo. 2004.

RUBIRA, F. C.; LOBO, C. A. Um Estudo para Implantação de Linha de Produto de Software baseada em Componentes. Universidade Estadual de Campinas. 2009.

SEI - SOFTWARE ENGINEERING INSTITUTE. A Framework for Software Product Line Practice, Version 5.0 <-
http://www.sei.cmu.edu/productlines/frame_report/index.html-> Acessado em 25/05/2010.

TANAKA, A. S. Rastreabilidade e Integração dos modelos de Casos de Uso com os outros diagramas da UML. Trabalho. Universidade Estadual de Londrina. 2009.

TEIXEIRA, M. L. Ligo: Uma linha de produtos de software para gerenciamento de igrejas cristãs. Escola Politécnica de Pernambuco. 2007.

TEIXEIRA, N. E. Flexibilização para representação de Características no ambiente odyssey. Projeto Final. UFRJ-RJ. Rio de Janeiro. 2008.

TRAVASSOS, H. G.; GIMENES, I. O Enfoque de Linha de Produto para Desenvolvimento de Software XXII Congresso da Sociedade Brasileira de Computação. JAI, 2002.

WEISS, D.; CHI TAU, R. L. Software product-line engineering: a family-based software development process. Boston: Addison Wesley, 1999.