

**BÁRBARA DE FÁTIMA SILVA**

**ANÁLISE COMPARATIVA DE ALGORITMOS DE  
TÉCNICA COMBINATÓRIA PARA TESTES EM  
LINHAS DE PRODUTOS DE SOFTWARE**

Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do curso de Sistemas de Informação para obtenção do título de Bacharel em Sistemas de Informação.

LAVRAS  
MINAS GERAIS - BRASIL  
2014

**BÁRBARA DE FÁTIMA SILVA**

**ANÁLISE COMPARATIVA DE ALGORITMOS DE  
TÉCNICA COMBINATÓRIA PARA TESTES EM  
LINHAS DE PRODUTOS DE SOFTWARE**

Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do curso de Sistemas de Informação para obtenção do título de Bacharel em Sistemas de Informação.

Área de Concentração:  
Engenharia de Software

Orientador:  
Prof. Dr. Heitor Augustus Xavier Costa

LAVRAS  
MINAS GERAIS - BRASIL  
2014

**Ficha Catalográfica preparada pela Divisão de Processo Técnico da Biblioteca  
Central da UFLA**

Silva, Bárbara de Fátima

Análise Comparativa de Algoritmos de Técnica Combinatória para Testes em Linhas de Produtos de Software / Bárbara de Fátima Silva. Lavras - Minas Gerais, 2014. 71p.

Monografia de Graduação - Universidade Federal de Lavras. Departamento de Ciência da Computação.

1. Linha de Produtos de Software. 2. Algoritmos de Técnica Combinatória. 3. Teste de Software. I. Silva, B. de F. II. Universidade Federal de Lavras. III. Análise Comparativa de Algoritmos de Técnica Combinatória para Testes em Linhas de Produtos de Software.

**BÁRBARA DE FÁTIMA SILVA**

**ANÁLISE COMPARATIVA DE ALGORITMOS DE  
TÉCNICA COMBINATÓRIA PARA TESTES EM  
LINHAS DE PRODUTOS DE SOFTWARE**

Trabalho de Conclusão de Curso de  
Graduação apresentado ao Colegiado do  
Curso de Bacharelado em Sistemas de  
Informação, para obtenção do título de  
Bacharel.

APROVADA em 28 de novembro de 2014.

Dr. André Pimenta Freire

Gustavo Andrade do Vale

MSc. Ramon Simões Abílio



Dr. Heitor Augustus Xavier Costa (Orientador)

LAVRAS-MG  
Novembro/2014

# AGRADECIMENTOS

Dedico este trabalho aos meus pais, Nasaré e Geraldo, e à minha irmã Isabel.

“Lembro-me do teu olhar, do teu sorriso, dos teus passos acompanhando o meu andar, da tua mão amparando-me quando eu caía, das tuas lágrimas lamentando quando eu perdia, mas me lembro principalmente do teu sorriso, do brilho de satisfação que percorria tua face, e lembro-me ainda do som das tuas palmas aclamando-me quando eu vencia”.  
(Joyce Maria Worschech Gabrielli - adaptado).

Agradeço a todos os meus irmãos que sempre contribuíram para a realização dos meus sonhos. Em especial agradeço ao meu irmão Reginaldo por ter me ensinado persistência, compaixão, paciência e que “querer é poder”.

Agradeço ao meu orientador, Heitor Augustus, que depositou em mim a confiança para realização deste trabalho e compartilhou comigo o seu conhecimento.

Finalmente, agradeço a todos os amigos que fiz em Lavras em especial, Patricia, Geraldo, Kamila e Vinicius.

# **Análise Comparativa de Algoritmos de Técnica Combinatória para Testes em Linhas de Produtos de Software**

## **RESUMO**

Linha de Produtos de Software (LPS) é uma abordagem que visa o desenvolvimento de uma família de produtos que possuem diferenças entre si. Essas diferenças são denominadas de comunalidade e variabilidade da LPS, e ambas podem ser representadas através do modelo de características. O aumento da variabilidade e o número de produtos tornam o teste de LPSs exaustivo e em alguns casos inviável. Gerar testes a partir de modelos de características, ou seja, gerar um subconjunto válido e representativo de todas as possíveis configurações de produtos, ainda é um desafio aberto. Dessa forma este trabalho descreve uma análise comparativa sobre algoritmos de teste combinatório, uma abordagem utilizada para gerar configurações de teste que cobrem todas as interações possíveis entre características. Essas técnicas foram aplicadas em seis LPS e os resultados foram avaliados por um conjunto de medidas específicas para linhas de produtos de software. Diante dos resultados da avaliação, foram destacados os principais desafios do teste combinatório e as principais vantagens dos algoritmos analisados. Um desafio consiste em selecionar um pequeno subconjunto de todas as configurações possíveis para o teste. Essa amostra deve cobrir características relevantes do modelo de características, enquanto permanece tão pequena quanto possível. Utilizando os algoritmos é possível selecionar um pequeno número de configurações em grandes modelos de características em tempo razoável.

**Palavras-chave:** Linha de Produto de Software, Teste de Software, Teste em Linhas de Produto de Software

## **Comparative Analysis of Algorithms of Combinatory Technique for Testing on Software Product Lines**

### **ABSTRACT**

Software Product Line (SPL) is an approach that aims to develop a family of products that have differences. These differences are called commonality and variability of SPL, and the feature model can represent both. The increased variability and the number of products make SPL test exhaustive and in some cases not feasible. Generating tests from models of features, i.e., generating a valid and representative test of all possible product configurations subset, is still an open challenge. Thus, this paper describes a comparative analysis of combinatorial testing algorithms, an approach used to generate test configurations that cover all possible interactions between features. These techniques were applied in six SPL and the results were evaluated by a set of specific measures for software product lines. Given the results of the evaluation, it is highlight the main challenges of combinatorial testing and the main advantages of the algorithms discussed. One challenge consists in selecting a small subset of all possible configurations for testing. This sample should cover relevant characteristics of the feature model, while staying as small as possible. Using the algorithms is possible to select a small number of settings in large models features in reasonable time.

**Keywords:** Software Product Line, Software Testing, Testing on Software Product Lines

# SUMÁRIO

## LISTA DE FIGURAS

## LISTA DE TABELAS

<b>1. INTRODUÇÃO .....</b>	<b>1</b>
1.1. Motivação.....	2
1.2. Objetivo.....	2
1.3. Metodologia de Desenvolvimento.....	3
1.3.1. Tipos de Pesquisa .....	3
1.3.2. Procedimentos Metodológicos .....	3
1.4. Estrutura do Trabalho .....	4
<b>2. LINHAS DE PRODUTOS DE SOFTWARE.....</b>	<b>5</b>
2.1. Considerações Iniciais .....	5
2.2. Definições .....	5
2.3. Processo de Desenvolvimento .....	6
2.4. Comunalidades e Variabilidades.....	9
2.5. Modelagem de Características .....	10
2.6. Abordagens de Construção .....	12
2.7. Vantagens e Desvantagens.....	13
2.8. Considerações Finais .....	16
<b>3. TESTE EM LINHAS DE PRODUTOS DE SOFTWARE.....</b>	<b>17</b>
3.1. Considerações Iniciais .....	17
3.2. Teste de Software.....	17
3.3. Teste em Linhas de Produto de Software .....	19
3.3.1. Teste Baseado em Modelos .....	21
3.3.2. Teste Combinatório .....	21
3.4. Algoritmos.....	24
3.4.1. ICPL .....	24
3.4.1.1. Algoritmo ICPL .....	25
3.4.1.2. Encontrando Características Válidas e Inválidas.....	27
3.4.1.3. Gerando uma Única Configuração.....	27
3.4.1.4. Identificando os $\tau$ -conjuntos Cobertos .....	28
3.4.1.5. Identificando os $\tau$ -conjuntos Inválidos .....	28
3.4.1.6. Paralelizando.....	29
3.4.2. Pacogen .....	30
3.5. Análise Comparativa .....	31
3.6. Considerações Finais .....	32
<b>4. MEDIDAS DE SOFTWARE .....</b>	<b>33</b>
4.1. Considerações Iniciais .....	33
4.2. Definição .....	33
4.3. Importância e Benefícios .....	34

4.4. Tipos de Medidas.....	35
4.5. Medidas de Teste para Linhas de Produtos de Software .....	35
4.6. Considerações Finais .....	36
<b>5. AVALIAÇÃO DA EXECUÇÃO DOS ALGORITMOS.....</b>	<b>37</b>
5.1. Considerações Iniciais .....	37
5.2. SPLOT .....	37
5.3. Escolha das LPSs .....	37
5.4. Análise dos Resultados .....	40
5.4.1. Execução dos Algoritmos.....	40
5.4.2. Performance e Tamanho do Conjunto de Teste Gerado .....	43
5.4.3. Ocorrência das t-tuplas e Frequência das t-tuplas .....	44
5.5. Considerações Finais .....	46
<b>6. TRABALHOS RELACIONADOS.....</b>	<b>47</b>
<b>7. CONSIDERAÇÕES FINAIS.....</b>	<b>49</b>
7.1. Conclusões.....	49
7.2. Contribuições.....	50
7.3. Limitações.....	50
7.4. Trabalhos Futuros .....	51
<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>52</b>
<b>ANEXO A - Modelos de Características .....</b>	<b>57</b>
<b>ANEXO B - Matrizes de Cobertura Geradas pelos Algoritmos .....</b>	<b>62</b>
<b>ANEXO C - Tabelas de Frequência e Ocorrência .....</b>	<b>69</b>

# LISTA DE FIGURAS

<b>Figura 2-1 - Atividades Essenciais (Fonte: [SEI, 2007]) .....</b>	<b>6</b>
<b>Figura 2-2 - Ciclo de Desenvolvimento de uma LPS (Fonte: [Van der Linden et al., 2007]) .....</b>	<b>7</b>
<b>Figura 2-3 - Modelo de Características da LPS Graph Product Line (Adaptada: SPLOT) .....</b>	<b>12</b>
<b>Figura 2-4 - Abordagens de Desenvolvimento em LPS (Fonte: [Krueger, 2002]) .....</b>	<b>13</b>
<b>Figura 3-1 - Modelo V (Fonte: [Tevanlinna et al., 2004]) .....</b>	<b>20</b>
<b>Figura 3-2 - Modelo de Características LPS Celular (Adaptada: SPLOT) .....</b>	<b>23</b>
<b>Figura 5-1 - Modelo de Características da LPS Documentation Generation (Adaptada: SPLOT) .....</b>	<b>39</b>
<b>Figura A-1 - Modelo de Características da LPS Inventory (Adaptado: SPLOT) .....</b>	<b>57</b>
<b>Figura A-2 - Modelo de Características da LPS TankWar (Adaptado: SPLOT) .....</b>	<b>58</b>
<b>Figura A-3 - Modelo de Características da LPS Fire Alarm (Adaptado: SPLOT) .....</b>	<b>59</b>
<b>Figura A-4 - Modelo de Características da LPS Arcade Game (Adaptado: SPLOT) .....</b>	<b>60</b>
<b>Figura A-5 - Modelo de Características da LPS Sienna Van (Adaptado: SPLOT) .....</b>	<b>61</b>

## LISTA DE TABELAS

Tabela 3-1 - Matriz de Cobertura da LPS Celular.....	23
Tabela 3-2 - Análise Comparativa (Adaptada de [Lochau et al., 2012]).....	32
Tabela 5-1 - Caracterização das LPSs Selecionadas.....	40
Tabela 5-2 - Matriz de Cobertura da LPS Documentation Gereration Gerada por ICPL .....	41
Tabela 5-3 - Matriz de Cobertura da LPS Documentation Gereration Gerada por Pacogen.....	42
Tabela 5-4 - Aplicação das Medidas nas LPSs.....	43
Tabela 5-5 - Frequência e Ocorrência das t-Tuplas da LPS Documentation Generation .....	45
Tabela B-1 - Matriz de Cobertura da LPS Inventory Gerada por ICPL .....	62
Tabela B-2 - Matriz de Cobertura da LPS TankWar Gerada por ICPL .....	62
Tabela B-3 - Matriz de Cobertura da LPS Fire Alarm Gerada por ICPL .....	63
Tabela B-4 - Matriz de Cobertura da LPS Arcade Gerada por ICPL .....	63
Tabela B-5 - Matriz de Cobertura da LPS Sienna Van Gerada por ICPL .....	64
Tabela B-6 - Matriz de Cobertura LPS Inventory Gerada por Pacogen.....	65
Tabela B-7 - Matriz de Cobertura LPS TankWar Gerada por Pacogen.....	66
Tabela B-8 - Matriz de Cobertura da LPS Fire Alarm Gerada por Pacogen.....	66
Tabela B-9 - Matriz de Cobertura LPS Arcade Game Gerada por Pacogen .....	67
Tabela B-10 - Matriz de Cobertura da LPS Sienna Van Gerada por Pacogen.....	68
Tabela C-1 - Frequência e Ocorrência das t-Tuplas da LPS Inventory.....	69
Tabela C-2 - Frequência e Ocorrência das t-Tuplas da LPS TankWar .....	69
Tabela C-3 - Frequência e Ocorrência das t-Tuplas da LPS Fire Alarm .....	70
Tabela C-4 - Frequência e Ocorrência das t-Tuplas da LPS Arcade Game .....	70
Tabela C-5 - Frequência e Ocorrência das t-Tuplas da LPS Sienna Van .....	71

# 1. INTRODUÇÃO

A abordagem de Linha de Produtos de Software (LPS) consiste na proposta de construção sistemática de software em larga escala baseada em família de produtos, incentivando e guiando organizações no desenvolvimento de novos produtos a partir de artefatos reutilizáveis [Brownsword; Clements, 1996; Linden et al., 2007]. Essa abordagem atua sobre um segmento de mercado ou domínio específico que se refere a uma área de conhecimento e pode ser representada utilizando modelos de características - artefatos desenvolvidos na fase inicial da construção da LPS.

Esses modelos auxiliam no entendimento da LPS como um todo e são utilizados como ponto de partida para a instanciação de novos produtos [Lee; Muthig, 2006]. Esses produtos são compostos pela combinação de características comuns e variáveis, possibilitando aumento na flexibilidade e adaptação às necessidades dos clientes. Características são atributos de um sistema de software que afetam diretamente os usuários finais. Uma característica pode ser um requisito específico ou uma seleção entre os requisitos opcionais e alternativos ou estar relacionada a certas características do produto (por exemplo, funcionalidade, usabilidade e desempenho) ou a características de implementação (por exemplo, tamanho, plataforma de execução e compatibilidade com certos padrões) [Kang et al., 1990].

As organizações utilizam LPS para melhorar a qualidade, aumentar a produtividade, e reduzir o time-to-market [McGregor, 2010]. Para desenvolver produtos de qualidade, é necessário que eles possuam o mínimo de erros, de defeitos e de falhas; logo, esses produtos necessitam ser testados para verificar se possuem qualidade. Testar uma LPS consiste em verificar seus principais ativos, os seus produtos e a interação entre eles. Esses testes são realizados nos processos de engenharia de domínio e engenharia de aplicação [Engström; Runeson, 2010]. Na engenharia de domínio, algumas atividades de teste estão relacionadas ao desenvolvimento de casos de teste e execução de testes com o objetivo de avaliar a qualidade do núcleo de artefatos a ser instanciado na fase de engenharia de aplicação. As duas principais atividades incluem o desenvolvimento de artefatos de teste que podem ser reutilizados de forma eficiente na engenharia de aplicação e a aplicação de testes no núcleo de artefatos criados na engenharia de domínio.

Por causa da quantidade de variabilidades das características e de produtos que podem ser gerados a partir de ativos reutilizáveis, a execução de um teste que tenha resultados totalmente confiáveis torna-se uma tarefa difícil para ser realizada e com alto custo [Neto et al., 2011]; no entanto, testes são necessários e essenciais.

### **1.1. Motivação**

Organizações estão tomando a decisão estratégica de adotar LPS para a produção de sistemas de software em larga escala [Bosch, 2005]. Essa decisão é tomada, muitas vezes, para que a organização possa alcançar vantagem competitiva dentro do seu segmento de mercado. Para obter esse benefício, são necessárias abordagens diferentes das utilizadas no paradigma de desenvolvimento de sistemas únicos para as etapas do desenvolvimento dos produtos, incluindo a etapa de testes que desempenha papel importante nesse esforço estratégico.

Testes são importantes para identificar defeitos e assegurar que os produtos tenham confiabilidade e que sejam capazes de desempenhar suas funções de acordo com as especificações predefinidas nos requisitos [Oliveira et al., 2007]. Esta prática comum no desenvolvimento de sistemas únicos possui desafios no contexto de LPS; alguns desses desafios estão relacionados com a variabilidade e a quantidade de testes requeridos, pois a amplitude da variabilidade dos artefatos, incluindo os artefatos de teste, impacta diretamente a quantidade de recursos necessários para os testes adequados. Os casos de teste e os dados de teste devem ser reutilizados, mas eles não são fáceis de gerenciar.

Para que esses desafios não comprometam o sucesso da implantação de LPSs, diferentes técnicas e algoritmos de teste de software têm que se adaptar ao contexto dessa nova abordagem para gerar resultados que comprovem a qualidade dos produtos desenvolvidos.

### **1.2. Objetivo**

Neste trabalho, o objetivo é apresentar uma análise comparativa de algoritmos de técnicas combinatórias de teste em LPS. Para atingir esse objetivo, foi necessário alcançar os seguintes objetivos específicos:

- Realizar estudo sobre algoritmos e técnicas de teste para aprimorar os conhecimentos;
- Selecionar e obter informações sobre os algoritmos a serem analisados;

- Identificar e definir critérios de comparação;
- Realizar busca na literatura para identificar medidas que possam ser utilizadas para obter os pontos fortes e fracos dos algoritmos.

### 1.3. Metodologia de Desenvolvimento

A metodologia de pesquisa é um conjunto de métodos, técnicas e procedimentos, cuja finalidade é viabilizar a execução da pesquisa que tem como resultado um novo produto, processo ou conhecimento [Jung, 2009].

#### 1.3.1. Tipos de Pesquisa

Inicialmente, quanto à natureza, este trabalho pode ser classificado como **pesquisa aplicada**, pois foi realizado sobre um problema que possui finalidade de aplicação. Quanto aos objetivos, é classificado como **pesquisa exploratória**, pois foram avaliados dois algoritmos de técnica combinatória para realizar teste em LPSs. Quanto às abordagens, pode ser classificado como **pesquisa qualitativa**, pois expõe um conjunto de referencial teórico, e **pesquisa quantitativa**, pois utiliza medidas para avaliar as técnicas de teste combinatório para LPS. Quanto aos procedimentos, pode ser classificado como **estudo de caso**, pois os algoritmos foram utilizados em um conjunto de LPS.

#### 1.3.2. Procedimentos Metodológicos

Este trabalho foi iniciado por um levantamento bibliográfico na Internet e nas bibliotecas digitais IEEE, Scopus e Elsevier ScienceDirect por artigos relacionados ao tema. Inicialmente, foi realizado um estudo sobre os conceitos, a história, a importância e o estado da arte da abordagem de LPS. Em seguida, foram estudados artigos cujo tema central foi teste em LPSs para identificar artigos que descrevem como é realizada a atividade de teste e quais técnicas são utilizadas nessa atividade. Com esse estudo, foram escolhidos para análise comparativa, os algoritmos ICPL e Pacogen de técnica combinatória. Esses algoritmos foram escolhidos, pois objetivam produzir o mesmo tipo de solução, ou seja, um conjunto de casos de teste relevante utilizando modelos de características de LPSs.

Para complementar a análise comparativa, foi realizada identificação de medidas de software em artigos cujo tema central foi teste combinatório em LPSs, com a finalidade de avaliar os algoritmos. Essas medidas foram brevemente apresentadas e contextualizadas.

#### **1.4. Estrutura do Trabalho**

O restante deste trabalho está organizado da seguinte forma.

Uma visão geral sobre linha de produtos de software é apresentada no Capítulo 2.

Definições, conceitos e técnicas de teste em LPS são explicados no Capítulo 3.

Medidas de software são brevemente descritas no Capítulo 4.

A avaliação dos algoritmos de técnica combinatória é apresentada no Capítulo 5.

Alguns trabalhos relacionados estão resumidos no Capítulo 6.

Conclusões, contribuições, limitações e sugestões de trabalhos futuros são tratadas no Capítulo 7.

## **2. LINHAS DE PRODUTOS DE SOFTWARE**

### **2.1. Considerações Iniciais**

O comportamento dos consumidores de sistemas de software modificou-se. Eles não se satisfazem com uma única versão de um produto que seja igual para todos e desejam produtos personalizados e com várias funções, mas não desejam que esse fato comprometa o custo final e o prazo de entrega dos produtos. Assim como diversos mercados, o mercado de software torna-se cada vez mais competitivo, lançando novos produtos a curto espaço de tempo. Entretanto, desenvolvimento de software personalizado possui alto custo [Díaz et al., 2011].

Para atender a demanda desses mercados, a abordagem de Linha de Produtos de Software (LPS) pode ser utilizada, pois tem base conceitual no reúso de software que possibilita a produção em larga escala de diversos produtos customizados e viabiliza o desenvolvimento de famílias de produtos ao invés de desenvolver produtos individuais. Uma LPS é um conjunto de sistemas que possuem produtos com domínio e base de artefatos em comum. Os produtos são definidos utilizando seleção de características. Essa seleção determina as semelhanças e as diferenças entre os produtos. As semelhanças permitem que produtos sejam desenvolvidos com comportamento padrão definido pela modelagem de características comuns e as variabilidades permitem que o comportamento dos produtos seja variável. Portanto, a modelagem de características variáveis deve permitir a adaptação e o desenvolvimento de produtos específicos [Weiss; Lai, 1999].

Este capítulo está organizado da seguinte forma. Definições de LPS são apresentadas na Seção 2.2. Os processos de desenvolvimento, abordando a Engenharia de Domínio e a Engenharia de Aplicação, são tratados na Seção 2.3. Aspectos importantes sobre comunalidade e variabilidade são discutidos na Seção 2.4. A modelagem de características é explicada na Seção 2.5. As abordagens de construção de uma LPS são apresentadas na Seção 2.6. Vantagens e desvantagens em adotar LPS são consideradas na Seção 2.7.

### **2.2. Definições**

Na década de setenta, o conceito de família de produtos, que, inicialmente, considerava apenas a variabilidade de requisitos não funcionais entre os produtos, foi introduzido na literatura [Parnas, 1976]. O termo "linha de produto de software" originou-

se no conceito de engenharia de domínio introduzido na década de noventa, tendo como uma das primeiras contribuições a descrição do método FODA (Feature-Oriented Domain Analysis) [Kang et al., 1990].

LPS é um conjunto de sistemas de software que compartilham um conjunto gerenciável comum de características que satisfazem as necessidades específicas de um segmento de mercado particular e são desenvolvidas a partir de um conjunto de ativos base comum de modo planejado [Clements; Northrop, 2007]. De acordo com o framework proposto pelo SEI (Software Engineering Institute), uma LPS é composta por três atividades essenciais, altamente iterativas e relacionadas (Figura 2-1) [SEI, 2007]:

- **Desenvolvimento dos ativos básicos** (core assets). Infraestrutura essencial para o reuso, base para o desenvolvimento do produto em uma LPS. Exemplos de ativos base são arquitetura, componentes, especificação de requisitos, modelo do domínio e plano de testes. Essa atividade é conhecida como Engenharia de Domínio;
- **Desenvolvimento dos produtos** a partir dos ativos base. Essa atividade é chamada de Engenharia de Aplicação;
- **Gerenciamento** em nível organizacional e técnico da LPS. Essa atividade está relacionada com a supervisão da construção dos ativos base e do desenvolvimento do sistema, com a finalidade de garantir que as equipes que constroem os ativos base e os sistemas estejam envolvidos no processo e na evolução da LPS. A gestão organizacional deve conduzir, controlar e assegurar a plena utilização dos ativos base [Long, 2002].



Figura 2-1 - Atividades Essenciais (Fonte: [SEI, 2007])

### 2.3. Processo de Desenvolvimento

A abordagem LPS é separada em dois processos [Clements; Northrop, 2007]: i) Engenharia de Domínio; e ii) Engenharia de Aplicação. O ciclo de desenvolvimento de

uma LPS é apresentado na Figura 2-2. A Engenharia de Domínio é o processo responsável por estabelecer a plataforma reutilizável da LPS, sendo composta por artefatos de software (por exemplo, requisitos, projeto, implementação e testes). A rastreabilidade entre esses artefatos atua como facilitadora para o reúso consistente dentro da LPS. Os objetivos do processo de engenharia de domínio são definir e identificar a comunalidade e a variabilidade da LPS, definir o escopo da LPS e definir e construir artefatos reutilizáveis que atendam a variabilidade desejada na LPS.

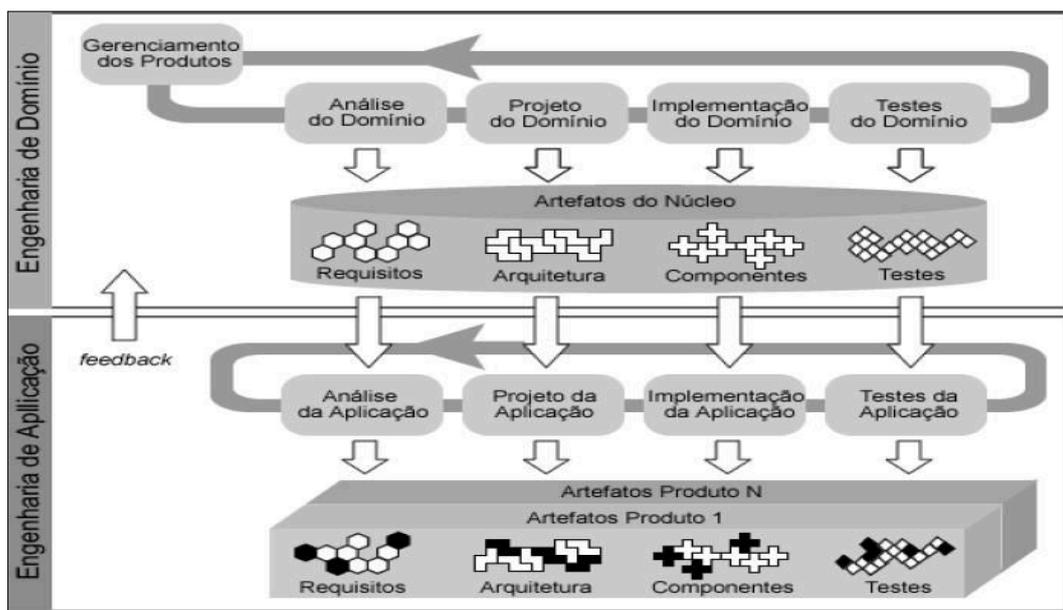


Figura 2-2 - Ciclo de Desenvolvimento de uma LPS (Fonte: [Van der Linden et al., 2007])

Os objetivos da Engenharia de Domínio são realizados por cinco subprocessos para detalhar e refinar a variabilidade determinada no subprocesso anterior e fornecer feedback sobre a viabilidade determinada no subprocesso anterior. Os subprocessos são [Linden et al., 2007]:

- **Gerenciamento dos produtos.** Define os produtos presentes na LPS, identifica comunalidades e variabilidades entre eles e os analisa economicamente;
- **Análise do domínio.** Analisa os requisitos para os vários produtos da LPS e constrói o modelo inicial de variabilidade;
- **Projeto do domínio.** Desenvolve a arquitetura da LPS;
- **Implementação do domínio.** Detalha o design e implementa os componentes de software reutilizáveis;
- **Teste do domínio.** Testa os componentes e gera ativos de teste que podem ser reutilizados nos testes de aplicação.

Engenharia de Aplicação é o processo responsável por construir produtos específicos utilizando a reutilização de artefatos de domínio e explorando a variabilidade da LPS. Os objetivos do processo de engenharia de aplicação são [Linden et al., 2007]:

- Atingir nível alto de reutilização dos artefatos de domínio na definição e no desenvolvimento de produtos;
- Explorar a comunalidade e a variabilidade da LPS no desenvolvimento de produtos;
- Documentar os artefatos de aplicação (os requisitos de aplicação, a arquitetura, os componentes e os testes) e relacioná-los com os artefatos de domínio;
- Associar a variabilidade dos requisitos, da arquitetura, dos componentes e dos casos de teste, de acordo com as necessidades do produto;
- Avaliar possíveis impactos das diferenças entre os requisitos da Engenharia de Aplicação e da Engenharia de Domínio.

A Engenharia de Aplicação possui quatro subprocessos e cada um utiliza artefatos de domínio e produz artefatos de aplicação. Os subprocessos são [Linden et al., 2007]:

- **Análise da aplicação.** Identifica os requisitos do produto, avaliando as comunalidades e as variabilidades existentes na arquitetura;
- **Projeto da aplicação.** Deriva uma instância da arquitetura, de acordo com os requisitos identificados, construindo adaptações específicas para o produto;
- **Implementação da aplicação.** Desenvolve a implementação final do produto, inclui reúso e configuração de componentes existentes ou construção de novos componentes de acordo com os requisitos do produto;
- **Teste da aplicação.** Testa o produto, validando-o de acordo com seus requisitos.

A vantagem da divisão entre Engenharia de Domínio e Engenharia de Aplicação é ser possível construir uma plataforma robusta e aplicações específicas para o cliente em um curto espaço de tempo. Para que esses processos sejam eficazes, ambos devem interagir de maneira benéfica entre si. Por exemplo, a plataforma deve ser construída com o objetivo de ser utilizada no desenvolvimento de produtos e o desenvolvimento de produtos deve ser auxiliado utilizando a plataforma [Linden et al., 2007]. Além disso, a separação em dois processos simplifica o gerenciamento da variabilidade. A Engenharia de Domínio é responsável por garantir que a variabilidade disponível seja adequada para a produção de produtos, utilizando mecanismos comuns para derivar um produto específico.

A plataforma comum da LPS é definida com a quantidade certa de artefatos reutilizáveis. A Engenharia de Aplicação reutiliza essa plataforma e distribui a variabilidade para os diferentes produtos, conforme exigido nos requisitos [Linden et al., 2007].

## 2.4. Comunalidades e Variabilidades

Dentre os ativos comuns criados para uma LPS, existem os ativos utilizados por todos os produtos da LPS e os ativos utilizados apenas em alguns produtos. São as chamadas comunalidades e variabilidades. Comunalidade denota características que fazem parte de cada produto, exatamente da mesma forma, contribuindo para a uniformização da LPS. A variabilidade de uma LPS é modelada para permitir o desenvolvimento de produtos customizados, reutilizando artefatos predefinidos. Por isso, a variabilidade distingue diferentes produtos da LPS [Bosch, 2005].

A variabilidade refere-se à capacidade de um artefato ser configurado, personalizado, estendido ou modificado para o seu uso em um contexto específico [Linden et al., 2007; Van Gurp et al., 2001]. Ela deve ser definida, representada, explorada, implementada e desenvolvida durante o ciclo de vida da LPS, pois é utilizada para a construção de diferentes produtos. Ela é representada utilizando pontos de variação e seus respectivos variantes. Ponto de variação é o local do sistema em que artefatos variáveis são relacionados com o restante do sistema e variantes são as possíveis instâncias em um ponto de variação. Dessa forma, a variabilidade é representada por pontos de variação, cada um associado a um conjunto de variantes e cada variante corresponde a uma alternativa formulada para instanciar uma determinada variabilidade [Van Gurp et al., 2001]. Os pontos de variação podem ser apresentados em vários níveis de abstração [Krueger, 2002]:

- **Descrição da arquitetura.** Tipicamente, o sistema é descrito utilizando uma combinação de documentos do seu projeto, linguagens de descrição de arquitetura e documentação textual;
- **Documentação por diagramas.** Nesse nível, o sistema pode ser descrito utilizando notações UML (Unified Modeling Language) e modelos de características;
- **Código fonte.** Uma descrição completa na forma de código fonte é criada;
- **Código Compilado.** O código fonte é convertido para o código compilado utilizando um compilador. Os resultados dessa compilação podem ser influenciados com diretivas de pré-processamento;

- **Código de Execução.** Durante a execução, o sistema construído é iniciado e configurado. Ao contrário das representações anteriores, o sistema de execução é dinâmico e muda constantemente.

Gerenciamento de variabilidade é um processo que apoia a Engenharia de Domínio e a Engenharia de Aplicação, pois disponibiliza recursos para a manipulação sistemática das partes variáveis dos sistemas. Nesse processo, a variabilidade pode ser identificada e projetada na Engenharia de Domínio e fornecer apoio para desenvolver diferentes produtos na Engenharia de Aplicação, [Schmid; John, 2004]. Gerenciar variabilidades é uma tarefa complexa que necessita ser apoiada por abordagens adequadas, técnicas e ferramentas, pois requer saber exatamente como identificar, representar, delimitar e implementar a variabilidade [Bosh et al., 2002].

Documentar e gerir a variabilidade estão entre as principais características que diferenciam a engenharia de LPS e o desenvolvimento de sistemas únicos, pois causa impacto em artefatos e nos estágios do ciclo de vida do desenvolvimento da LPS; por exemplo, a variabilidade pode atingir o desempenho, a demanda de recursos e a interoperabilidade do sistema [Becker, 2003]. Variantes podem requerer ou excluir outras variantes, resultando em interdependência entre elas, o que interfere na consistência e na eficiência do gerenciamento da variabilidade. Essas e outras alterações/adaptações a serem gerenciadas podem afetar a qualidade e o comportamento do sistema. Assim, o sucesso da LPS está diretamente interligado com bom gerenciamento da variabilidade [Becker, 2003].

## **2.5. Modelagem de Características**

Modelos de características representam as propriedades comuns e variáveis da LPS e suas interdependências, organizando-as em um modelo coerente. Esses modelos têm o objetivo de fornecer uma representação abstrata, concisa e explícita da variabilidade [Czarnecki; Eisenecker, 2000]. O conceito de característica (feature) foi originalmente apresentado pelo método Feature Oriented Domain Analysis (FODA). Característica é uma propriedade do sistema visível ao usuário final, que pode ser definida como uma unidade lógica de comportamento especificada por um conjunto de requisitos funcionais e de qualidade. Além disso, seu comportamento deve ser relevante para um ou vários stakeholders de um produto [Kang et al., 1990].

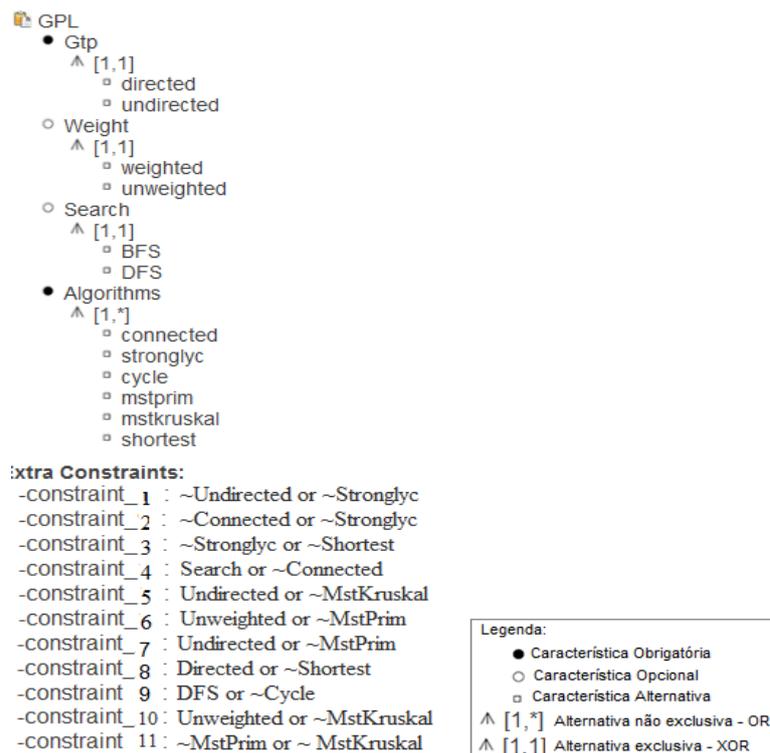
As diferenças entre uma característica e outras abstrações conceituais, por exemplo, funções, objetos e aspectos, estão relacionadas com a forma de descrever um sistema. Funções, objetos e aspectos são utilizados para especificar os detalhes internos de um sistema. Métodos estruturados especificam detalhes internos de um sistema em termos de funções, que representam abstrações processuais; métodos orientados a objetos especificam a estrutura e o comportamento de um sistema em termos de objetos, abstrações, chaves identificáveis ou entidades. Em Programação Orientada a Aspectos, funções, objetos e aspectos são abstrações conceituais identificáveis a partir de pontos de vista internos [Lee et al., 2002]. Por outro lado, as características são as particularidades visíveis externamente que se diferenciam entre os produtos. Portanto, modelagens de características devem concentrar-se na identificação de características externas dos produtos em termos de aspectos comuns e variáveis, ao invés de descrever os detalhes dos produtos [Kang et al., 2002; Lee et al., 2002].

Na modelagem de características, as características do produto são identificadas e classificadas em termos de capacidades, de tecnologias de domínio, de técnicas de implementação e de ambientes operacionais [Kang et al., 1998]. Capacidades são características visíveis ao usuário que podem ser identificadas como serviços distintos (por exemplo, transferência de chamada no domínio de telefonia), operações e características não funcionais (por exemplo, desempenho). Tecnologias de domínio (por exemplo, métodos de navegação no domínio aviãoico) representam a forma de implementação de serviços ou operações. Técnicas de implementação (por exemplo, mecanismos de sincronização) são funções genéricas ou técnicas utilizadas para implementar serviços, operações e funções de domínio. Ambientes operacionais (por exemplo, sistemas operacionais) representam ambientes em que as aplicações são utilizadas.

Características comuns entre diferentes produtos são modeladas como obrigatórias, características diferentes entre eles podem ser opcionais ou alternativas. As características opcionais representam características selecionáveis para os produtos de determinado domínio e características alternativas indicam que há mais de uma característica que pode ser selecionada para um produto [Kang et al., 1990].

Um modelo de características é representado no formato de árvore (Figura 2-3), contendo três tipos de relacionamento. O relacionamento de composição é utilizado se

houver uma relação integral entre uma característica e suas subcaracterísticas. O relacionamento de generalização/especialização para os casos em que as características são generalizações de subcaracterísticas. A relação "implementado - por" é utilizada quando uma característica de uma técnica de implementação é necessária para implementar outra característica. Regras de composição complementam o modelo com as relações de dependência mútua e de exclusão mútua, utilizadas para restringir a seleção de recursos opcionais ou alternativos para a configuração de um produto [Kang et al., 1990].



**Figura 2-3 - Modelo de Características da LPS Graph Product Line (Adaptada: SPLOT<sup>1</sup>)**

## 2.6. Abordagens de Construção

Existem três abordagens principais de desenvolvimento em LPS (Figura 2-4) [Krueger, 2002]:

- **Proativa:** desenvolvimento dos artefatos da LPS, cujo ponto inicial é a análise, o projeto e a implementação de domínio e, posteriormente, a engenharia de aplicação. Essa estratégia busca desenvolver a maior quantidade de produtos existentes de um domínio específico;
- **Extrativa:** o desenvolvimento da LPS ocorre a partir de um conjunto de sistemas de software existentes, extraíndo suas características comuns e variáveis;

<sup>1</sup> <http://www.splot-research.org/>

- **Reativa:** a LPS é desenvolvida incrementalmente e é ampliada quando há demanda de novos requisitos ou produtos.

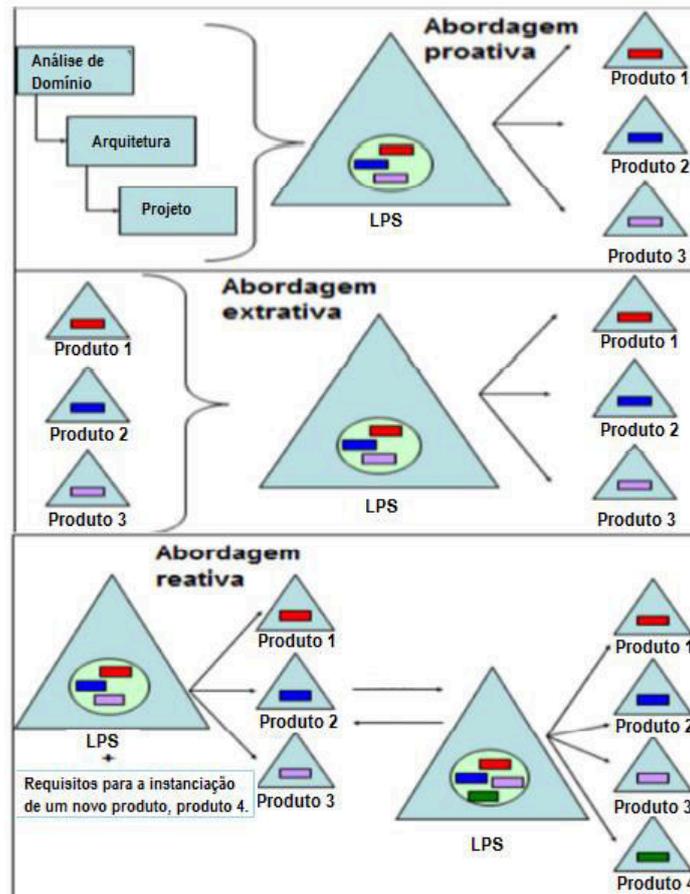


Figura 2-4 - Abordagens de Desenvolvimento em LPS (Fonte: [Krueger, 2002])

## 2.7. Vantagens e Desvantagens

Existem vantagens atribuídas às LPSs, especialmente no ambiente acadêmico. Essas vantagens podem ser classificadas em benefícios tangíveis e benefícios intangíveis. Benefícios tangíveis podem ser medidos diretamente, por exemplo, redução de defeitos, e benefícios intangíveis não podem ser medidos diretamente, por exemplo, satisfação do cliente. Os benefícios tangíveis são [Cohen, 2002]:

- **Lucratividade:** o repositório de ativos permite que a organização produza produtos voltados a um segmento específico de mercado. O benefício é observado no aumento da participação de mercado e da lucratividade;
- **Qualidade:** a redução na quantidade de defeitos relatados é comum em sistemas desenvolvidos em LPS. Além disso, qualidade pode ser medida em termos de redução do tempo de correções e da redução do efeito ripple (geração de novos defeitos a partir de correções executadas);

- **Desempenho dos sistemas de software:** a utilização de ativos aumenta o desempenho em relação ao desenvolvimento tradicional, especialmente com o aumento da maturidade da LPS, o que faz com que os ativos estejam cada vez mais otimizados;
- **Produtividade:** a equipe de desenvolvimento pode ser reduzida, o custo total de desenvolvimento é reduzido, o cronograma é reduzido (maior velocidade de lançamento de novos produtos);
- **Tempo de integração:** o tempo de integração no desenvolvimento incremental é completado mais rápido do que em porções sem ativos ou um aplicativo;
- **Volume de Código:** a quantidade de objetos projetados para subsistemas utilizando ativos base é menor do que os estimados para sistemas individuais, resultando em redução semelhante no tamanho real de código fonte.

Os benefícios intangíveis são:

- **Desgaste de profissionais:** menor desgaste dos profissionais, o que resulta em redução do turnover de membros da equipe;
- **Aceitabilidade dos desenvolvedores:** após 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 foi realizado (desenvolvimento dos ativos de software); assim, eles podem concentrar-se em atividades mais interessantes, como aperfeiçoamento/inação de elementos específicos;
- **Satisfação do cliente:** reutilização de ativos reduz riscos, aumentando a previsibilidade da entrega e diminuindo a taxa de defeitos.

No entanto, existem dificuldades atribuídas ao processo de adoção de LPS pelas organizações [Cohen, 2002; Clements et al., 2002]. Algumas dessas dificuldades são:

- **Falta de um líder comprometido:** para o processo de adoção, sugere-se a alocação de um líder, uma pessoa que acredita nos princípios da abordagem de LPS, que supervisiona o processo e motiva as pessoas. Caso essa pessoa não receba a autoridade necessária, não acredite no modelo ou simplesmente não exista, será difícil manter o foco e a confiança durante o processo e há um grande risco de desistência;

- **Falta de compromisso da gerência:** a gerência deve estar convencida da viabilidade e das vantagens que o projeto vai trazer para a organização. É comum que projetos de implantação de LPS comecem com força total, mas, ao longo do tempo, por pressão do mercado e por falta de foco, tornam-se projetos coadjuvantes e caminham para o esquecimento, gerando perda de recursos;
- **Abordagem inadequada:** a LPS deve ser um projeto estratégico para atingir objetivos corporativos. Apesar de seus objetivos variarem de empresa para empresa, eles estão baseados no ganho de produtividade, de custo e de qualidade e na exploração das similaridades de produtos de uma família. Caso os produtos planejados para a LPS não possuam similaridades suficientes para garantir a viabilidade do modelo, a LPS poderá não trazer o retorno planejado;
- **Falta de compromisso da equipe:** assim como a gerência, a equipe desenvolvedora deve manter um compromisso com os princípios e com o processo de desenvolvimento da LPS. Uma equipe que não acredita na LPS impossibilita o processo. Uma maneira de evitar esse tipo de problema é envolver membros da equipe nas atividades de elaboração da LPS fazendo-os sentir parte do projeto;
- **Interação insuficiente entre as equipes:** a iniciativa de uma LPS transcende os limites tradicionais da organização. O processo envolve diversas equipes como gerência, marketing e engenharia. Uma falta de interação e de colaboração entre essas equipes pode impedir a obtenção dos objetivos planejados;
- **Padronizações desapropriadas:** a institucionalização prematura e impensada de padrões, muitas vezes obsoletos, pode limitar as opções de tecnologia da LPS, prejudicando seu escopo;
- **Adaptação insuficiente:** as práticas organizacionais (padronizadas ou não) devem permitir adaptação e personalização voltadas para as características da LPS. A falta de adaptabilidade ou adaptação mal pensada pode subotimizar o desempenho das equipes ou gerar desvios imprevistos das atividades envolvidas no processo da LPS;
- **Evolução da abordagem:** esse fator poderia ser chamado de melhoria contínua, pois, caso o processo da LPS não seja revisado e atualizado periodicamente, suas práticas tornam-se obsoletas e inapropriadas às novas necessidades que surgirão;
- **Falha de disseminação:** o líder da LPS deve tomar a responsabilidade de desenvolver e de distribuir a documentação em nível e tipo apropriados para a organização, treinar os envolvidos e apoiar o processo no que for necessário.

## **2.8. Considerações Finais**

Uma LPS consiste em um conjunto de sistemas de software que possuem um conjunto de funções em comum, que satisfazem as necessidades de um segmento de mercado ou missão em particular, e são desenvolvidos utilizando a mesma plataforma base.

As atividades essenciais de uma LPS são constituídas por Engenharia de Domínio, Engenharia de Aplicação e Gerenciamento organizacional e técnico. A Engenharia de Domínio engloba o desenvolvimento dos ativos base, a Engenharia de Aplicação engloba o desenvolvimento de produtos que utilizem os ativos base. Essas atividades envolvem os atributos mais importantes sobre o desenvolvimento de uma LPS, por exemplo, a variabilidade, que define as particularidades de uma LPS e deve ser gerenciada.

Muitas organizações adotam LPS desenvolvendo, inicialmente, os artefatos básicos (abordagens pró-ativas). Outras organizações iniciam com apenas um ou uma pequena quantidade de produtos e desenvolvem a LPS (artefatos básicos e aplicações futuras) a partir deles (abordagens extrativas). No entanto, independentemente da abordagem de construção adotada, a LPS gera benefícios para a organização e pode gerar algumas dificuldades.

## **3. TESTE EM LINHAS DE PRODUTOS DE SOFTWARE**

### **3.1. Considerações Iniciais**

A construção de software é uma atividade complexa cuja execução está diretamente relacionada com as características e as dimensões do sistema a ser desenvolvido. Dessa forma, está sujeita a diversos e diferentes tipos de problemas que resultam na obtenção de um produto final diferente do esperado [Delamaro et al., 2007]. A etapa de testes é vital no processo de desenvolvimento de software, em que busca identificar potenciais defeitos e assegurar que os produtos tenham confiabilidade, exatidão e qualidade. As organizações que utilizam a abordagem de LPS devem definir conjuntos de atividades abrangentes que possam validar a veracidade do que foi construído e verificar se o produto correto foi construído [McGregor, 2001].

Este capítulo está organizado da seguinte forma. Conceitos relacionados ao processo de teste em sistemas únicos e sua importância são abordados na Seção 3.2. Como processo de teste é realizado na abordagem de LPS é discutido na Seção 3.3. Algoritmos de teste combinatório são apresentados na Seção 3.4. Uma análise dos algoritmos estudados é mostrada na Seção 3.5.

### **3.2. Teste de Software**

Existem várias definições para a atividade de teste, desde definições intuitivas a definições formais. Essas definições conduzem a uma ideia geral de que teste de software é o processo de executar o software de maneira controlada com o objetivo de avaliar se ele comporta-se conforme o especificado [Hetzl et al., 1998; Myers et al., 1979; Beizer et al., 1995]. A atividade de teste deve deliberadamente fazer com que o sistema opere de forma errada em condições normais e anormais a fim de determinar se erros acontecem quando supostamente não deveriam ocorrer no sistema [Shepard et al., 2001]

Teste de software é um dos aspectos mais importantes de garantia de qualidade de software, que pode informar a qualidade do sistema em termos quantitativos e qualitativos às partes interessadas. É um termo genérico que engloba diferentes atividades, desde o teste de uma pequena parte do código realizado pelo desenvolvedor à validação do usuário de um complexo sistema de software. Em vários estágios, os casos de teste podem ser planejados em direção a diferentes objetivos, por exemplo, estimar a conformidade de uma

especificação padrão, avaliar a robustez para entradas maliciosas, medir atributos (p. e. desempenho e usabilidade) e estimar a confiabilidade operacional [Oliveira et al., 2007].

Por causa da diversidade de fatores que podem colaborar para a ocorrência de erros, a atividade de teste torna-se extremamente complexa. Dessa forma, ela é dividida em fases com objetivos distintos [Pressman, 2009]:

- **Teste de unidade** tem como foco as menores unidades de um programa, por exemplo, funções, procedimentos, métodos ou classes. Tem como objetivo identificar erros diretamente relacionados a algoritmos incorretos ou mal interpretados, estruturas de dados incorretas, ou simples erros de programação;
- **Teste de integração** deve ser executado após a finalização do teste de unidade, enfatizando a construção da estrutura do sistema. À medida que as diversas partes do software são colocadas para trabalhar juntas, torna-se necessário verificar se a interação entre elas funciona de maneira adequada;
- **Teste de sistema** tem por finalidade verificar se as funções foram implementadas corretamente de acordo com a especificação do sistema. Aspectos de correção, de completude e de coerência devem ser explorados, bem como os requisitos não funcionais, tais como, segurança e desempenho. De forma geral, a maioria das organizações prefere utilizar uma equipe independente para realizar o teste de sistema.

O processo de teste de software envolve a escolha de uma ou mais técnicas de teste. Essas técnicas podem ser categorizadas em [Wallmüller, 2001; Perry; Kaiser, 1990]:

- **Técnicas estáticas** coletam informações sobre o sistema sem executá-lo com o objetivo de encontrar falhas na implementação. Métodos típicos desse tipo de técnica são auditorias, revisões e inspeções, executados no código fonte do sistema em teste. A análise estática de código fonte pode ser realizada por ferramentas que examinam informações sintáticas (complexidade e gráficos de dependência), informações semânticas (anomalias) e informação lexicais (comprimento de procedimentos). Na verificação formal, características são definidas e verificadas em função dos requisitos especificados para o sistema. Esse tipo de verificação raramente é utilizado, pois a extração das características pode demandar tempo;
- **Técnicas dinâmicas** envolvem a geração de casos de teste, a execução de casos de teste, a comparação entre o resultado da execução do teste e o resultado esperado

definido pelo comportamento do sistema. Essas técnicas podem ser classificadas em caixa preta, caixa branca ou uma combinação de ambas (caixa cinza).

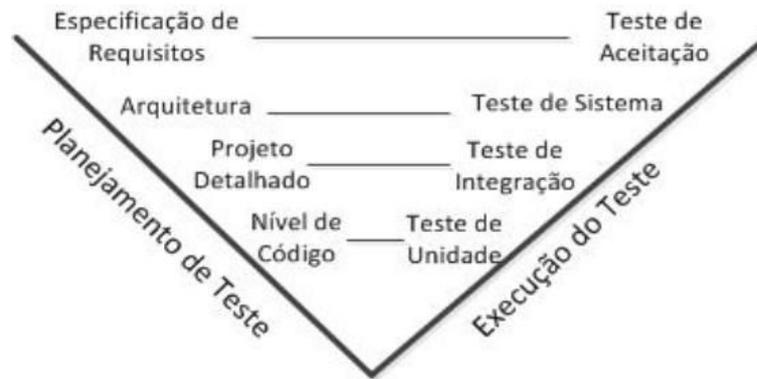
### 3.3. Teste em Linhas de Produto de Software

Teste em LPS consiste em verificar os principais ativos da LPS, os seus produtos e a interação entre eles. O processo de teste em LPS é diferente do processo de teste no desenvolvimento de produtos únicos. As atividades de teste são executadas em conjunto com as atividades de desenvolvimento de software. O processo de desenvolvimento produz artefatos verificados no processo de testes. Um processo de teste produz resultados dos testes e identifica e relata os defeitos para fazer correções nos artefatos [McGregor, 2001].

Testes em LPS são realizados nos processos de Engenharia de Domínio e Engenharia de Aplicação [Czerwonka, 2011; Deutsch, 1978; Engström; Runeson, 2010; Grave et al., 2001]. Na Engenharia de Domínio, algumas atividades de teste estão relacionadas ao desenvolvimento de casos de teste e execução de testes com o objetivo de avaliar a qualidade dos ativos base, que, mais tarde, são instanciados na fase de Engenharia de Aplicação. As duas principais atividades incluem o desenvolvimento de artefatos de teste que possam ser reutilizados de forma eficiente na engenharia de aplicação e a aplicação de testes nos ativos base criados na Engenharia de Domínio. Na Engenharia de Aplicação, as atividades de teste estão relacionadas à seleção e à instanciação de artefatos para construir casos de teste específicos. São alguns dos tipos de testes que podem ser executados:

- **Teste de sistema.** O objetivo é assegurar que o produto final está de acordo com o especificado;
- **Teste de aceitação.** Consiste na execução pelo cliente de um roteiro de teste.

Da forma semelhante ao teste de sistemas únicos que segue o modelo tradicional de desenvolvimento de software, teste em LPS requer planejamento cuidadoso e processo bem definido, pois testes ocorrem nas fases do processo de desenvolvimento [McGregor, 2001]. Um modelo utilizado para representar as diferentes fases do processo de desenvolvimento é o modelo de processo de desenvolvimento do software em V (Figura 3-1). Artefatos produzidos na Engenharia de Domínio e na Engenharia de Aplicação podem ser testados segundo os níveis desse modelo [Tevanlinna et al., 2004].



**Figura 3-1 - Modelo V (Fonte: [Tevanlinna et al., 2004])**

O lado esquerdo inclui os níveis de abstração comuns no software. Nesse lado, testes são projetados no projeto e na implementação do software. Os modelos de projeto podem ser validados pela utilização de inspeções guiadas. O lado direito do modelo une as fases de teste aos níveis de abstração da implementação. Nesse lado, os testes projetados são executados. Testes em LPS podem ser categorizados em três melhores práticas [Tevanlinna et al., 2004; Reuys et al., 2006]:

- **Técnicas contra a filosofia LPS.** Essas técnicas contradizem a filosofia de reutilização adotada pelas LPSs. Cada produto é testado individualmente não considerando a divisão entre Engenharia de Domínio e Engenharia de Aplicação. Essa prática é conhecida como teste de produto a produto, mas, considerando a quantidade de produtos que podem ser construídos a partir de uma LPS, torna-se inviável;
- **Técnicas de reutilização.** Essas técnicas utilizam testes de regressão para testar os produtos de forma incremental ou reutilizam testes de domínio durante os testes de aplicação. Em testes baseados em modelos, a reutilização de testes de domínio é popular;
- **Heurísticas de subconjuntos.** Essa abordagem objetiva reduzir o esforço de testes extraindo um subconjunto de combinações de características ou produtos; assim, esse subconjunto é testado representando a LPS. Há duas metodologias: i) métodos que geram um subconjunto para testar a LPS; e ii) técnicas que utilizam teste combinatório que geram um subconjunto de produtos em que as interações entre as características ocorrem pelo menos uma vez.

Existem alguns desafios ao testar uma LPS [Engström; Runeson, 2010]. Um deles está relacionado com a quantidade de testes requeridos. De forma a testar completamente uma LPS, as possibilidades de uso de cada componente dos ativos base e cada

variabilidade deveriam ser testados, mas a variabilidade torna o teste exaustivo e inviável. Outro desafio é saber como balancear o esforço de teste entre os componentes reutilizáveis e as instâncias da LPS, pois se torna vital identificar componentes a serem testados na Engenharia de Domínio e componentes a serem testados na Engenharia de Aplicação.

### **3.3.1. Teste Baseado em Modelos**

A estrutura e o comportamento do sistema podem ser explicitamente descritos utilizando modelos baseados em abordagens de modelagem. Um modelo de teste representa o comportamento, os requisitos e a estrutura de um sistema em teste e é utilizado para gerar casos de teste [Philipps, 2004]. Teste Baseado em Modelo (TBM) gera especificações de teste e casos de teste tendo como ponto inicial um ou mais modelos formais ou semiformais de um sistema. Exemplos desses modelos são máquinas de estado, diagramas de atividade e sequência e modelos de características. Esses modelos são complementados com informações adicionais antes que possam ser utilizados para gerar especificações de teste [Pretschner, 2004].

A geração de casos de teste é realizada utilizando a seleção de caminhos no modelo; logo, o comportamento do sistema é a soma dos caminhos possíveis no modelo. Algoritmos para a seleção de caminhos, verificação de modelos, resolução de restrições e algoritmos que realizam busca em grafos são alguns métodos para TBM [Weißleder, 2010]. TBM é uma abordagem de teste dinâmico e a classificação entre teste caixa preta ou caixa branca depende da granularidade do modelo de teste. Se o modelo de teste é baseado nos requisitos ou na especificação do sistema, ele pode ser interpretado como teste caixa preta. Se o modelo de teste está relacionado com a aplicação ou é utilizado para gerar códigos para aplicação, ele pode ser interpretado como teste caixa branca.

Modelos de teste devem incluir informações suficientes para a geração automática de casos de teste. Além disso, eles devem ser simples para facilitar o seu desenvolvimento e a sua manutenção [Pretschner; Philipps, 2004; Robinson, 2000].

### **3.3.2. Teste Combinatório**

Teste combinatório é um método de teste caixa preta para diminuir o esforço de teste, reduzindo a quantidade de casos de teste. Para provar a correção de um programa, ele precisa ser testado com as combinações possíveis de valores de parâmetros de entrada

[Beizer, 1990]. No entanto, por causa da complexidade e da quantidade de produtos das LPSs, testar as combinações possíveis de valores de parâmetros de entrada é impraticável. Por exemplo, considerando um software com cinco parâmetros de entrada diferentes que podem ser inicializados de dez maneiras diferentes, exigiria  $10^5 = 100.000$  casos de teste diferentes para ser validado. No teste combinatório, determinadas combinações de valores de parâmetros são utilizadas como entrada.

O método pairwise é uma das aplicações de teste combinatório que se baseia na observação da maioria das falhas serem causadas por interações entre uma pequena quantidade de fatores ou parâmetros. No contexto de LPS, as comunalidades e as variabilidades são representadas por características que podem ser interpretadas como parâmetros no processo de engenharia [McGregor, 2011]. Pairwise ou T-wise é um conjunto de restrições aplicadas a um conjunto de variáveis matematicamente definidas em matrizes denominadas matrizes de cobertura (CA - Coveraging Arrays) [Phadke, 1995].

Uma CA é uma matriz com  $N$  linhas e  $K$  colunas (fatores). Os elementos das colunas são escolhidos a partir de um conjunto  $v$  de símbolos (valores). A propriedade fundamental de uma CA é, para qualquer escolha de  $t$  colunas, os conjuntos possíveis de  $t$  símbolos ( $t$ -conjuntos) aparecerem em pelo menos uma linha; os  $t$ -conjuntos são classificados como cobertos. Essa matriz de força  $t$ ,  $k$  fatores,  $v$  símbolos por fator e tamanho  $N$ , denotada por  $CA(N; t, k, v)$ , é uma matriz  $N \times K$  com símbolos  $v$  tal que, em cada submatriz  $t \times N$ , cada  $t$ -tupla em  $v^t$  é coberta pelo menos uma vez [Kuhn et al., 2004].

Para calcular essas matrizes, a tecnologia de resolução dos solucionadores SAT pode ser utilizada [Torlak; Jackson, 2007; Mahajan; Fu, 2004]. Uma fórmula booleana  $\varphi$  é satisfatível (SAT) caso exista ao menos uma atribuição de valores para suas variáveis que tornem  $\varphi$  verdadeira; caso contrário,  $\varphi$  não é satisfatível. A satisfabilidade booleana pertence a classe de problemas NP-completo, significando que para  $n$  variáveis existem  $2^n$  diferentes atribuições [Cook, 1971].

A matriz de cobertura da LPS Celular (Figura 3-2) é apresentada na Tabela 3-1. Nessa tabela, há sete produtos a serem testados para assegurar que a interação entre as suas características funcionem corretamente. Cada linha da tabela representa uma característica e cada coluna um produto. Para identificar que uma característica está incluída em um

produto, o símbolo 'X' é colocado na célula correspondente; por outro lado, caso uma característica não esteja presente em um produto é colocado o símbolo '-'. Algumas características estão presentes em todos os produtos, pois elas são as principais características (por exemplo, Celular). Alguns pares não são cobertos, uma vez que são pares inválidos segundo o critério de adequação do teste pairwise.

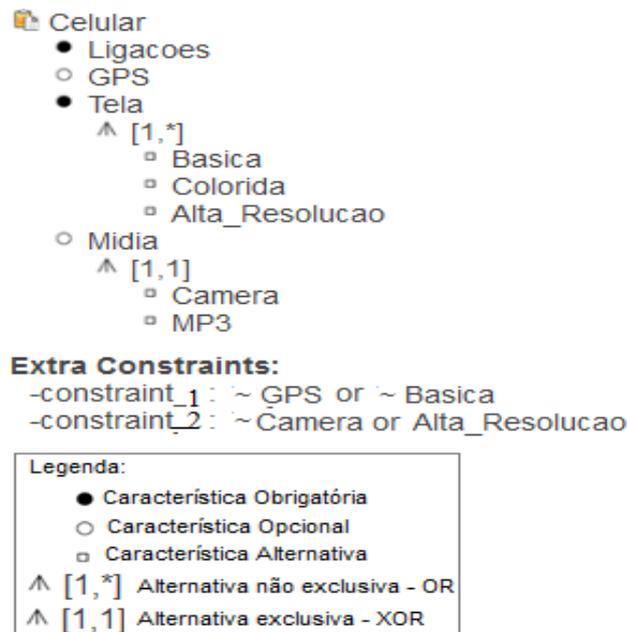


Figura 3-2 - Modelo de Características LPS Celular (Adaptada: SPLOT<sup>2</sup>)

Tabela 3-1 - Matriz de Cobertura da LPS Celular

Produtos	1	2	3	4	5	6	7
Alta Resolução	X	X					X
Básica				X	X		
Câmera		X					
Celular	X	X	X	X	X	X	X
Colorida			X			X	
GPS		X	X				
Ligações	X	X	X	X	X	X	X
Mídia		X			X	X	X
MP3		X			X	X	
Tela	X	X	X	X	X	X	X

Modelos de características são artefatos importantes para gerar conjuntos de casos de teste utilizando o método pairwise. Para tal finalidade, alguns critérios devem ser respeitados [Oster et al., 2010]:

- **Casos de teste:** um caso de teste deve referenciar-se a um produto válido da LPS e é formado por uma configuração válida do modelo de características;

<sup>2</sup> <http://www.splot-research.org/>

- **Válida/inválida  $t$ -tupla:** uma  $t$ -tupla ( $t$  é um número natural inteiro que representa a quantidade de características presentes na tupla) é válida (inválida) se for possível (impossível) derivar um produto que contém o par ( $t$ -tupla), desde que as restrições do modelo de características sejam satisfeitas. Por exemplo, um par inválido é a combinação entre uma característica opcional e uma característica não obrigatória;
- **Critério de adequação de teste:** para determinar se um conjunto de testes é capaz de cobrir o modelo de característica, considera-se o  $T$ -wise critério de adequação, sendo que cada  $t$ -tupla válida deve estar presente em pelo menos um caso de teste.

Existem três principais fases na aplicação de testes combinatórios para LPSs. Na primeira fase, o modelo de característica deve ser construído; na segunda fase, o subconjunto de casos de teste deve ser gerado a partir do modelo de características para uma força  $t$  de cobertura; e, na terceira fase, após a construção de cada produto, uma técnica de teste de sistema deve ser selecionada e aplicada [Kuhn et al., 2004].

### 3.4. Algoritmos

Nesta seção, são brevemente descritos dois algoritmos utilizados na segunda fase dos testes combinatórios.

#### 3.4.1. ICPL

ICPL (Covering array generation algorithm for Product Lines) [Johansen et al., 2012] é um algoritmo especializado para gerar conjuntos de teste  $t$ -wise para LPS, adaptado do algoritmo de Chvátal's utilizado para solucionar problemas nos conjuntos das matrizes de cobertura. Ele funciona essencialmente da mesma forma e pelas mesmas razões que o algoritmo de Chvátal's. No entanto, ICPL evita redundâncias, aprende com os resultados intermediários e permite paralelizar as etapas de processamento de dados independentes.

Em sua execução o algoritmo gera um conjunto denominado  $T_t$  que contém os  $t$ -conjuntos válidos que necessitam ser cobertos. Em seguida, o primeiro conjunto  $f_s$  de características que representa um produto é produzido utilizando a seleção de um subconjunto de  $t$ -conjuntos em  $T_t$  que constituem um produto válido segundo o modelo de características. O conjunto  $f_s$  é adicionado ao conjunto de testes  $C_t$  que inicialmente está vazio. ICPL começa a gerar produtos a serem adicionados ao conjunto de testes  $C_t$  até

o conjunto  $T_t$  estar vazio, ou seja, até os  $t$ -conjuntos válidos cobertos por, pelo menos, um produto serem removidos de  $T_t$ . Para a explicação do funcionamento do algoritmo ICPL, alguns conceitos necessitam ser compreendidos:

- Uma atribuição  $a$  é um par  $(f, \text{incluído})$ , sendo  $f$  a representação de uma característica presente no modelo de características da LPS e  $\text{incluído}$  um booleano que especifica se a característica  $f$  está ou não incluída no par;
- A configuração  $C$  é um conjunto de atribuições em que as características do modelo de características possuem uma atribuição;
- O  $t$ -conjunto é um conjunto de  $t$  atribuições, ou seja,  $t = \{a_1, a_2, \dots\}$ ;
- O conjunto dos  $t$ -conjuntos é denotado por  $T_t$ , por exemplo,  $T_1, T_2, \dots$ ;
- O conjunto dos  $t$ -conjuntos inválidos é denotado por  $I_t$ , por exemplo,  $I_1, I_2, \dots$ ;
- O conjunto dos  $t$ -conjuntos válidos é denotado por  $U_t = T_t \setminus I_t$ ;
- Uma matriz de cobertura  $\tau$  é um conjunto de configurações  $C\tau$ , em que os  $t$ -conjuntos válidos e os  $t$ -conjuntos contidos em  $U_t$  são um subconjunto ou o mesmo conjunto de pelo menos uma das configurações.

O modelo de características é armazenado em um objeto cujos métodos são:

- `size()`: o tamanho do modelo de características é a quantidade de características que ele possui.  $|FM|$  é uma notação alternativa para o tamanho de um modelo de característica;
- `genT( $\tau$ )`: retorna o conjunto  $T_t$  das combinações de atribuições das características válidas e inválidas;
- `satisfy( $c$ )`: retorna uma configuração válida do modelo de características, dadas as atribuições em  $c$ . Embora não seja uma exigência, esse método retorna uma configuração cuja tendência é ter somente as características necessárias. Esse comportamento é padrão do SAT4J utilizado na implementação desse método;
- `is_satisfiable( $c$ )`: retorna verdadeiro se o modelo de características é satisfatório dadas as atribuições em  $c$ .

#### 3.4.1.1. Algoritmo ICPL

A entrada do algoritmo ICPL (**Algoritmo 1**) é um modelo de características  $FM$  e um parâmetro  $\tau$  que indica a cobertura dos pares  $t$ -wise e a sua saída é a matriz de cobertura

para  $t$  ( $C_t$ ). O conjunto dos  $t$ -conjuntos inválidos  $I_t$  também é retornado. Nas linhas 2 a 4, é representada uma recursão em que o conjunto  $I_t$  dos  $t$ -conjuntos inválidos é identificado. Na linha 2, é identificado o conjunto  $U_t$  dos  $t$ -conjuntos válidos armazenado em  $T_t$  na linha 3. Assim que os conjuntos  $I_t$  forem encontrados, `invalidRemoved` é definido como `true` na linha 4. Uma matriz de cobertura  $t-1$  é um subconjunto (ou o mesmo conjunto) de uma matriz de cobertura  $t$ ; dessa maneira, pode-se dizer que  $CA(FM, t-1) \subseteq CA(FM, t)$  [Fouché et al., 2009]. Assim, antes de cobrir a força  $t$ , o algoritmo cobre os valores de  $1 \leq n < t$  e, antes de gerar o conjunto  $C_t$ , ele conhece  $C_{t-1}$  e  $I_{t-1}$ . Isto é realizado nas linhas 6 a 8.

<b>Algoritmo 1</b> ICPL(FM,t) : (C <sub>t</sub> , I <sub>t</sub> )	
1:	<b>if</b> t = 1 <b>then</b>
2:	(C <sub>t</sub> , I <sub>t</sub> ) ← genCompleto <sub>t</sub> (FM)
3:	T <sub>t</sub> ← FM.genT(1) \ I <sub>t</sub>
4:	invalidRemoved ← true
5:	<b>else</b>
6:	(C <sub>t</sub> , I <sub>t-1</sub> ) ← ICPL(FM, t - 1)
7:	(T <sub>t</sub> , I <sub>t</sub> ) ← generateTsets(FM.getT(t), I <sub>t-1</sub> , C <sub>t</sub> )
8:	invalidRemoved ← false
9:	<b>end if</b>
10:	<b>while</b> T <sub>t</sub> ≠ 0 <b>do</b>
11:	C ← genConfiguration(FM, T <sub>t</sub> )
12:	C <sub>t</sub> ← C <sub>t</sub> ∪ {C}
13:	CO ← getCovered(FM, C, T <sub>t</sub> )
14:	T <sub>t</sub> ← T <sub>t</sub> \ CO
15:	<b>if</b> (¬invalidRemoved) ∧ (⌊log <sub>10</sub>  CO ⌋ ≤ ⌊log <sub>10</sub>  FM ⌋) <b>then</b>
16:	(T <sub>t</sub> , TempI) ← genInvalid(FM, T <sub>t</sub> )
17:	I <sub>t</sub> ← I <sub>t</sub> ∪ TempI
18:	invalidRemoved ← true
19:	<b>end if</b>
20:	<b>end while</b>
21:	<b>return</b> (C <sub>t</sub> , I <sub>t</sub> )

As linhas 10 a 20 estão em um laço executado até os  $t$ -conjuntos sejam classificados em válidos ou inválidos. Nas linhas 11, 13, 16 e 17, estão representações de subalgoritmos (explicados nas próximas seções). Na linha 11, é gerada uma única configuração de atribuições armazenada em  $C$ ; na linha 13, são encontrados os  $t$ -conjuntos cobertos; nas linhas 16 e 17, são encontrados os  $t$ -conjuntos inválidos; e, na linha 15, os  $t$ -conjuntos inválidos são removidos. Esse ponto foi determinado empiricamente para ser igual ao ponto em que a ordem de magnitude dos  $t$ -conjuntos cobertos for a mesma ordem de magnitude do tamanho do modelo de características, quanto  $\lfloor \log_{10}|CO| \rfloor \leq \lfloor \log_{10}|FM| \rfloor$ . Por exemplo, considerando um modelo de características com 500 características, os  $t$ -

conjuntos inválidos são removidos quando a quantidade de  $\tau$ -conjuntos cobertos for igual a 999 ou menos.

### 3.4.1.2. Encontrando Características Válidas e Inválidas

Um passo no algoritmo ICPL é encontrar as características válidas (fundamentais) e inválidas (inoperantes). As características fundamentais são aquelas que devem sempre ser incluídas em uma configuração e as inoperantes são aquelas que não podem ser incluídas. Esse passo é realizado definindo cada característica como incluída ou excluída; logo, se o modelo de característica é insatisfatível. Isso é realizado com o **Algoritmo 2** cujos parâmetros são o modelo de características FM e um conjunto  $T_1$ . Ele verifica cada atribuição classificando-a como válida ou inválida e as inválidas são adicionadas ao  $\tau$ -conjunto  $I_1$ .

Algoritmo 2 getInvalidAssignments (FM, $T_1$ ) : $I_1$	
1:	$I_1 \leftarrow 0$
2:	<b>for</b> each t-set e in $T_1$ <b>do</b>
3:	<b>if</b> $\neg$ FM.is satisfiable(e) <b>then</b>
4:	$I_1 \leftarrow I_1 \cup \{e\}$
5:	<b>end if</b>
6:	<b>end for</b>
7:	<b>return</b> $I_1$

### 3.4.1.3. Gerando uma Única Configuração

No **Algoritmo 3**, é gerada uma única configuração de atribuições armazenada em C, abrangendo vários  $\tau$ -conjuntos. Nesse algoritmo, o conjunto CF mantém uma lista de características que possuem uma atribuição. Isso permite ao ICPL saltar um  $\tau$ -conjunto sem ter que passá-lo pela satisfabilidade (linhas 9 a 15) e permite que uma busca não necessite ser executada se as características do conjunto T estiverem armazenadas em CF e possuírem uma atribuição (linhas 28 a 30). Quando um  $\tau$ -conjunto conduz a insatisfabilidade (linha 19) e as atribuições, exceto uma, estiverem na configuração do  $\tau$ -conjunto C (linha 21), a atribuição oposta à única exceção é válida. Isso permite atualizar as configurações (linhas 22 a 24). Na linha 32, é retornada a configuração válida completa.

Algoritmo 3 genConfiguration (FM, T): C	
1:	$(C, CF, PF) \leftarrow (0, 0, 0)$
2:	<b>for</b> each t-set e in T <b>do</b>
3:	<b>for</b> each assignment a in e <b>do</b>
4:	$PF \leftarrow PF \cup \{a.\}$
5:	<b>end for</b>
6:	<b>end for</b>

```

7:      Loop:
8:          for each t-set e in T do
9:              F ← 0
10:             for each assignment a in e do
11:                 F ← F ∪ {a.†}
12:             end for
13:             if F ⊆ CF then
14:                 continue
15:             end IF
16:             if FM.is_satisfiable(C ∪ e) then
17:                 C ← C ∪ e
18:                 CF ← CF ∪ F
19:             else
20:                 for each assignment a in e do
21:                     if ((e \ {a}) ⊆ C) ∧ (a ∉ C) then
22:                         b ← a with assignment inverted
23:                         C ← C ∪ {b}
24:                         CF ← CF ∪ {b.†}
25:                     end if
26:                 end for
27:             end if
28:             if |CF| = |PF| then
29:                 break Loop
30:             end if
31:         end for
32: return FM.satisfy(C)

```

#### 3.4.1.4. Identificando os $\tau$ -conjuntos Cobertos

Após encontrar uma única configuração, é necessário encontrar o conjunto  $CO$  que possui os  $\tau$ -conjuntos cobertos (**Algoritmo 4**). Na linha 2, há uma iteração sobre o conjunto  $T$ , que diminui de tamanho ao longo da iteração; assim, os  $\tau$ -conjuntos são cobertos e o conjunto  $CO$  pode ser retornado.

Algoritmo 4 genCovered(FM,C,T): CO	
1:	CO ← 0
2:	for each t-set e in T do
3:	if e ⊆ C then
4:	CO ← CO ∪ {e}
5:	end if
6:	end for
7:	return CO

#### 3.4.1.5. Identificando os $\tau$ -conjuntos Inválidos

No **Algoritmo 5**, são utilizados como parâmetros o modelo de características  $FM$  e o conjunto  $T_\tau$  de  $\tau$ -conjuntos descobertos para classificar os  $\tau$ -conjuntos restantes em válidos ( $V$ ) ou inválidos ( $I$ ) e retorná-los ao algoritmo ICPL (linha 2). Nas linhas 4, 5, 6, 8 e 9, cada  $\tau$ -conjunto é classificado e adicionado ou removido dos seus respectivos

conjuntos. Nas linhas 11 a 18, há otimização; se uma configuração encontrada na linha 10 possuir um  $t$ -conjunto não classificado, ele pode ser classificado como válido. A execução do algoritmo termina quando não houver mais elementos em  $T$  (os elementos de  $T$  são classificados).

Algoritmo 5 genInvalid (FM, T): (V,I)	
1:	$(I, V) \leftarrow (0,0)$
2:	<b>while</b> $T \neq 0$ <b>do</b>
3:	$e \leftarrow$ any $t$ -set in $T$
4:	<b>if</b> $\neg FM.is\_satisfiable(e)$ <b>then</b>
5:	$I \leftarrow I \cup \{e\}$
6:	$T \leftarrow T \setminus \{e\}$
7:	<b>else</b>
8:	$V \leftarrow V \cup \{e\}$
9:	$T \leftarrow T \setminus \{e\}$
10:	$C \leftarrow FM.satisfy(e)$
11:	$NewV \leftarrow 0$
12:	<b>for each</b> $t$ -set $g$ in $T$ <b>do</b>
13:	<b>if</b> $g \subseteq C$ <b>then</b>
14:	$NewV \leftarrow NewV \cup \{g\}$
15:	<b>end if</b>
16:	<b>end for</b>
17:	$V \leftarrow V \cup NewV$
18:	$T \leftarrow T \setminus NewV$
19:	<b>end if</b>
20:	<b>end while</b>
21:	<b>return</b> $(V, I)$

### 3.4.1.6. Paralelizando

Processamento em paralelo pode ser definido como uma forma eficiente de processar a informação com ênfase na exploração de eventos concorrentes no processo computacional [Hwang et al., 1985]. Na paralelização, os processadores trabalham simultaneamente com o objetivo de resolver um problema específico. Em algoritmos paralelos, o problema a ser resolvido é decomposto em problemas menores e atribuídos aos processadores para serem solucionados simultaneamente e de forma coordenada.

Em ICPL, o **Algoritmo 2**, o **Algoritmo 3**, o **Algoritmo 4** e o **Algoritmo 5** são de dados paralelos, pois cada um deles obtém um conjunto de  $t$ -conjuntos para processar. O **Algoritmo 2** utiliza dois parâmetros, o modelo de características  $FM$  e o conjunto  $T_1$ . Esse conjunto é dividido em  $n$  partes e o algoritmo é executado em paralelo em  $n$  threads que possuem uma das  $n$  partes e uma cópia do objeto  $FM$ . Ao fim das execuções, existem  $n$

resultados disponíveis, que podem ser combinados no resultado final  $\mathbb{T}_1$ . O **Algoritmo 3**, o **Algoritmo 4** e o **Algoritmo 5** recebem o parâmetro  $\mathbb{T}$  em paralelo e seus valores de retorno podem ser combinados no resultado final.

### 3.4.2. Pacogen

No algoritmo Pacogen [Hervieu et al., 2011], são analisados modelos de características e gerado automaticamente um conjunto de configurações para teste que se adapta ao critério  $\tau$ -wise. Ele utiliza o paradigma de programação de restrição [Karatas et al., 2010; Hentenryck et al., 1991] para lidar com os relacionamentos entre as características da LPS. O processo de geração do conjunto de configurações de teste implementado por Pacogen possui duas etapas principais.

Na primeira etapa, o modelo de características é transformado em um modelo de restrições de domínio finito que possui um conjunto de restrições do tipo herança. Essas restrições são os relacionamentos opcional (OPT), AND, OR, XOR e CARD entre as características e um conjunto de restrições transversais. Que correspondem aos relacionamentos REQUIRE e MUTEX. Para representar as restrições do tipo herança, a restrição global opt (A, [B, C,...], and (A, [, [B,C,...]), xor (A,[B,C,...]), card (A, [B, C, ...]) N, M) pode ser utilizada, sendo que as variáveis de domínio utilizam dois valores (0 e 1). Essa restrição global expressa a relação hierárquica entre a característica A e seus filhos B, C, . . . . As variáveis N e M representam o mínimo e o máximo de filhos, respectivamente.

Na segunda etapa, os pares válidos são adicionados ao modelo de restrições e uma matriz de restrições é construída e preenchida com as variáveis. Essa matriz deve ser preenchida para satisfazer o critério  $\tau$ -wise. O algoritmo Pacogen possui quatro componentes principais:

- **Analisador de características:** transforma o modelo de características em um modelo de restrição representado pela estrutura de árvore de síntese abstrata;
- **Verificador de consistência:** avalia as restrições do modelo de características dentro da matriz de restrições. Essa matriz possui o tamanho  $K \times n$ , sendo  $K$  a quantidade de características e  $n$  uma estimativa da quantidade de configurações de teste;

- **Gerador de restrições pairwise:** adiciona as restrições globais que satisfazem o critério  $\tau$ -wise ao processo de geração de configuração de teste;
- **Gerador de configurações de teste:** soluciona as restrições e gera a solução encontrada sob a forma de uma matriz de restrições.

### 3.5. Análise Comparativa

A análise comparativa está baseada na caracterização dos dois algoritmos abordados neste trabalho. Essas características compõem os critérios de comparação que permitem observar semelhanças e diferenças. Para a análise, foram adotados os seguintes critérios de comparação [Lochau et al., 2012]:

- **Entrada.** Qual é o tipo de entrada necessária ou qual elemento pode ser processado;
- **Saída.** São esperados casos de teste ou uma descrição deles para que possam ser gerados. O grau de automação é examinado identificando se a geração da saída é obtida de forma automática ou semi-automática;
- **Níveis de Teste.** Identificação de qual nível específico de teste o algoritmo cobre;
- **Rastreabilidade.** Uma propriedade importante dos algoritmos de teste em LPS é o mapeamento entre casos de teste e requisitos, arquitetura e modelo de características. A rastreabilidade permite encontrar o componente responsável por falhas encontradas e adaptar testes caso algum requisito seja alterado;
- **Processo de Desenvolvimento.** É importante saber se o algoritmo pode ser integrado no processo de desenvolvimento da LPS. Para isso, deve ficar claro se o algoritmo diferencia-se entre Engenharia de Domínio e Engenharia de Aplicação e como ele modela e manipula a variabilidade para fins de teste;
- **Aplicação.** Verifica se o algoritmo pode ser integrado no processo de desenvolvimento de uma empresa. Para esse propósito, o algoritmo deve conter uma descrição passo-a-passo que permita sua reprodução.

A análise comparativa dos algoritmos de teste em LPS em relação aos critérios apresentados anteriormente é apresentada na Tabela 3-2. Os algoritmos estão posicionados na horizontal e os critérios na vertical. Os algoritmos possuem elementos importantes, tais como, identificação de variabilidade, geração de casos de teste e rastreabilidade. Esses elementos são para os desenvolvedores de LPS indicadores de como utilizar as vantagens fornecidas pela LPS, como reutilização de artefatos, redução de custo e de time-to-market.

**Tabela 3-2 - Análise Comparativa (Adaptada de [Lochau et al., 2012])**

<b>Algoritmos</b>	<b>ICPL</b>	<b>Pacogen</b>
<b>Entrada</b>	Modelo de características no formato XML ( <i>eXtensible Markup Language</i> )	Modelo de características no formato XML ( <i>eXtensible Markup Language</i> )
<b>Saída</b>	Configuração de casos de teste gerados de forma automática	Configuração de casos de teste gerados de forma automática
<b>Nível de Teste</b>	Teste combinatório	Teste combinatório
<b>Rastreabilidade</b>	Atende	Atende
<b>Processo de Desenvolvimento - Integração</b>	Engenharia de domínio	Engenharia de domínio
<b>Processo de Desenvolvimento - Variabilidade</b>	Características obrigatórias, OR, opcionais e alternativas.	Características obrigatórias, OR, opcionais, alternativas e cardinalidade.
<b>Reúso de Software</b>	Atende	Atende
<b>Aplicação</b>	Atende	Atende

Os algoritmos possuem um método diferente para alcançar um objetivo comum. ICPL utiliza um algoritmo guloso e Pacogen utiliza programação de restrições e a técnica de minimização cujo ponto principal é retornar uma solução viável com uma quantidade ótima de configurações de teste encontradas em determinado período de tempo que pode ser estipulado pelo usuário. Além disso, ao contrário de ICPL, Pacogen não utiliza solucionadores SAT, cuja solução intuitiva consiste em valorizar variáveis com as combinações possíveis. Uma vantagem da programação de restrições sobre SAT é a sua capacidade para resolver problemas de otimização, ou seja, não apenas encontrar uma solução para o problema, mas encontrar uma solução ótima para o problema [Cohen et al., 2008].

### **3.6. Considerações Finais**

Para garantir a qualidade de um software, diversas visões sobre testes de software emergem. Essas visões exigem da equipe de teste esforço substancial para cobrir as possíveis formas de testar um sistema. Em LPS, esse desafio torna-se maior, pois a quantidade de pontos de variação faz o teste dos produtos, que podem ser construídos em uma LPS, impossível de ser realizado. Assim, os algoritmos para teste de software têm que se adaptar ao contexto de LPS para gerar resultados que comprovem a qualidade dos produtos desenvolvidos.

## 4. MEDIDAS DE SOFTWARE

### 4.1. Considerações Iniciais

Um dos objetivos básicos da Engenharia de Software é a transformação da criação de sistemas de software de uma maneira artística, indisciplinada e pouco entendível para uma forma devidamente controlada, quantificada e previsível. Medidas de software é um assunto discutido há mais de 20 anos na Engenharia de Software, pois medir é fundamental para analisar a qualidade e a produtividade dos processos de software [Pressman, 2009].

Este capítulo está organizado da seguinte forma. Medidas de software são definidas e descritas na Seção 4.2. A importância e os benefícios em realizar medição são apresentados na Seção 4.3. Tipos de medidas são descritos na Seção 4.4. As medidas utilizadas para a análise dos algoritmos de técnica combinatória para realizar testes em LPS são abordadas na Seção 4.5.

### 4.2. Definição

Medição de software é uma área de estudo que visa medir atributos de determinada entidade, consistindo em uma técnica para medir propriedades ou características de produtos, processos ou recursos. O tema "Medição de Software" é alvo de discussão na Engenharia de Software há mais de 20 anos, porém, na prática, é pouco utilizada nos projetos de desenvolvimento de software. Existem algumas definições complementares [Sommerville, 2010]:

- **Medida:** fornece indicação quantitativa da extensão, de quantidade, de dimensão, de capacidade ou de tamanho de algum atributo de um produto ou processo. É uma função de mapeamento;
- **Medição:** ato de determinação de uma medida;
- **Métrica:** medida quantitativa do grau em que um sistema encontra-se em relação a um determinado atributo;
- **Indicadores:** métrica ou combinação de métricas que fornece uma compreensão de um processo, projeto ou produto.

Uma medida de software é qualquer tipo de medição que se refira a um sistema de software, processo ou documentação relacionada [Pressman, 2009]. Métricas tratam de dados numéricos que demonstram em forma de indicadores o estado atual de um

determinado projeto, ou produto. Elas devem ser válidas quantificando o objeto de medição, confiável produzindo os mesmos resultados dadas as mesmas condições, práticas e fáceis de interpretar.

As medidas devem ser facilmente entendidas, calculadas, testadas e devem gerar uma sugestão de melhoria no ambiente organizacional; para esse fim, é importante que a medida seja obtida o mais cedo possível no processo de desenvolvimento de um software, visando obter o máximo de vantagens que ela possa proporcionar. As medidas de software devem ser a rigor práticas, com baixo custo e confiáveis.

### **4.3. Importância e Benefícios**

Medir um software é importante, principalmente, porque com a medição pode-se entender e aperfeiçoar o processo de gestão e desenvolvimento de software, inclusive melhorando a relação com os clientes. Outros pontos importantes ligados à medição de um software estão relacionados com a melhoria na gerência dos contratos da empresa, às melhorias na produtividade do processo como um todo, aos benefícios da utilização de métodos padronizados e ao retorno de investimento previamente estimado para um determinado projeto [Pressman, 2009].

Utilizando medidas é possível conseguir melhorias e resultados mais satisfatórios do software e mais segurança para os gerentes de projeto. É uma maneira de eliminar os obstáculos, corrigir erros e falhas, antes que o produto seja entregue ao cliente. Alguns benefícios da utilização de medidas são:

- Entender e aperfeiçoar o processo de desenvolvimento de software;
- Melhorar a gerência de projetos e o relacionamento com clientes;
- Reduzir frustrações e pressões de cronograma;
- Indicar a qualidade de um produto de software;
- Avaliar a produtividade do processo;
- Avaliar os benefícios (em termos de produtividade e qualidade) de novos métodos e ferramentas de engenharia de software;
- Identificar as melhores práticas de desenvolvimento de software;
- Avaliar o impacto da variação de um ou mais atributos do produto ou do processo na qualidade e produtividade;
- Formar uma baseline para estimativas e melhorar a exatidão das estimativas;

- Oferecer dados qualitativos e quantitativos para o gerenciamento de desenvolvimento de software, de forma a realizar melhorias em todo o processo de desenvolvimento de software.

#### 4.4. Tipos de Medidas

Medidas podem ser classificadas em diversos tipos considerando o contexto de sua aplicação, os critérios utilizados na sua definição e os métodos utilizados para obter a medida [Mills, 1998]. Uma maneira geral de classificá-las é em relação ao objeto de medição; sobre esse aspecto, as medidas podem ser subdivididas em:

- **Medidas de produto:** são aquelas obtidas a partir de características de um produto ou artefato em qualquer estágio do desenvolvimento de software, representam características exclusivas do produto. Exemplos desse tipo de medida são o tamanho de um produto, a complexidade de um código e a quantidade de páginas de um documento;
- **Medidas de processo:** medem os atributos do processo de desenvolvimento do produto, são obtidas com as atividades executadas no desenvolvimento. Exemplos desse tipo de medida são o esforço dedicado a cada atividade do processo e a quantidade de defeitos detectados;
- **Medidas de projeto:** representam indicadores do projeto de software. Exemplos desse tipo de medida é a avaliação do status de um projeto em andamento, acompanhamento de riscos potenciais, ajuste do fluxo de atividades e avaliação da produtividade da equipe do projeto;
- **Medidas de LPS:** medem atributos da LPS. Essas medidas geralmente são adaptações das medidas de produto, processo e projeto. Exemplos desse tipo de medida são a complexidade da arquitetura da LPS e a quantidade de características presentes no código fonte.

#### 4.5. Medidas de Teste para Linhas de Produtos de Software

As medidas utilizadas para avaliar a eficiência e a qualidade dos algoritmos de teste para LPS são [Perrouin et al., 2010]:

- **Performance.** É o tempo necessário que os algoritmos utilizam para calcular e gerar a matriz de cobertura;

- **Tamanho do conjunto de teste gerado.** É a quantidade de casos de teste que determinada estratégia de geração calcula;
- **Ocorrência das  $\tau$ -tuplas.** É definida pela quantidade de vezes que determinada  $\tau$ -tupla aparece em um conjunto de testes;
- **Frequência das  $\tau$ -tuplas.** É a relação entre a ocorrência das  $\tau$ -tuplas e o tamanho do conjunto de testes, dada por um valor no intervalo  $[0, 1]$ . O valor '0' para determinada  $\tau$ -tupla significa que não há ocorrências dessa tupla no conjunto de testes. Isso normalmente não deve ocorrer, pois, segundo o critério de adequação  $\tau$ -wise, uma tupla válida necessita aparecer pelo menos uma vez no conjunto de testes. O valor '1' significa que a  $\tau$ -tupla aparece em todas as configurações de teste geradas, o que implica que ela é composta por características comuns. Essa medida pode ser utilizada como um controle de conformidade, pois se uma ou mais  $\tau$ -tupla compostas de características comuns tem frequência inferior a 1, então a configuração de teste não é válida no que diz respeito ao modelo de características.

#### **4.6. Considerações Finais**

Neste capítulo, foram apresentados conceitos importantes para a compreensão da definição e aplicação das medidas de software. Elas podem auxiliar desde o processo de análise de custos do projeto às atividades de teste. A utilização de medidas de software pode ser considerada uma forma de fundamentar decisões de gerenciamento, elas permitem avaliar tecnologias e compará-las em relação a diversos aspectos permitindo escolher qual a melhor tecnologia para um projeto específico.

## **5. AVALIAÇÃO DA EXECUÇÃO DOS ALGORITMOS**

### **5.1. Considerações Iniciais**

Neste capítulo, as medidas para LPSs apresentadas no capítulo anterior são aplicadas sobre os algoritmos ICPL e Pacogen, objetivando comparar e avaliar suas soluções propostas.

Este capítulo está organizado da seguinte forma. A ferramenta SPLOT utilizada como fonte de obtenção das LPSs selecionadas para análise é descrita na Seção 5.2. A definição e a descrição das LPSs utilizadas para a aplicação dos algoritmos de testes em LPS são apresentadas na Seção 5.3. Análise dos resultados obtidos é descrita na Seção 5.4.

### **5.2. SPLOT**

A ferramenta web SPLOT (Software Product Line Online Tools) é uma coleção de ferramentas interativas on-line para edição, para configuração e para análise de modelos de características. Está focada na automatização de estatísticas computacionais e na verificação da consistência dos modelos de características. SPLOT possui uma base de dados que recebe submissões de modelos de características desenvolvidos por pesquisadores da área de LPS e mantém esses modelos disponíveis para que outros pesquisadores possam utilizá-los em suas pesquisas [Mendonça et al., 2009]. SPLOT é utilizada como fonte para a seleção das LPSs testadas neste trabalho, sua escolha deve-se ao fato de:

- Ser comumente utilizada no ambiente acadêmico e profissional da área de LPS [Mendonça et al., 2009; Wasowski et al., 2009; Classen et al., 2011; She et al., 2010];
- Possuir disponibilidade de download dos modelos de características no formato compatível com o formato requerido pelos algoritmos ICPL e Pacogen;
- Garantir que os modelos possuem pelo menos uma configuração de produto válida;
- Não possuir características inoperantes;
- Identificar os autores dos modelos de características.

### **5.3. Escolha das LPSs**

A partir do repositório SPLOT, foram selecionadas para análise seis LPSs cuja quantidade de características varia de 19 a 61. A criação desse grupo busca avaliar um

conjunto diversificado de LPS cujas principais diferenças são em relação ao tamanho, domínio, à quantidade, à classificação e às restrições entre as características. As LPSs *Documentation Generation*, *Inventory*, *TankWar*, *Fire Alarm*, *Arcade Game* e *Sienna Van* foram selecionadas, pois são comumente utilizadas por pesquisadores como estudo de caso para avaliar testes combinatórios [Bagheri et al., 2011; Perrouin et al., 2010; Lopez-Herrejon et al., 2014; Hervieu et al., 2011]:

- *LPS Documentation Generation*. É um gerador de documentação comercial para sistemas de software. Ele opera preenchendo um repositório com uma série de fatos utilizados para derivar documentação baseada na web para os sistemas analisados. Essa documentação inclui, por exemplo, resumos textuais, gráficos de controle de fluxo e informação arquitetônica. Essa LPS expressa e implementa diferentes sistemas de geração de documentação em vários idiomas e linguagens de programação, incluindo Cobol, SQL e JCL. Essas possibilidades são facilmente encontradas no seu modelo de características (Figura 5-1);
- *LPS Inventory*. É um sistema de controle de inventário, cujas características permitem gerenciar um catálogo de itens classificados utilizando código de barras e de categoria (genéricos, perecíveis e matéria-prima). A gestão de estoques é controlada por operações que identificam as transações de adição, de redução, de transferência, de ajuste, de consignação, de atribuição e de montagem de itens (Figura A-1, Anexo A);
- *LPS TankWar*. É um jogo desenvolvido por estudantes da Universidade de Magdeburg, situada na Alemanha [Schulze et al., 2010]. Essa LPS possui características que permitem a portabilidade comum a jogos de dispositivos móveis, a possibilidade de alternar entre dois idiomas (inglês ou alemão), escolha de níveis de dificuldade (fácil ou difícil), escolha do tamanho da tela, habilitação de som, escolha da quantidade de tipos de tanque, escolha de ferramentas para incorporar ao tanque e armazenamento de resultados em um ranking (Figura A-2, Anexo A);
- *LPS Fire Alarm*. É um sistema de alarme contra incêndios, cujos dispositivos permitem a detecção de focos de incêndio com a utilização de laser ultra-sensível e sensores de temperatura e de fumaça. Caso algum foco de incêndio seja detectado o sistema gera alertas automáticos sonoros, um relatório composto por informações sobre o incêndio e aciona o sistema de extintores. Essa variabilidade de funções é expressa pelo modelo de características (Figura A-3, Anexo A);

- 📁 documentation\_generation
  - analysis
    - language\_analysis
      - ^ [1,\*]
        - cobol
          - ^ [1,1]
            - ibm\_cobol
            - microfocus\_cobol
          - jcl
          - sql
          - delphi
          - progress
        - version\_mngt
        - subsystems
      - presentation
        - localization
          - ^ [1,\*]
            - english
            - dutch
          - interaction
            - ^ [1,1]
              - static
              - dynamic
            - main\_pages
              - ^ [1,\*]
                - program\_page
                  - ^ [1,\*]
                    - annotation\_section
                    - activation\_section
                    - entities\_section
                    - parameters\_section
                  - copybook\_page
                  - statistics\_page
                  - ^ [1,1]
                    - with\_history
                    - without\_history
                  - indexes
                  - search\_page
                  - subsystem\_page
                  - source\_page
                  - source\_difference
                - visualizations
                  - ^ [1,\*]
                    - perform\_graph
                    - conditional\_perform\_graph
                    - jcl\_graph
                    - subsystem\_graph
                    - overview\_graph
                - database
                  - ^ [1,1]
                    - db2
                    - oracle
                    - mysql

**Extra Constraints:**

- constraint\_1 : ~Subsystem\_page or Subsystems
- constraint\_2 : ~Subsystem\_graph or Subsystems
- constraint\_3 : ~JCL\_graph or JCL
- constraint\_4 : ~Conditional\_perform\_graph or Cobol
- constraint\_5 : ~Perform\_graph or Cobol
- constraint\_6 : ~Source\_difference or Version\_mngt
- constraint\_7 : ~Static or ~Search\_page
- constraint\_8 : ~Static or ~Annotation\_section

| Legenda: |                                |
|----------|--------------------------------|
| ●        | Característica Obrigatória     |
| ○        | Característica Opcional        |
| ▫        | Característica Alternativa     |
| ^ [1,*]  | Alternativa não exclusiva - OR |
| ^ [1,1]  | Alternativa exclusiva - XOR    |

**Figura 5-1 - Modelo de Características da LPS Documentation Generation (Adaptada: SPLOT<sup>3</sup>)**

<sup>3</sup> <http://www.splot-research.org/>

- LPS Arcade Game. Foi desenvolvida para apoiar o aprendizado sobre LPS, sendo composta por três jogos de arcade simples. O principal objetivo foi o de ser abrangente, ao invés de concentrar-se em gráficos de alta qualidade. Suas características permitem instalar e desinstalar o jogo, escolher a trilha sonora, salvar o jogo e sua respectiva pontuação (Figura A-4, Anexo A);
- LPS Sienna Van. É um sistema que permite a escolha das características que irão compor uma van para um cliente. É composta por modelo, código do modelo, categoria, motor, cor do interior e exterior do carro, acessórios, antena e vidros que compõem uma minivan (Figura A-5, Anexo A).

A quantidade de possíveis configurações de produtos induzidas pelas características cresce exponencialmente o que leva rapidamente a muitos produtos para testar como pode ser visto na Tabela 5-1, mas é importante ressaltar que ICPL e Pacogen não objetivam gerar configurações individuais de testes para todos os produtos que uma LPS pode gerar. Esses algoritmos geram um conjunto de configurações de teste que deve ser o menor possível para cobrir as características relevantes e uma amostra eficiente para testar a LPS.

**Tabela 5-1 - Caracterização das LPSs Selecionadas**

| ESPECIFICIDADE \ LPS                       | Fire Alarm                    | Inventory                         | TankWar | Sienna Van                             | Documentation Generation                                   | Arcade Game         |
|--|-------------------------------|-----------------------------------|---------|--|--|---------------------|
| Domínio                                    | Sistema de Alarme de Incêndio | Sistema de Controle de Inventário | Jogo    | Sistema online para escolha de minivan | Gerador de documentos comerciais para sistemas de software | Jogo                |
| Quantidade de Características              | 19                            | 37                                | 37      | 38                                     | 44   | 61                  |
| Quantidade de Características Obrigatórias | 5                             | 10                                | 7       | 12                                     | 8  | 19                  |
| Quantidade de Características Opcionais    | 0                             | 6                                 | 8       | 5                                      | 3  | 5                   |
| Quantidade de Características Alternativas | 14                            | 21                                | 22      | 21                                     | 33   | 37                  |
| Quantidade de Produtos                     | 18                            | 2.028.096                         | 39.060  | 2.520                                  | 5,57x10 <sup>7</sup>                                       | 3,3x10 <sup>9</sup> |

## 5.4. Análise dos Resultados

Nesta seção, são apresentados os resultados obtidos após executar os algoritmos ICPL e Pacogen e utilizar as medidas performance, tamanho do conjunto de teste gerado, ocorrência das t-tuplas e frequência das t-tuplas sobre as seis LPS selecionadas para análise.

### 5.4.1. Execução dos Algoritmos

A solução dada pelos algoritmos representa um conjunto de casos de teste gerado a partir do modelo de características no formato XML para uma força  $t = 2$  de cobertura. Ao

executar o algoritmo ICPL<sup>4</sup>, obteve-se o resultado apresentados na Tabela 5-2 (LPS Documentation Generation). Ao executar o algoritmo Pacogen<sup>5</sup>, obteve-se o resultado apresentados na Tabela 5-3 (LPS Documentation Generation). As demais tabelas estão no Anexo B.

As matrizes de cobertura são apresentadas nessas tabelas, em que há o conjunto de configurações de casos de teste que equivalem aos produtos a serem testados para assegurar que a interação entre as características da LPS funcione corretamente. Cada linha da tabela representa uma característica e cada coluna representa um produto. Para identificar que uma característica está incluída em um produto, o símbolo 'X' é colocado na célula correspondente; por outro lado, caso uma característica não esteja presente em um produto, é colocado o símbolo '-'. Ambos os algoritmos possuem como principal vantagem o fato dos casos de teste gerados poderem ser testados utilizando métodos de teste para sistemas únicos.

**Tabela 5-2 - Matriz de Cobertura da LPS Documentation Generation Gerada por ICPL**

| Características           | Produtos |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |
|---------------------------|----------|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|
|                           | 1        | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
| Activation section        | -        | - | - | X | X | X | X | - | - | -  | -  | -  | X  | X  | -  | X  | X  |
| Analysis                  | X        | X | X | X | X | X | X | X | X | X  | X  | X  | X  | X  | X  | X  | X  |
| Annotation section        | -        | - | - | - | X | - | X | - | - | X  | X  | X  | X  | -  | -  | -  | -  |
| Cobol                     | -        | - | X | X | X | X | X | X | - | X  | X  | -  | -  | X  | X  | -  | X  |
| Conditional perform graph | -        | - | X | X | - | - | X | - | - | -  | X  | -  | -  | X  | -  | -  | -  |
| Copybook page             | -        | - | - | X | X | X | X | - | - | -  | X  | X  | -  | -  | X  | -  | -  |
| Database                  | X        | X | X | X | X | X | X | X | X | X  | X  | X  | X  | X  | X  | X  | X  |
| Db2                       | -        | X | - | - | - | X | - | X | - | -  | X  | -  | -  | -  | -  | X  | -  |
| Delphi                    | -        | - | X | X | - | - | X | X | X | X  | -  | -  | -  | -  | -  | -  | -  |
| Doc_gen                   | X        | X | X | X | X | X | X | X | X | X  | X  | X  | X  | X  | X  | X  | X  |
| Dutch                     | X        | X | - | - | X | - | X | X | - | -  | X  | -  | -  | -  | -  | -  | -  |
| Dynamic                   | X        | X | - | X | X | - | X | - | - | X  | X  | X  | X  | -  | X  | -  | -  |
| English                   | -        | X | X | X | - | X | - | - | X | X  | -  | X  | X  | X  | X  | X  | X  |
| Entities section          | -        | - | - | X | X | X | X | - | - | -  | X  | -  | X  | X  | -  | -  | -  |
| lbm_cobol                 | -        | - | - | X | X | X | - | - | - | -  | -  | -  | -  | X  | X  | -  | -  |
| Indexes                   | -        | X | X | X | - | X | - | - | - | X  | X  | -  | -  | -  | -  | -  | -  |
| Interaction               | X        | X | X | X | X | X | X | X | X | X  | X  | X  | X  | X  | X  | X  | X  |
| Jcl                       | -        | X | X | X | - | - | X | X | X | -  | -  | X  | X  | -  | X  | X  | -  |
| Jcl_graph                 | -        | - | X | X | - | - | X | X | - | -  | -  | X  | -  | -  | X  | -  | -  |
| Language analysis         | X        | X | X | X | X | X | X | X | X | X  | X  | X  | X  | X  | X  | X  | X  |
| Localization              | X        | X | X | X | X | X | X | X | X | X  | X  | X  | X  | X  | X  | X  | X  |
| Main_pages                | X        | X | X | X | X | X | X | X | X | X  | X  | X  | X  | X  | X  | X  | X  |
| Microfocus_cobol          | -        | - | X | - | - | - | X | X | - | X  | X  | -  | -  | -  | -  | -  | X  |
| Mysql                     | X        | - | X | - | - | - | X | - | - | X  | -  | X  | -  | -  | X  | -  | -  |
| Oracle                    | -        | - | - | X | X | - | - | - | X | -  | -  | -  | X  | X  | -  | -  | X  |
| Overview_graph            | -        | X | - | X | - | - | X | X | - | -  | X  | -  | -  | -  | -  | -  | -  |
| Presentation              | X        | X | X | X | X | X | X | X | X | X  | X  | X  | X  | X  | X  | X  | X  |
| Parameters section        | -        | - | - | X | X | X | X | - | - | -  | -  | X  | -  | X  | -  | -  | -  |
| Perform_graph             | -        | - | X | - | X | - | X | - | - | -  | X  | -  | -  | -  | -  | -  | -  |
| Program page              | -        | - | - | X | X | X | X | - | - | X  | X  | X  | X  | X  | -  | X  | X  |
| Progress                  | X        | X | - | X | - | X | - | - | - | X  | X  | -  | -  | -  | -  | -  | -  |
| Search_page               | -        | - | - | X | X | - | X | - | - | X  | X  | X  | -  | -  | X  | -  | -  |

<sup>4</sup> O código fonte de ICPL está disponível em <http://heim.ifi.uio.no/martifag/splc2012/>

<sup>5</sup> O código fonte de Pacogen está disponível em <http://www.irisa.fr/lande/gotlieb/resources/Pacogen/Pacogen.html>

**Tabela 5-2 - Matriz de Cobertura da LPS Documentation Gereneration Gerada por ICPL (cont.)**

| Características   | Produtos |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |
|-------------------|----------|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|
|                   | 1        | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
| Source_difference | -        | X | X | - | - | X | X | - | - | -  | -  | -  | X  | -  | -  | -  | -  |
| Source_page       | X        | - | X | - | X | - | - | X | - | -  | -  | -  | -  | -  | -  | -  | -  |
| Sql               | -        | - | X | X | - | - | - | X | X | X  | X  | -  | -  | -  | -  | -  | -  |
| Static            | -        | - | X | - | - | X | - | X | X | -  | -  | -  | -  | X  | -  | X  | X  |
| Statistics page   | -        | - | - | X | X | X | X | X | X | X  | X  | -  | X  | -  | X  | -  | -  |
| Subsystem_graph   | -        | X | - | - | X | - | - | X | - | -  | X  | X  | -  | -  | -  | -  | -  |
| Subsystem_page    | -        | X | - | - | X | X | - | X | - | X  | -  | -  | -  | X  | -  | -  | -  |
| Subsystems        | -        | X | X | - | X | X | - | X | - | X  | X  | X  | -  | X  | -  | -  | -  |
| Version_mngt      | -        | X | X | - | X | X | X | - | - | -  | -  | -  | X  | -  | -  | -  | -  |
| Visualizations    | -        | X | X | X | X | - | X | X | - | -  | X  | X  | -  | X  | X  | -  | -  |
| With_history      | -        | - | - | - | X | X | - | X | - | X  | -  | -  | -  | -  | -  | X  | -  |
| Without_history   | -        | - | - | X | - | - | X | X | - | X  | -  | -  | X  | -  | -  | -  | -  |

**Tabela 5-3 - Matriz de Cobertura da LPS Documentation Gereneration Gerada por Pacogen**

| Características           | Produtos |   |   |   |   |   |   |   |   |    |    |    |    |
|---------------------------|----------|---|---|---|---|---|---|---|---|----|----|----|----|
|                           | 1        | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| Activation section        | X        | - | - | - | - | - | - | X | X | X  | -  | X  | X  |
| Analysis                  | X        | X | X | X | X | X | X | X | X | X  | X  | X  | X  |
| Annotation section        | X        | - | - | - | X | - | - | X | X | -  | -  | X  | X  |
| Cobol                     | -        | X | X | - | X | X | X | X | X | -  | X  | X  | -  |
| Conditional_perform_graph | -        | - | - | - | X | X | - | X | X | -  | X  | -  | -  |
| Copybook page             | -        | X | X | X | - | - | - | X | - | X  | X  | X  | -  |
| Database                  | X        | X | X | X | X | X | X | X | X | X  | X  | X  | X  |
| Db2                       | -        | - | X | - | - | - | - | X | X | X  | -  | -  | X  |
| Delphi                    | -        | X | X | X | - | - | - | X | X | -  | X  | X  | -  |
| Doc_gen                   | X        | X | X | X | X | X | X | X | X | X  | X  | X  | X  |
| Dutch                     | X        | X | - | - | - | X | X | - | X | -  | X  | X  | X  |
| Dynamic                   | X        | - | - | - | X | X | - | X | X | -  | -  | X  | X  |
| English                   | -        | X | X | X | X | - | - | X | - | X  | -  | -  | -  |
| Entities section          | X        | - | - | - | X | - | - | - | X | X  | -  | X  | X  |
| lbn_cobol                 | -        | X | X | - | X | - | - | - | X | -  | -  | -  | -  |
| Indexes                   | X        | X | X | X | X | X | X | X | X | X  | X  | X  | X  |
| Interaction               | X        | X | X | X | X | X | X | X | X | X  | X  | X  | X  |
| Jcl                       | -        | X | X | X | - | X | - | - | X | X  | X  | -  | -  |
| Jcl_graph                 | -        | X | - | X | - | X | - | - | X | -  | X  | -  | -  |
| Language analysis         | X        | X | X | X | X | X | X | X | X | X  | X  | X  | X  |
| Localization              | X        | X | X | X | X | X | X | X | X | X  | X  | X  | X  |
| Main_pages                | X        | X | X | X | X | X | X | X | X | X  | X  | X  | X  |
| Microfocus_cobol          | -        | - | - | - | - | X | X | X | - | -  | X  | X  | -  |
| Mysql                     | -        | X | - | X | - | X | X | - | - | -  | -  | X  | -  |
| Oracle                    | X        | - | - | - | X | - | - | - | - | -  | X  | -  | -  |
| Overview_graph            | -        | X | - | X | X | X | - | - | - | X  | -  | -  | -  |
| Parameters section        | X        | - | - | - | X | - | - | X | X | X  | -  | X  | -  |
| Perform_graph             | -        | X | - | - | - | X | - | X | X | -  | X  | -  | -  |
| Presentation              | X        | X | X | X | X | X | X | X | X | X  | X  | X  | X  |
| Program page              | X        | - | - | - | X | - | - | X | X | X  | -  | X  | X  |
| Progress                  | X        | - | - | - | X | - | X | - | X | -  | -  | X  | X  |
| Search_page               | X        | - | - | - | - | X | - | X | X | -  | -  | -  | -  |
| Source_difference         | -        | - | X | X | - | - | X | X | - | -  | X  | X  | -  |
| Source_page               | -        | - | X | X | X | X | X | X | - | -  | -  | -  | -  |
| Sql                       | -        | X | X | X | - | - | X | - | X | -  | X  | -  | -  |
| Static                    | -        | X | X | X | - | - | X | - | - | X  | X  | -  | -  |
| Statistics page           | -        | - | X | X | X | X | X | - | X | X  | X  | X  | -  |
| Subsystem_graph           | -        | X | - | X | X | - | - | X | - | X  | X  | -  | -  |
| Subsystem_page            | -        | X | - | X | - | - | X | X | - | X  | X  | -  | -  |
| Subsystems                | -        | X | - | X | X | - | X | X | - | X  | X  | -  | -  |
| Version_mngt              | -        | - | X | X | X | - | X | X | - | X  | X  | -  | -  |
| Visualizations            | -        | X | - | X | X | X | - | X | X | X  | X  | -  | -  |
| With_history              | -        | - | X | X | X | X | - | - | - | -  | -  | X  | -  |
| Without_history           | -        | - | - | - | - | - | - | X | - | X  | X  | X  | -  |

#### 5.4.2. Performance e Tamanho do Conjunto de Teste Gerado

Performance é utilizada como medida de desempenho, sendo o tempo necessário para os algoritmos executarem. Os resultados da aplicação das medidas sobre as soluções resultantes da execução dos algoritmos são apresentados na Tabela 5-4.

**Tabela 5-4 - Aplicação das Medidas nas LPSs**

| LPS                      | Algoritmos | Performance (milissegundos) | Tamanho do Conjunto de Teste Gerado |
|--------------------------|------------|-----------------------------|-------------------------------------|
| Fire Alarm               | ICPL       | 722                         | 10                                  |
|                          | Pacogen    | 6.520                       | 7                                   |
| Inventory                | ICPL       | 617                         | 13                                  |
|                          | Pacogen    | 43.600                      | 12                                  |
| Arcade Game              | ICPL       | 1.057                       | 18                                  |
|                          | Pacogen    | 125.980                     | 13                                  |
| Documentation Generation | ICPL       | 1.140                       | 17                                  |
|                          | Pacogen    | 87.310                      | 13                                  |
| Sienna Van               | ICPL       | 1.089                       | 22                                  |
|                          | Pacogen    | 66.420                      | 20                                  |
| TankWar                  | ICPL       | 672                         | 15                                  |
|                          | Pacogen    | 53.310                      | 11                                  |

Estatisticamente, ao analisar esses resultados com o teste Wilcoxon [Araujo et al., 2006], cujo resultado aponta valor  $W = 21$ ,  $p\text{-valor} = 0,03125 < 5\%$ , pode-se afirmar que, com nível de significância de 5%, há evidências de diferença entre os resultados das medidas para ambos os algoritmos. O coeficiente de Spearman [Araujo et al., 2006] aponta correlação positiva entre a performance e o tamanho do conjunto de teste, comprovada pelo  $p\text{-valor}$  de 0,0027.

Uma observação inicial que pode ser feita a partir dessa tabela é o algoritmo ICPL ser mais rápido do que o algoritmo Pacogen. Isto era esperado pois o objetivo de Pacogen é minimizar o processo de geração do conjunto de configuração de teste e não otimizar o tempo necessário para a execução do processo. No entanto, diante desses resultados, pode-se dizer que a diferença entre a performance dos algoritmos não é tão relevante, pois para as LPSs, sejam elas com poucas ou muitas características, o conjunto de teste gerado foi encontrado em tempo total de computação razoável dado em milissegundos.

Não há relação direta entre a quantidade de produtos, de características e a performance, pois a LPS Fire Alarm e a LPS Sienna Van que possuem a menor quantidade de produtos válidos, 18 e 2.520, respectivamente, não são as LPSs que necessitaram de uma performance menor. Mas, pode-se notar que, para a LPS Documentation Generation e a LPS Arcade Game, que possuem a maior quantidade de produtos válidos e características,  $5,57 \times 10^7$ ; 44 e  $3,3 \times 10^9$ ; 61,

respectivamente, foi necessário tempo maior para o cálculo de uma solução. Para a LPS `Fire Alarm`, a LPS `Tankwar` e a LPS `Inventory`, que possuem a menor quantidade de características 19, 37 e 37, respectivamente, foi necessário tempo menor para o cálculo de uma solução.

A medida tamanho do conjunto de testes gerados confirma maior diversidade de configurações nas soluções de ICPL para 100% das LPSs, mas esse resultado era esperado, pois existem diferenças explícitas na comparação entre dois algoritmos que utilizam estratégias diferentes, otimização e algoritmo guloso. Otimização requer a quantidade mínima de configurações de teste que satisfaz o critério  $t$ -wise e o algoritmo guloso visa encontrar um conjunto de configurações de teste de tamanho ótimo dentro de um conjunto finito. Dependendo do problema, ótimo pode significar uma quantidade máxima ou mínima. É importante ressaltar que uma quantidade maior de configurações de teste proporciona o benefício de uma cobertura de teste maior, mas possuir mais configurações de teste do que o necessário implica em esforço maior.

A relação entre a quantidade de produtos e de características e o tamanho do conjunto de casos de testes não possui correlação explícita apesar de ser esperado que a quantidade de produtos válidos aumente quando mais características são adicionadas a uma LPS. Isso pode ser confirmado ao observar a LPS `Arcade Game` que possui a maior quantidade de características, mas não possui a maior quantidade de produtos. O mesmo se aplica para a LPS `Tankwar` e para a LPS `Inventory` que possuem quantidade de características iguais, mas quantidade de produtos e tamanho do conjunto de testes diferentes e para a LPS `Sienna Van` que possui quantidade de características inferior a LPS `Documentation Generation` e a LPS `Arcade Game`, mas maior conjunto de testes.

#### **5.4.3. Ocorrência das t-tuplas e Frequência das t-tuplas**

A frequência e a ocorrência das t-tuplas da LPS `Documentation Generation` são apresentadas na Tabela 5-5. A frequência e a ocorrência das t-tuplas das LPSs `Inventory`, `TankWar`, `Fire Alarm`, `LPS Arcade Game` e `LPS Sienna Van`, são apresentadas no Anexo C.

A ocorrência avalia a otimização da geração  $\tau$ -wise utilizando a quantidade de repetições de uma t-tupla em um conjunto de testes. Tuplas compostas por características comuns aos produtos possuem ocorrência igual ao tamanho do conjunto de testes gerado. A frequência de uma tupla é definida pela divisão entre a ocorrência e o tamanho do conjunto de testes gerados, cujo valor resultante deve estar no intervalo [0, 1] para indicar que a tupla é válida e que o algoritmo satisfaz o critério  $\tau$ -wise. Conforme os dados nessas tabelas, pode ser verificado que os dois algoritmos satisfazem o critério  $\tau$ -wise e que o conjunto de configurações de testes gerados é válido no que diz respeito às restrições do modelo de características.

**Tabela 5-5 - Frequência e Ocorrência das t-Tuplas da LPS Documentation Generation**

| Tuplas   | Frequência Pacogen | Ocorrência Pacogen | Frequência ICPL | Ocorrência ICPL |
|--|--------------------|--------------------|-----------------|-----------------|
| Analysis, subsystems                                   | 8                  | 0.6                | 9               | 0.5             |
| Analysis, version mngr                                 | 4                  | 0.3                | 6               | 0.3             |
| Database, DB2  | 4                  | 0.3                | 5               | 0.2             |
| Database, Mysql  | 4                  | 0.3                | 6               | 0.3             |
| Database, Oracle                                       | 5                  | 0.3                | 6               | 0.3             |
| Doc_Gen, Analysis                                      | 13                 | 1                  | 17              | 1               |
| Doc_Gen, Presentation                                  | 13                 | 1                  | 17              | 1               |
| Doc_Gen, Localization                                  | 13                 | 1                  | 17              | 1               |
| Doc_Gen, Main pages                                    | 13                 | 1                  | 17              | 1               |
| Doc_Gen, Database                                      | 13                 | 1                  | 17              | 1               |
| Doc_Gen, Interaction                                   | 13                 | 1                  | 17              | 1               |
| Interaction, dynamic                                   | 8                  | 0.6                | 10              | 0.5             |
| Doc_Gen, Laguage analysis                              | 13                 | 1                  | 17              | 1               |
| Interaction, static                                    | 5                  | 0.3                | 7               | 0.4             |
| Language analysis, cobol                               | 10                 | 0.7                | 11              | 0.6             |
| Language analysis, cobol, IBM cobol                    | 5                  | 0.3                | 5               | 0.2             |
| Language analysis, cobol, microfocus cobol             | 5                  | 0.3                | 6               | 0.3             |
| Language analysis, Delphi                              | 5                  | 0.3                | 6               | 0.3             |
| Language analysis, JCL                                 | 7                  | 0.5                | 10              | 0.5             |
| Language analysis, Progress                            | 5                  | 0.3                | 6               | 0.3             |
| Language analysis, SQL                                 | 5                  | 0.3                | 6               | 0.3             |
| Localization, dutch                                    | 5                  | 0.3                | 6               | 0.3             |
| Localization, english                                  | 9                  | 0.6                | 12              | 0.7             |
| Main pages, copybook page                              | 7                  | 0.5                | 7               | 0.4             |
| Main pages, indexes                                    | 5                  | 0.3                | 6               | 0.3             |
| Main pages, program page                               | 9                  | 0.6                | 11              | 0.6             |
| Main pages, program page, activation_section           | 6                  | 0.4                | 8               | 0.4             |
| Main pages, program page, annotation section           | 5                  | 0.3                | 6               | 0.3             |
| Main pages, program page, entities_section             | 6                  | 0.4                | 7               | 0.4             |
| Main pages, program page, parameters_section           | 6                  | 0.4                | 6               | 0.3             |
| Main pages, search page                                | 7                  | 0.5                | 7               | 0.4             |
| Main pages, source difference                          | 3                  | 0.2                | 5               | 0.2             |
| Main pages, source page                                | 2                  | 0.1                | 4               | 0.2             |
| Main pages, statistics page                            | 9                  | 0.6                | 10              | 0.5             |
| Main pages, statistics page, with_history              | 5                  | 0.3                | 5               | 0.2             |
| Main pages, statistics page, without_history           | 4                  | 0.3                | 5               | 0.2             |
| Main pages, subsystem page                             | 6                  | 0.4                | 6               | 0.3             |
| Presentation, visualization, conditional_perform_graph | 4                  | 0.3                | 5               | 0.2             |
| Presentation, visualization, JCL_graph                 | 5                  | 0.3                | 6               | 0.3             |
| Presentation, visualization, overview_graph            | 5                  | 0.3                | 5               | 0.2             |
| Presentation, visualization, perform_graph             | 3                  | 0.2                | 4               | 0.2             |
| Presentation, visualization, subsystem_graph           | 5                  | 0.3                | 5               | 0.2             |

Como pode ser observado, nas tabelas de frequência e ocorrência, algumas tuplas repetem-se para diversos produtos. As principais razões para isso estão relacionadas com o fato da ocorrência de uma tupla ser fortemente influenciada pela natureza das características que a compõem.

Características obrigatórias devem estar presentes nas configurações de produto válidos em relação ao modelo de características, o que inclui as características obrigatórias e suas restrições (requer/excluir). Portanto, uma característica não-obrigatória pode ser incluída na tupla para satisfazer essas restrições; tais características podem ser consideradas como falso-opcional. A ocorrência de tuplas compostas por características obrigatórias corresponde ao tamanho do conjunto de configurações gerado pelos algoritmos. As restrições entre as características impõem o seu tipo de relacionamento; dessa maneira, elas são um fator que aumenta a quantidade de vezes que uma determinada tupla aparece nas configurações de teste.

## **5.5. Considerações Finais**

Neste capítulo, foram identificados e expostos os resultados da aplicação de medidas utilizadas para avaliar um conjunto composto por seis LPSs. Esses resultados indicam parâmetros capazes de mostrar a eficiência dos algoritmos ICPL e Pacogen, facilitando a escolha entre ambos para sua aplicação na fase de testes da LPS.

A principal contribuição dessa avaliação foi encontrar a melhor solução gerada pelos algoritmos utilizando medidas que permitem identificar a qualidade dos resultados obtidos confirmando que construir um conjunto de configurações de teste para LPSs requer pensar em novas estratégias; principalmente, porque alguns métodos de teste em LPS consideram que as características são independentes, mas as interações entre elas são possíveis e influenciam na construção e no teste de produtos como foi mostrado com a utilização do método pairwise.

## 6. TRABALHOS RELACIONADOS

Teste combinatório (TC) é identificado como uma estratégia para reduzir a quantidade de produtos para teste. Algoritmos TC requerem o uso de solucionadores de restrições para gerar configurações pairwise [Perrouin et al., 2010]. São descritas as abordagens Alloy-Based [Perrouin et al., 2010] e Dedicated CSP Based [Oster et al., 2010] que possuem diferentes implementações dos algoritmos TC para teste de LPS. Alloy-Based gera produtos utilizando a combinação de uma linguagem denominada Alloy [Jackson, 2006] e de solucionadores SAT. Alloy-Based é composta por quatro etapas: i) um modelo genérico de características é transformado automaticamente em um modelo denominado AFalloy; ii) geração do conjunto composto pelas combinações  $t$ -wise possíveis de características; iii) detecção das combinações válidas - uma combinação é válida se é possível derivar um produto no qual ela está presente; e iv) conjuntos de múltiplas combinações são gerados e solucionados, um conjunto de testes é gerado de acordo com os critérios de adequação  $t$ -wise. Dedicated CSP Based aplica algoritmos TC semelhantes a AETG [Cohen et al., 1997] e IPO [Lei; Tai, 1998]. Possui três etapas: i) o algoritmo CSP-Translation converte o modelo de características em uma solução de problemas de restrição binária (CSP); ii) geração do conjunto  $\mathcal{I}$  composto pelas combinações  $t$ -wise possíveis de características; e iii) um algoritmo incremental utiliza o conjunto  $\mathcal{I}$  para gerar configurações de teste válidas. Foram utilizadas cinco medidas para avaliar as duas abordagens objetivando verificar sua capacidade de gerar conjuntos de testes utilizando cálculo da quantidade de produtos, casos de testes gerados e a redundância das combinações de características. Os resultados apontam que Dedicated CSP Based é mais estável e encontra solução melhor independentemente do tamanho do modelo de características.

Na mesma linha, em um segundo trabalho, os três algoritmos para teste CASA, PGS e ICPL foram comparados [Lopez-Herrejon et al., 2014]. CASA é um algoritmo de arrefecimento simulado (simulated annealing) projetado para gerar matrizes de cobertura  $n$ -wise, ele possui três estratégias de busca aninhadas para combinar as características válidas [Garvin et al., 2011]. PGS é um algoritmo genético construtivo que adiciona um novo produto para a solução parcial em cada iteração, até que as combinações de características sejam cobertas [Ferrer et al., 2012]. ICPL é um algoritmo guloso adaptado

do algoritmo de Chvátal's, que gera um conjunto inicial de combinações de características possíveis e iterativamente; a partir deste conjunto, são gerados outros conjuntos como as combinações de características válidas e inválidas [Johansen et al., 2012]. Os algoritmos foram comparados utilizando medidas cujo foco é verificar o tamanho do conjunto gerado, a ocorrência das combinações de características e a performance dos algoritmos. Os resultados apontaram que CASA obtém menores conjuntos de testes, ICPL é o algoritmo mais rápido e obtém a maior quantidade de produtos diferentes e o melhor conjunto de testes. PGS foi o algoritmo mais lento e obteve maior conjunto de teste, mas não melhor.

Os trabalhos relacionados encontrados abordam formas de comparar abordagens de teste TC, as principais diferenças entre eles são em relação à quantidade de abordagens comparadas e à quantidade de LPSs em que foram aplicadas. O primeiro trabalho comparou duas abordagens aplicadas em 5 LPSs e o segundo trabalho comparou três abordagens aplicadas em 181 LPSs. No entanto, ambos aplicam medidas semelhantes para comprovar a eficiência das abordagens comparadas. Os trabalhos mencionados se assemelham a este no aspecto de lidar com teste TC e buscar por meio de medidas qual abordagem é mais eficiente e eficaz. Este trabalho diferencia-se dos demais por comparar duas abordagens que utilizam práticas diferentes (algoritmos gulosos e programação de restrição) para a implementação do teste TC.

## 7. CONSIDERAÇÕES FINAIS

### 7.1. Conclusões

Linhas de Produtos de Software (LPSs) são adotadas como abordagem de Engenharia de Software bem sucedida, pois contribuem para reduzir o tempo de desenvolvimento de produtos pertencentes a um segmento de mercado. Além disso, possibilita adaptar rapidamente às mudanças, reduzir custos e atender a demanda do mercado de software lançando novos produtos a um curto espaço de tempo.

No decorrer do desenvolvimento deste trabalho, foram apresentadas a importância de realizar testes em LPSs e as suas dificuldades relacionadas com as diferenças nos processos de Engenharia de Domínio e Engenharia de Aplicação e no Gerenciamento da variabilidade das LPSs. A quantidade de características variáveis entre os produtos está diretamente relacionada à quantidade de testes requeridos, o que torna, em alguns casos, o teste exaustivo e inviável.

O objetivo principal desse trabalho foi apresentar uma análise comparativa de algoritmos de técnica combinatória para testes em LPSs. O teste combinatório surgiu como alternativa para superar as dificuldades nos testes utilizando extração de um subconjunto de combinações de características ou produtos testados representando a LPS e reduzindo o esforço de teste. Ele se baseia na observação da maioria das falhas serem desencadeadas nas interações entre uma pequena quantidade de características, o que levou à definição de pares ( $t$ -tuplas) de teste.

A análise comparativa foi feita com a caracterização dos algoritmos ICPL e Pacogen e da sua aplicação em seis LPSs (Documentation Generation, Inventory, TankWar, Fire Alarm, Arcade Game e Sienna Van). Para fornecer suporte à análise, de forma objetiva, foram utilizadas medidas capazes de identificar a performance dos algoritmos, o tamanho do conjunto de teste gerado, a otimização da geração  $t$ -wise medindo a ocorrência das  $t$ -tuplas e a conformidade da configuração de teste no que diz respeito ao modelo de características, medindo a frequência das  $t$ -tuplas. Nessa análise, pôde-se perceber que, a utilização de ambos os algoritmos, trazem resultados significativos. A aplicação das medidas mostraram que ICPL possui performance melhor do que Pacogen e as suas soluções possuem maior diversidade de configurações de casos

de teste para 100% das LPSs utilizadas, indicando que Pacogen alcança o seu principal objetivo quando propõe menos configurações de casos de teste. Esses resultados podem ser considerados fatores importantes na escolha entre os algoritmos.

Ambos os algoritmos satisfazem o critério  $\tau$ -wise e os conjuntos de configurações de teste gerado possuem alto grau de cobertura. Todas as configurações possuem diferenças entre si e tem capacidade de cobrir a variedade de produtos que pode ser gerada a partir do modelo de características, aumentando assim as chances de detectar qualquer comportamento defeituoso nos produtos.

Portanto, os objetivos foram cumpridos, pois, em uma busca em trabalhos relacionados a teste em LPS, os algoritmos ICPL e Pacogen de teste combinatório foram escolhidos para serem analisados. Critérios e medidas de comparação foram identificados e aplicados para avaliar esses algoritmos e obter suas principais características e benefícios. Com os resultados obtidos, pôde-se perceber que eles são eficazes e facilitam a execução de testes em LPSs, reduzindo o esforço requerido para testes em LPSs.

## **7.2. Contribuições**

Modelos de características, em termos de sua expressividade, são equivalentes a fórmulas proposicionais e o que os torna teoricamente difíceis para processar e analisar. Essa complexidade teórica pode impedir a utilização de algoritmos que auxiliam no teste de LPSs. O presente trabalho mostrou que utilizar modelos de características para fins de testes produz contribuições benéficas, pois eles representam características comuns e variáveis e são frequentemente utilizados como parâmetros para o desenvolvimento de produtos das LPSs. Assim, mostra-se que o teste combinatório é viável na prática.

A segunda contribuição é formada pela comprovação da hipótese de que o conjunto de medidas utilizado proposto para avaliar a técnica Alloy-Based [Perrouin et al., 2010] pode ser aplicado em outras técnicas de teste combinatório. Além disso, pode ser utilizado como critério de decisão para o testador de software disposto a aplicar esse tipo de teste.

## **7.3. Limitações**

A principal limitação deste trabalho está relacionada às medidas de frequência e ocorrência das  $\tau$ -tuplas válidas, pois calcular a frequência e a ocorrência de todas as  $t$ -

tuplas válidas é inviável. Dessa forma, foi adotada a alternativa de avaliar um conjunto representativo de t-tuplas válidas que pudesse indicar a influência das combinações entre características obrigatórias e opcionais e seus relacionamentos na configuração de casos de teste.

#### **7.4. Trabalhos Futuros**

Como sugestão de trabalhos futuros, pode ser realizada nova análise comparativa, considerando quantidade maior de LPSs e outros algoritmos e medidas, pois pode ser utilizada para aumentar a confiança na abordagem de teste combinatório e incentivar sua prática na comunidade de LPS. Utilizar quantidade maior de LPSs significa replicar mais vezes a aplicação do algoritmo, aumentando as possibilidades de encontrar possíveis problemas ou seu índice de bons resultados aumentando sua confiabilidade. Além disso, sugere-se utilizar outros formatos de modelos de características como artefato de entrada para os algoritmos, aumentando o seu grau de aplicabilidade.

## REFERÊNCIAS BIBLIOGRÁFICAS

- Araújo, M. A.; Barros, M.; Travassos, G.; Murta, L. Métodos Estatísticos aplicados em Engenharia de Software Experimental. In: XXI SBBD-XX SBES, 2006.
- Bagheri, E.; Gasevic, D. Assessing the Maintainability of Software Product Line Feature Models Using Structural Metrics. In: Software Quality Journal, v. 19, n. 3, pp. 579-612, 2011.
- Becker, M. Towards a General Model of Variability in Product Families. In: Software Variability Management Workshop, p.19-27, 2003.
- Beizer, B. Software Testing Techniques. In: Institute for software research international Carnegie mellon university. 1990
- Bertolino, A. Software Testing Research: Achievements, Challenges, Dreams. In: Future of Software Engineering, 29th International Conference on Software Engineering, p. 85-103, 2007.
- Bosch, J. Staged Adoption of Software Product Families. In: Software Process: Improvement and Practice, v. 10, pp. 125-142, 2005.
- Bosch, J.; Florijn, G.; Greefhorst, D.; Kuusela, J.; Obbink, J.; Pohl, K. Variability Issues in Software Product Lines. In: 4th International Workshop Software Product-Family Engennering, v. 2290, pp.303-338, 2002.
- Brownsword, L.; Clements, P. A Case Study in Successful Product Line Development. Technical Report no. CMU/SEE-96-TR-016, Carnegie-Mellon Software Engineering Institute, 95p., 1996.
- Classen, A.; Boucher, Q.; Heymans, P. A Text-based Approach to Feature Modelling: Syntax and Semantics of TVL. In: Science of Computer Programming, v. 76, n. 12, pp. 1130-1143, 2011.
- Clements, P.; Northrop, L. A Framework for Software Product Line Practice.v. 5.0, 2007. Disponível em: <<http://www.sei.cmu.edu/productlines/>>.
- Clements, P.; Northrop, L. Software Product Lines: Practices and Patterns. Boston: Addison-Wesley, 563 p., 2002.
- Cohen, S. Product Line State of the Practice. Technical Report, no. CMU/SEI-2002-TN-017 Carnegie Mellon University, 2002.
- Cook, S. A. The Complexity of Theorem-proving Procedures. In: Third Annual ACM Symposium on Theory of Computing, pp. 151-158, 1971.
- Czarnecki, K.; Eisenecker, U. Generative Programming: Methods, Tools, and Applications. In: Computer Science, v. 2487, 2000.

- Czerwonka; J. Pairwise Testing - Combinatorial Test Case Generation. 2011. Disponível em: <<http://www.pairwise.org/>>.
- Delamaro, M. E.; Jino, M.; Maldonado, J. C. Introdução ao Teste de Software. In: Revista de Engenharia de Software, v. 1, 2007.
- Deutsch, M. Verification and Validation. In: Software Engineering. pp. 329-408, 1978.
- Díaz, J.; Pérez, J.; Alarcón, P. P.; Garbajosa, J. Agile Product Line Engineering- a Systematic Literature Review. In: Software: Practice and Experience, v. 41, pp. 921-941, 2011.
- Engström, E.; Runeson, P. Software Product Line Testing - A Systematic Mapping Study. In: Information and Software Technology, pp. 2-13, 2010.
- Ferrer, J.; Kruse, P.; Chicano, J.; Alba, E. Evolutionary Algorithm for Prioritized Pairwise Test Data Generation. In: Fourteenth International Conference on Genetic and Evolutionary Computation Conference, pp. 1213-1220, 2012.
- Fouché, S.; Cohen, M. B.; Porter, A. Incremental Covering Array Failure Characterization in Large Configuration Spaces. In: XVIII International Symposium on Software testing and analysis, pp. 177-188, 2009.
- Garvin, B.; Cohen, M.; Dwyer, M. Evaluating Improvements to a Metaheuristic Search for Constrained Interaction Testing. In: Empirical Software Engineering, v. 16, n. 1, p. 61-102, 2011.
- Grave, T. L.; Harrold, M. J.; Kim, J.; Porter, A.; Rothermel, G. An Empirical Study of Regression Test Selection Techniques. In: ACM Transaction on Software Engineering Methodology, pp. 184-208, v.10, 2001.
- Hentenryck, P.; Deville, Y. The cardinality operator: A new logical connective for constraint logic programming. In: International Conference on Logic Programming (ICLP'91), pp. 745-759, 1991.
- Hervieu, A.; Baudry, B.; Gotlieb, A. Pacogen: Automatic generation of pairwise test configurations from feature models. In: 22nd International Symposium on Software Reliability Engineering (ISSRE), pp. 120-129, 2011.
- Hwang, k.; Faye, A. Computer Architecture and Parallel Processing. McGraw-Hill International Editions, 613 p., 1985.
- Johansen, M. F.; Haugen, Ø.; Fleurey, F. An Algorithm for Generating T-wise Covering Arrays from Large Feature Models. In: 16th International Software Product Line Conference, v. 1. pp. 46-55, 2012.
- Jung, C. F. Metodologia Aplicada a Projetos de Pesquisa: Sistemas de Informação & Ciência da Computação. Taquara. 2009. Disponível em: <<http://www.jung.pro.br/moodle/mod/resource/view.php?id=102>>, Acesso em: 27/11/2014.

- Kang, K.; Cohen, S.; Hess, J.; Nowak, W.; Peterson, S. Feature-Oriented Domain Analysis Feasibility Study. Technical report, no. CMU/SEI-90-TR-21 Carnegie Mellon University, 163p., 1990.
- Kang, K.; Kim, S.; Lee, J.; Kim, K.; Shin E.; Huh, M. FORM: A Feature-Oriented Reuse Method with Domain-Specific Reference Architectures. In: *Annals of Software Engineering*, pp. 143-168, 1998.
- Kang, K.; Lee, J.; Donohoe, P. Feature-Oriented Product Line Engineering. In: *IEEE Software*, v. 19, n. 4, pp. 58-65, 2002.
- Karatas, A.; Oguztuzun, H.; Dogru, A. Global constraints on feature models. In: *Principles and Practice of Constraint Programming*, pp. 537-551, 2010.
- Krueger, C. Variation Management for Software Production Lines. In: *Conference on Software Product Lines (SPLC 02)*, pp. 107-108, 2002.
- Kuhn, D.; Wallace, D. R.; Gallo, A. M. Software Fault Interactions and Implications for Software Testing. In: *IEEE Transactions on Software Engineering*, v. 30, n. 6, p. 418-421, 2004.
- Lee, J.; Muthig, D. Feature-Oriented Variability Management in Product Line Engineering. In: *Communications of the ACM*, v. 49, n. 12, pp.55-59, 2006.
- Lee, K.; Kang, K. C.; Lee, J. Concepts and Guidelines of Feature Modeling for Product Line Software Engineering. In: *Software Reuse: Methods, Techniques, and Tools*. Springer Berlin Heidelberg, v. 2319, pp. 62-77, 2002.
- Linden, F.; Schimid, K.; Rommes, E. *Software Product Lines in Action*. Springer, 329p, 2007.
- Lochau, M.; Oster, S., Goltz, U., Schürr. A. Model-based Pairwise Testing for Feature Interaction Coverage in Software Product Line Engineering. In: *Software Quality Journal*, pp. 567-604, 2012.
- Long, C. A. Software Product Lines: Practices and Patterns. In: *IEEE Software*, v.19(4), pp.131-132, 2002.
- Lopez-Herrejon, R; Ferrer, J.; Haslinger, E.; Chicano, F.; Egyed, A.; Alba, E. Comparing Pairwise Testing in Software Product Lines: A Framework Assessment. In: *International Software Product Line Conference*, 2014.
- Mahajan, Y. S.; Fu; Z. Malik; S. Zchaff2004: An Efficient Sat Solver. In: *Theory and Applications of Satisfiability Testing*, pp. 360-375, 2004
- McGregor, J. D. *A Pratical Guide to Testing Object-Oriented Software*. Addison Wesley, 375p., 2001.
- McGregor, J. *Testing a Software Product Line*. Technical Report, no CMU/SEI-2001-TR-022 Carnegie Mellon University, 2001.

- Mendonça, M.; Branco, M.; Cowan, D. SPLOT: Software Product Lines Online Tools. In: 24th ACM SIGPLAN Conference Companion on Object Oriented Programming Systems Languages and Applications, pp. 761-762, 2009.
- Neto, P., Machado, I., McGregor, J., Almeida, E., Meira, S. A Systematic Mapping Study of Software Product Lines Testing. In: Special Section on Best Papers from XP2010, Information and Software Technology, v. 53, n. 5, pp. 407-423, 2011.
- Oliveira, F. M.; Menna, R. D. S.; Vieira, H. V.; Ruiz, D. D. Performance Testing from UML Models with Resource Descriptions. In: Brazilian Workshop on Systematic and Automated Software Testing, p. 47-54, 2007.
- Oster, S.; Markert, F.; Ritter, P. Automated Incremental Pairwise Testing of Software Product Lines. In: Software Product Lines: Going Beyond, v.6287, p. 196-210, 2010.
- Parnas, D. On the Design and Development of Program Families. In: IEEE Transactions on Software Engineering, v. SE-2, pp. 1-9, 1976.
- Perrouin, G.; Oster, S.; Sen, S.; Klein, J.; Baudry, B.; Le Traon, Y. Pairwise Testing for Software Product Lines: Comparison of Two Approaches. In: Software Quality Journal, v. 20, n. 3-4, pp. 605-643, 2012.
- Perrouin, G.; Sen, S.; Klein, J.; Baudry, B.; Le Traon, Y. Automated and Scalable t-wise Test Case Generation Strategies for Software Product Lines. In: Third International Conference on Software Testing, Verification and Validation, pp. 459-468, 2010.
- Perry, D. E.; Kaiser, G. E. Adequate Testing and Object-Oriented Programming. In: Journal on Object-Oriented Programming, 1990.
- Phadke, M. S.. Quality Engineering Using Robust Design. Prentice Hall PTR, 250 p.,1995.
- Pressman, S. Software Engineering: a Practitioner's Approach. McGraw-Hill, 340p., 2009.
- Pretschner, A.; Philipps, J. Methodological Issues in Model-Based Testing. In: Model-based testing of reactive systems. Springer Berlin Heidelberg, pp. 281-291, 2005.
- Reuys, A.; Reis, S.; Kamsties, E.; Pohl, K. The Scented Method for Testing Software Product Lines. In: Software Product Lines, Springer, pp. 479-520, 2006.
- Robinson, H. Intelligent Test Automation. In: Software Testing & Quality Engineering, pp. 24-32, 2000.
- Saake, G.; Schaefer, I.; Godfrey, M. W. Analysis and Removal of Code Clones in Software Product Lines. University of Magdeburg, 163 p., 2012.
- Schmid, K.; John, I. A Customizable Approach to Full Lifecycle Variability Management. In: Science Comput. Program, v. 53, pp. 259-284, 2004.
- Schulze, S.; Thüm, T.; Kuhlemann, M.; Saake, G.. Variant-preserving Refactoring in Feature-oriented Software Product Lines. In: Sixth International Workshop on Variability Modeling of Software-Intensive Systems, pp. 73-81, 2012.

- She, S.; Lotufo, R.; Berger, T.; Wasowski, A.; Czarnecki, K.. The Variability Model of The Linux Kernel. In: VaMoS, v. 10, pp. 45-51, 2010.
- Shepard, T.; Lamb, M.; Kelly, D. More Testing Should be Taught. In: Communications of the ACM, v. 44, n. 6, pp. 103-108, 2001.
- Tevanlinna, A.; Taina, J.; Kauppinen, R. Product Family Testing: a Survey. In: ACM SIGSOFT Software Engineering Notes, v. 29. n. 2.pp. 12-12, 2004.
- Torlak, E.; Jackson, D. Kodkod: A Relational Model Finder. In: Tools and Algorithms for Construction and Analysis of Systems, v. 4424. pp. 632-647, 2007.
- Uzuncaova, E.; Garcia, D.; Khurshid, S.; Batory, D.. Testing Software Product Lines Using Incremental Test Generation. In: 19th International Symposium on Software Reliability Engineering, pp. 249-258, 2008.
- Van Der Linden, F.; Schmid, K.; Rommes, E. Software Product Lines in Action: The Best Industrial Practice in Product Line Engineering. In: Secaucus, 333p. 2007.
- Van Deursen, A.; Klint, P. Domain-specific Language Design Requires Feature Descriptions. In: Journal of Computing and Information Technology, v. 10, n. 1, p. 1-17, 2002.
- Van Gorp, J.; Bosch, J.; Svahnberg, M. On the Notion of Variability in Software Product Lines. In: Conference on Software Architecture, p. 45-54, 2001.
- Wallmüller, E. Software-Qualitäts Management. In: Carl der Praxis, 71p. 2001.
- Wasowski, A.; Mendonça, M.; Czarnecki, K.. SAT-based Analysis of Feature Models is Easy. In: 13th International Software Product Line Conference, pp. 231-240, 2009.
- Weiss, D. M.; Lai, C. T. R. Software Product-Line Engineering: A Family-Based Software Development Process. In: Addison-Wesley, 426p., 1999.
- Weißleder, S. ParTeG (Partition Test Generator). 2009. Disponível em: <<http://parteg.sourceforge.net>>, Acesso em: 27/11/2014.

## ANEXO A - Modelos de Características



### Extra Constraints:

- constraint\_1 : Assembly or ~ Assembling
- constraint\_2 : Warehouse Management or ~ Transfer
- constraint\_3 : Identification or ~ Serial Tracking
- constraint\_4 : Expiration Control or ~ Perishable
- constraint\_5 : Identification or ~ Serialized
- constraint\_6 : Item Associations or ~ Assembly

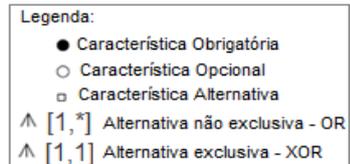
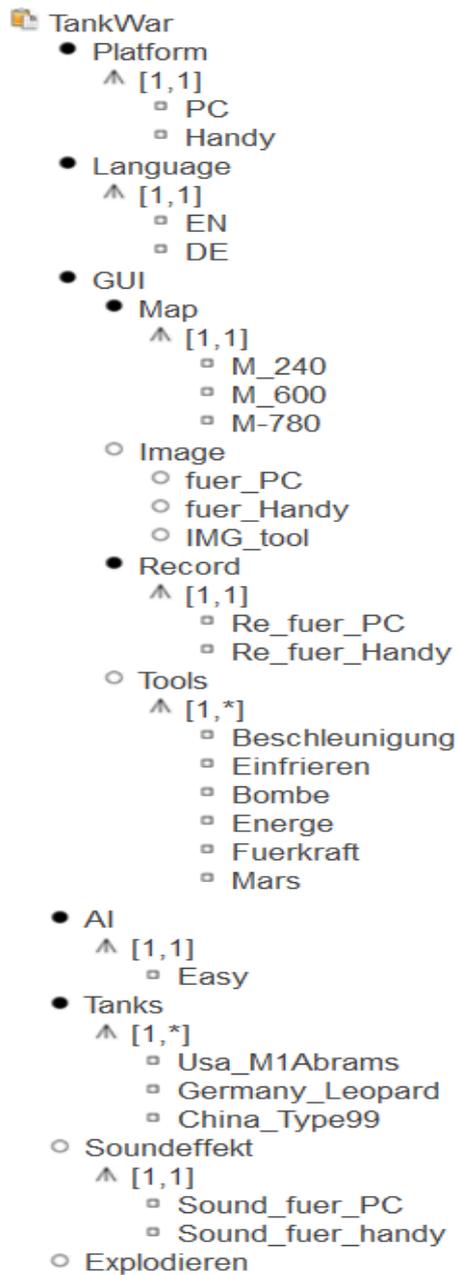


Figura A-1 - Modelo de Características da LPS Inventory (Adaptado: SPLIT<sup>6</sup>)

<sup>6</sup> <http://www.splot-research.org/>



**Extra Constraints:**

- constraint\_1 : ~IMG\_tool or ~ Tools
- constraint\_2 : PC or ~ Re\_fuer\_PC
- constraint\_3 : Handy or ~ Fuer\_Handy
- constraint\_4 : PC or ~ Sound\_fuer\_pc
- constraint\_5 : Handy or ~ Re\_fuer\_Handy
- constraint\_6 : Handy or ~ Sound fuer\_handy
- constraint\_7 : PC or ~ Fuer\_PC
- constraint\_8 : Fuer\_PC or ~ Re\_fuer\_PC
- constraint\_9 : Image or ~ Fuer\_PC

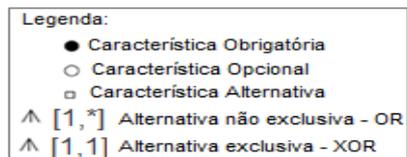


Figura A-2 - Modelo de Características da LPS TankWar (Adaptado: SPLOT<sup>7</sup>)

<sup>7</sup> <http://www.splot-research.org/>

## 📁 FIRE-ALARM SYSTEM

- Devices
  - Input
    - ^ [1,1]
      - Advancel
      - ^ [1,\*]
        - Ultra-sensitivelaser
        - Laser smoke
      - Conventioll
    - ^ [1,\*]
      - Temperature
      - Smoke sensor
  - Output
    - ^ [1,1]
      - AdvanceO
      - ^ [1,\*]
        - Automatic alert
        - High-speed stinguisher system
      - Conventiolo
    - ^ [1,\*]
      - Alarm bell
      - Alarm texts on display
- Configuration
  - ^ [1,1]
    - Low-end systems
    - High-end systems

### Extra Constraints:

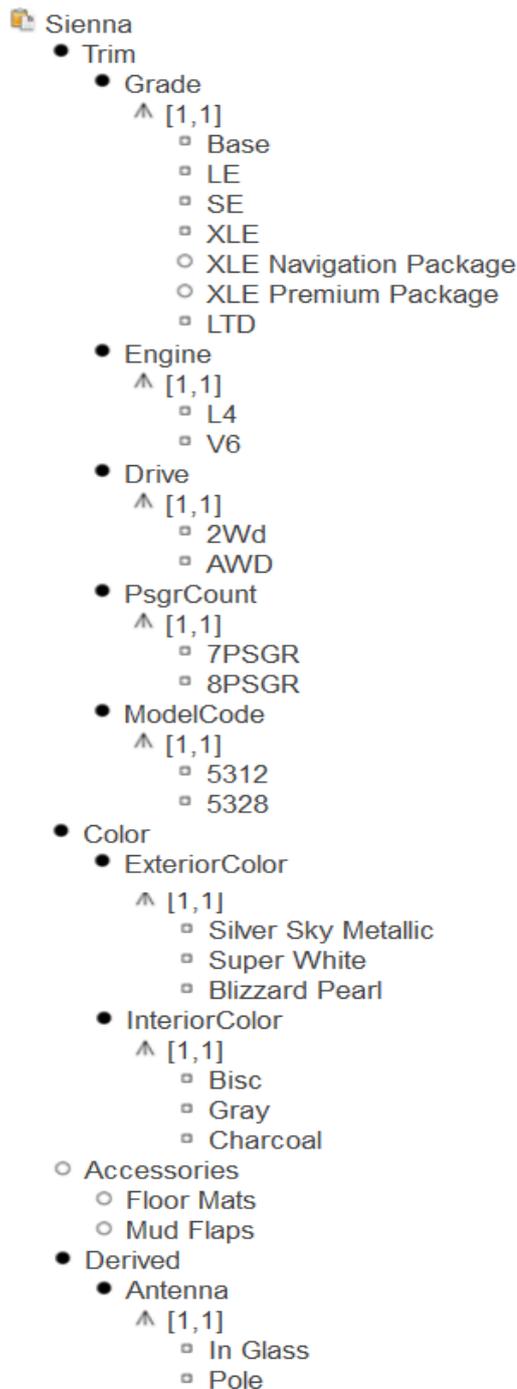
- constraint\_1 : ~Low-end systems or Conventiolo
- constraint\_2 : ~Low-end systems or Conventioll
- constraint\_3 : ~High-end systems or AdvanceO
- constraint\_4 : ~High-end systems or Advancel

| Legenda: |                                |
|----------|--------------------------------|
| ●        | Característica Obrigatória     |
| ○        | Característica Opcional        |
| ▫        | Característica Alternativa     |
| ^ [1,*]  | Alternativa não exclusiva - OR |
| ^ [1,1]  | Alternativa exclusiva - XOR    |

**Figura A-3 - Modelo de Características da LPS Fire Alarm (Adaptado: SPLOT<sup>8</sup>)**

<sup>8</sup> <http://www.splot-research.org/>





**Extra Constraints:**

- constraint\_1 : Base or ~\_5312
- constraint\_2 : In\_Glass or ~ XLE\_Premium\_Package
- constraint\_3 : In\_Glass or ~ XLE\_Navigation\_Package
- constraint\_4 : ~LTD or ~In\_Glass
- constraint\_5 : ~LE or Pole
- constraint\_6 : ~SE or Pole
- constraint\_7 : SE or ~Charcoal
- constraint\_8 : ~Base or Pole

| Legenda: |                                |
|----------|--------------------------------|
| ●        | Característica Obrigatória     |
| ○        | Característica Opcional        |
| □        | Característica Alternativa     |
| ^ [1,*]  | Alternativa não exclusiva - OR |
| ^ [1,1]  | Alternativa exclusiva - XOR    |

**Figura A-5 - Modelo de Características da LPS Sienna Van (Adaptado: SPLOT)**

## ANEXO B - Matrizes de Cobertura Geradas pelos Algoritmos

Tabela B-1 - Matriz de Cobertura da LPS Inventory Gerada por ICPL

| Características      | Produtos |   |   |   |   |   |   |   |   |    |    |    |    |
|----------------------|----------|---|---|---|---|---|---|---|---|----|----|----|----|
|                      | 1        | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| Adjustment           | -        | X | - | X | - | - | X | X | X | X  | -  | X  | -  |
| Allocation           | -        | X | X | X | - | - | X | - | X | X  | -  | -  | -  |
| Assembling           | -        | X | - | - | X | - | X | - | X | X  | -  | -  | -  |
| Assembly             | -        | X | - | X | X | - | X | - | X | X  | -  | -  | -  |
| AVG                  | X        | - | - | - | - | - | X | X | - | X  | -  | -  | -  |
| Bar Code             | -        | X | - | X | - | X | X | X | - | -  | -  | -  | -  |
| Catalog Management   | X        | X | X | X | X | X | X | X | X | X  | X  | X  | X  |
| Categories           | X        | - | - | - | X | X | X | - | - | X  | -  | -  | -  |
| Consignment          | -        | X | - | X | - | - | X | X | X | -  | X  | -  | -  |
| Decrease             | X        | X | - | - | X | X | - | - | - | -  | -  | -  | -  |
| Expiration Control   | -        | X | X | - | X | - | X | X | - | -  | -  | -  | -  |
| FIFO                 | -        | X | - | X | - | X | - | - | - | X  | -  | X  | -  |
| Generic              | X        | - | - | X | X | - | - | - | X | -  | -  | -  | -  |
| Identification       | -        | X | - | X | X | X | X | X | - | X  | X  | -  | -  |
| Increase             | -        | X | - | X | - | - | X | X | X | X  | -  | -  | X  |
| Inventory            | X        | X | X | X | X | X | X | X | X | X  | X  | X  | X  |
| Inventory Method     | X        | X | X | X | X | X | X | X | X | X  | X  | X  | X  |
| Item Associations    | -        | X | - | X | X | X | X | - | X | X  | -  | -  | -  |
| Item Classification  | X        | X | X | X | X | X | X | X | X | X  | X  | X  | X  |
| Items                | X        | X | X | X | X | X | X | X | X | X  | X  | X  | X  |
| LIFO                 | -        | - | X | - | X | - | X | - | - | -  | -  | -  | X  |
| Periodic             | X        | X | - | - | - | - | X | X | - | -  | -  | -  | -  |
| Perishable           | -        | X | X | - | X | - | - | X | - | -  | -  | -  | -  |
| Perpetual            | -        | - | X | X | X | X | - | - | X | X  | X  | X  | X  |
| Physical Counting    | X        | X | X | X | X | X | X | X | X | X  | X  | X  | X  |
| Raw Material         | -        | X | X | X | - | - | X | - | X | -  | X  | X  | X  |
| RFID                 | -        | X | - | X | X | - | - | X | - | X  | X  | -  | -  |
| Serial Tracking      | -        | X | - | - | X | - | X | X | - | X  | -  | -  | -  |
| Serialized           | -        | X | - | X | - | X | X | - | - | X  | X  | -  | -  |
| Services             | -        | - | X | - | X | X | X | X | - | -  | -  | -  | -  |
| Stock Management     | X        | X | X | X | X | X | X | X | X | X  | X  | X  | X  |
| Tags                 | -        | X | X | X | - | X | - | X | X | -  | X  | X  | X  |
| Transactions         | X        | X | X | X | X | X | X | X | X | X  | X  | X  | X  |
| Transfer             | -        | X | - | - | - | X | X | X | X | -  | -  | -  | -  |
| Units Conversion     | X        | X | X | X | X | X | X | X | X | X  | X  | X  | X  |
| Valuation Method     | X        | X | X | X | X | X | X | X | X | X  | X  | X  | X  |
| Warehouse Management | -        | X | - | - | X | X | X | X | X | -  | -  | -  | -  |

Tabela B-2 - Matriz de Cobertura da LPS TankWar Gerada por ICPL

| Características | Produtos |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
|-----------------|----------|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
|                 | 1        | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| AI              | X        | X | X | X | X | X | X | X | X | X  | X  | X  | X  | X  | X  |
| Beschleunigung  | -        | X | X | X | X | - | - | X | - | -  | -  | -  | -  | -  | -  |
| Bombe           | -        | X | - | X | X | X | - | - | X | -  | -  | X  | -  | X  | X  |
| China_Type99    | X        | X | - | - | X | - | - | X | X | -  | -  | -  | -  | -  | -  |
| DE              | X        | - | X | - | X | - | X | - | - | -  | -  | -  | -  | -  | -  |
| Easy            | -        | X | X | - | X | - | - | - | X | X  | X  | X  | X  | X  | X  |
| Einfrieren      | -        | X | - | X | X | X | - | - | - | X  | -  | -  | -  | -  | -  |
| EN              | -        | X | - | X | - | X | - | X | X | X  | X  | X  | X  | X  | X  |
| Energie         | -        | X | - | X | - | X | X | - | - | X  | X  | -  | X  | -  | -  |
| Explodierem     | -        | X | - | X | - | X | X | - | - | X  | X  | -  | -  | X  | -  |
| Feuerkraft      | -        | X | - | X | - | X | X | - | X | X  | -  | -  | -  | -  | -  |
| Fuer Handy      | -        | - | X | - | - | X | - | - | X | -  | -  | -  | X  | -  | -  |
| Fuer PC         | -        | X | - | X | - | - | X | - | - | -  | -  | X  | -  | -  | -  |
| Germany_Leopard | -        | - | X | X | X | X | - | - | - | X  | X  | -  | -  | -  | -  |
| GUI             | X        | X | X | X | X | X | X | X | X | X  | X  | X  | X  | X  | X  |
| Handy           | X        | - | X | - | - | X | - | X | X | -  | X  | -  | X  | X  | -  |
| Hard            | X        | - | - | X | - | X | X | X | - | -  | -  | -  | -  | -  | -  |
| Image           | -        | X | X | X | - | X | X | - | X | -  | -  | X  | X  | -  | -  |
| IMG_tool        | -        | X | - | - | X | - | - | X | - | -  | X  | -  | -  | X  | -  |

**Tabela B-2 - Matriz de Cobertura da LPS TankWar Gerada por ICPL (cont.)**

| Características  | Produtos |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
|------------------|----------|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
|                  | 1        | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| Language         | X        | X | X | X | X | X | X | X | X | X  | X  | X  | X  | X  | X  |
| M_240            | -        | - | X | - | - | X | - | X | - | X  | -  | X  | -  | -  | -  |
| M_600            | -        | - | - | X | X | - | X | - | X | -  | -  | -  | -  | X  | -  |
| M_780            | X        | X | - | - | - | - | - | - | - | -  | X  | -  | X  | -  | X  |
| Map              | X        | X | X | X | X | X | X | X | X | X  | X  | X  | X  | X  | X  |
| Mars             | X        | - | - | X | - | X | X | - | X | -  | -  | -  | -  | -  | -  |
| PC               | -        | X | - | X | X | - | X | - | - | X  | -  | X  | -  | -  | X  |
| Platform         | X        | X | X | X | X | X | X | X | X | X  | X  | X  | X  | X  | X  |
| Re_fuer_Handy    | -        | - | X | - | - | X | - | - | X | -  | X  | -  | -  | -  | -  |
| Re_fuer_PC       | -        | - | - | - | X | - | X | - | - | X  | -  | -  | -  | -  | X  |
| Record           | -        | - | X | - | X | X | X | - | X | X  | X  | -  | -  | -  | X  |
| Sound_fuer_handy | -        | - | X | - | - | X | - | X | - | -  | X  | -  | -  | -  | X  |
| Sound_fuer_pc    | -        | X | - | - | X | - | X | - | - | -  | -  | X  | -  | -  | -  |
| Soundeffekt      | -        | X | X | - | X | X | X | X | - | -  | X  | X  | -  | X  | -  |
| Tanks            | X        | X | X | X | X | X | X | X | X | X  | X  | X  | X  | X  | X  |
| TankWar          | X        | X | X | X | X | X | X | X | X | X  | X  | X  | X  | X  | X  |
| Tools            | X        | X | X | X | X | X | X | X | X | X  | X  | X  | X  | X  | X  |
| USA_M1Abrams     | -        | - | X | X | X | - | X | - | - | -  | -  | X  | X  | X  | X  |

**Tabela B-3 - Matriz de Cobertura da LPS Fire Alarm Gerada por ICPL**

| Características               | Produtos |   |   |   |   |   |   |   |   |    |
|-------------------------------|----------|---|---|---|---|---|---|---|---|----|
|                               | 1        | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Advance 0                     | X        | X | - | X | - | - | - | - | X | X  |
| Advance 1                     | X        | X | - | X | - | - | - | - | X | X  |
| Alarm bell                    | -        | - | X | - | - | X | X | - | - | -  |
| Alarm texts on display        | -        | - | X | - | X | - | - | X | - | -  |
| Automatic alert               | -        | X | - | X | - | - | - | - | - | X  |
| Configuration                 | X        | X | X | X | X | X | X | X | X | X  |
| Conventional 0                | -        | - | X | - | X | X | X | X | - | -  |
| Conventional 1                | -        | - | X | - | X | X | X | X | - | -  |
| Devices                       | X        | X | X | X | X | X | X | X | X | X  |
| Fire Alarm                    | X        | X | X | X | X | X | X | X | X | X  |
| High-end systems              | X        | X | - | X | - | - | - | - | X | X  |
| High-speed stinguisher system | X        | - | - | X | - | - | - | - | X | -  |
| Input                         | X        | X | X | X | X | X | X | X | X | X  |
| Laser smoke                   | -        | X | - | X | - | - | - | - | X | -  |
| Low-end systems               | -        | - | X | - | X | X | X | X | - | -  |
| Output                        | X        | X | X | X | X | X | X | X | X | X  |
| Smoke sensor                  | -        | - | X | - | X | - | X | - | - | -  |
| Temperature                   | -        | - | X | - | - | X | - | X | - | -  |
| Ultra-sensitivelaser          | X        | X | - | - | - | - | - | - | - | X  |

**Tabela B-4 - Matriz de Cobertura da LPS Arcade Gerada por ICPL**

| Características           | Produtos |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |
|---------------------------|----------|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|
|                           | 1        | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| Arcade_game               | X        | X | X | X | X | X | X | X | X | X  | X  | X  | X  | X  | X  | X  | X  | X  |
| Alley_bowling             | -        | X | X | - | - | - | - | X | - | -  | -  | -  | -  | X  | -  | -  | -  | -  |
| Animation_loop            | X        | X | X | X | X | X | X | X | X | X  | X  | X  | X  | X  | X  | X  | X  | X  |
| Ball_bowling              | -        | - | - | X | - | X | - | X | - | -  | -  | -  | X  | X  | -  | -  | -  | -  |
| Board                     | X        | X | X | X | X | X | X | X | X | X  | X  | X  | X  | X  | X  | X  | X  | X  |
| Board_bowling             | -        | - | X | X | - | X | - | X | - | -  | -  | -  | -  | X  | -  | -  | -  | -  |
| Board_brickles            | -        | X | - | - | X | X | X | - | X | -  | X  | -  | X  | -  | -  | -  | X  | X  |
| Board_pong                | X        | - | - | X | X | - | - | X | - | X  | -  | X  | -  | -  | X  | X  | -  | -  |
| Bottom_paddle_pong        | -        | X | - | - | - | X | - | X | - | X  | -  | -  | -  | X  | -  | -  | -  | -  |
| Bowling_game_menu         | -        | - | X | X | - | - | X | X | - | -  | -  | -  | X  | -  | -  | -  | X  | -  |
| Brick_brickles            | -        | X | - | X | X | - | - | - | - | X  | -  | -  | -  | X  | -  | -  | -  | X  |
| Brick_pile_brickles       | -        | X | - | X | X | - | X | - | - | -  | X  | -  | -  | -  | -  | -  | -  | -  |
| Brickles_game_menu        | X        | X | - | - | X | - | X | - | - | -  | X  | -  | X  | X  | -  | -  | -  | -  |
| Ceiling_brickles          | -        | X | - | - | - | - | X | - | - | X  | X  | -  | -  | X  | -  | -  | -  | -  |
| Ceiling_pong              | -        | X | - | X | X | - | - | - | - | X  | -  | X  | -  | -  | -  | -  | -  | -  |
| Check_previous_best_score | -        | - | X | X | X | - | - | - | X | X  | -  | -  | -  | -  | -  | -  | -  | -  |
| Class_diagram             | X        | X | X | X | X | X | X | X | X | X  | X  | X  | X  | X  | X  | X  | X  | X  |
| Dividing_line_pong        | -        | X | - | X | X | - | - | X | - | X  | -  | -  | -  | -  | -  | -  | -  | -  |

**Tabela B-4 - Matriz de Cobertura da LPS Arcade Gerada por ICPL (cont.)**

| Produtos             | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|----------------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|
| Edge_bowling         | - | X | - | X | - | - | X | X | - | -  | -  | -  | -  | -  | -  | -  | -  | -  |
| Exit_game            | X | X | X | X | X | X | X | X | X | X  | X  | X  | X  | X  | X  | X  | X  | X  |
| Floor_brickles       | - | X | - | X | X | - | - | - | - | X  | X  | -  | -  | -  | -  | -  | -  | X  |
| Floor_pong           | - | X | - | X | X | - | - | X | - | -  | -  | X  | -  | -  | -  | -  | -  | -  |
| Game_menu            | X | X | X | X | X | X | X | X | X | X  | X  | X  | X  | X  | X  | X  | X  | X  |
| Game_sprite          | X | X | X | X | X | X | X | X | X | X  | X  | X  | X  | X  | X  | X  | X  | X  |
| Gutter_bowling       | - | X | X | - | - | - | - | X | - | -  | -  | -  | -  | -  | X  | -  | -  | -  |
| Initialization       | X | X | X | X | X | X | X | X | X | X  | X  | X  | X  | X  | X  | X  | X  | X  |
| Install_game         | X | X | X | X | X | X | X | X | X | X  | X  | X  | X  | X  | X  | X  | X  | X  |
| Lane_bowling         | - | - | X | X | - | - | X | - | - | -  | -  | -  | -  | X  | -  | -  | -  | -  |
| Left_brickles        | - | X | - | - | X | - | X | - | - | X  | -  | -  | -  | X  | -  | -  | X  | -  |
| Left_wall_pong       | - | X | - | - | X | - | - | X | - | -  | -  | X  | -  | X  | -  | -  | -  | -  |
| List_game            | X | X | X | X | X | X | X | X | X | X  | X  | X  | X  | X  | X  | X  | X  | X  |
| Menu                 | X | X | X | X | X | X | X | X | X | X  | X  | X  | X  | X  | X  | X  | X  | X  |
| Menu_bowling         | - | X | X | X | - | X | X | X | - | -  | -  | -  | X  | X  | X  | -  | X  | -  |
| Menu_brickles        | X | X | - | X | X | X | X | - | X | X  | X  | -  | X  | X  | -  | -  | X  | X  |
| Menu_pong            | X | X | - | X | X | X | - | X | X | X  | -  | X  | -  | X  | X  | X  | -  | X  |
| Movable_sprite       | - | X | - | X | - | X | - | X | X | X  | X  | -  | X  | X  | X  | X  | -  | -  |
| Paddle               | - | X | - | X | - | X | - | X | X | X  | -  | -  | -  | X  | -  | -  | -  | -  |
| Pin_bowling          | - | X | - | X | - | X | - | - | - | -  | -  | -  | X  | -  | X  | -  | -  | -  |
| Play_bowling         | - | X | X | X | - | X | X | X | - | -  | -  | -  | -  | X  | X  | X  | -  | X  |
| Play_brickles        | X | X | - | X | X | X | X | - | X | X  | X  | -  | X  | X  | -  | -  | X  | X  |
| Play_game            | X | X | X | X | X | X | X | X | X | X  | X  | X  | X  | X  | X  | X  | X  | X  |
| Play_pong            | X | X | - | X | X | X | - | X | X | X  | -  | X  | -  | X  | X  | X  | -  | X  |
| Point                | X | X | X | X | X | X | X | X | X | X  | X  | X  | X  | X  | X  | X  | X  | X  |
| Pong_game_menu       | - | X | - | X | - | X | - | - | X | X  | -  | X  | -  | -  | X  | X  | -  | X  |
| Puck_core_assets     | - | X | - | X | - | - | - | X | X | -  | X  | -  | -  | -  | -  | -  | X  | -  |
| Puck_supply_brickles | - | X | - | X | X | - | X | - | X | -  | -  | -  | -  | -  | -  | -  | -  | -  |
| Rack_pins_bowling    | - | - | X | X | - | - | X | X | - | -  | -  | -  | -  | -  | X  | -  | -  | -  |
| Rectangle            | X | X | X | X | X | X | X | X | X | X  | X  | X  | X  | X  | X  | X  | X  | X  |
| Right_brickles       | - | X | - | - | X | - | X | - | - | X  | -  | -  | -  | X  | -  | -  | -  | X  |
| Right_wall_pong      | - | X | - | - | X | - | - | X | - | X  | -  | -  | -  | X  | -  | -  | -  | -  |
| Save_game            | X | X | X | X | X | X | X | X | X | X  | X  | X  | X  | X  | X  | X  | X  | X  |
| Save_score           | - | X | X | - | X | - | - | - | X | -  | -  | -  | X  | -  | -  | -  | -  | -  |
| Score_board_pong     | X | X | - | X | - | - | - | X | - | -  | -  | -  | -  | -  | -  | -  | -  | -  |
| Size                 | X | X | X | X | X | X | X | X | X | X  | X  | X  | X  | X  | X  | X  | X  | X  |
| Sprite_pair          | - | X | X | - | X | X | - | - | - | -  | -  | -  | -  | -  | -  | -  | -  | -  |
| Stationary_sprite    | X | X | X | X | X | - | X | X | - | X  | X  | X  | -  | X  | X  | -  | X  | X  |
| Top_paddle_pong      | - | X | - | X | - | - | - | X | X | X  | -  | -  | -  | -  | -  | -  | -  | -  |
| Uninstall_game       | X | X | X | X | X | X | X | X | X | X  | X  | X  | X  | X  | X  | X  | X  | X  |
| Use_cases            | X | X | X | X | X | X | X | X | X | X  | X  | X  | X  | X  | X  | X  | X  | X  |
| Velocity             | - | X | - | X | - | X | - | X | X | X  | -  | X  | X  | X  | X  | X  | -  | -  |
| Wall                 | - | X | - | - | X | - | X | X | - | X  | -  | X  | -  | X  | -  | -  | X  | X  |

**Tabela B-5 - Matriz de Cobertura da LPS Sienna Van Gerada por ICPL**

| Produtos       | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 |
|----------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 2Wd            | X | - | - | X | - | X | - | X | - | -  | X  | X  | -  | -  | -  | -  | -  | -  | -  | -  | -  | -  |
| 5312           | - | X | - | - | - | - | - | X | - | -  | -  | X  | -  | -  | -  | -  | -  | -  | -  | -  | -  | -  |
| 5328           | X | - | X | X | X | X | X | - | X | X  | X  | -  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  |
| 7PSGR          | - | - | X | X | X | - | - | X | X | X  | -  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  |
| 8PSGR          | X | X | - | - | - | X | X | - | - | -  | X  | -  | -  | -  | -  | -  | -  | -  | -  | -  | -  | -  |
| Accessories    | - | X | X | X | X | - | - | - | X | X  | -  | X  | -  | -  | X  | -  | X  | -  | -  | X  | -  | X  |
| Antenna        | X | X | X | X | X | X | X | X | X | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  |
| AWD            | - | X | X | - | X | - | X | - | X | X  | -  | -  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  |
| Base           | - | X | - | - | - | - | - | X | - | -  | -  | X  | X  | -  | -  | -  | -  | -  | -  | -  | -  | -  |
| Bisc           | X | X | - | - | X | - | X | - | - | -  | -  | -  | X  | -  | -  | X  | -  | -  | -  | -  | -  | -  |
| Blizzard Pearl | - | X | - | X | - | - | X | - | - | -  | X  | -  | -  | X  | X  | X  | -  | X  | -  | -  | -  | -  |
| Charcoal       | - | - | - | - | - | X | - | - | - | X  | -  | -  | -  | X  | -  | -  | -  | -  | -  | -  | -  | -  |
| Color          | X | X | X | X | X | X | X | X | X | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  |
| Derived        | X | X | X | X | X | X | X | X | X | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  |
| Drive          | X | X | X | X | X | X | X | X | X | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  |
| Engine         | X | X | X | X | X | X | X | X | X | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  |
| ExteriorColor  | X | X | X | X | X | X | X | X | X | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  |

**Tabela B-5 - Matriz de Cobertura da LPS Sienna Van Gerada por ICPL (cont.)**

| Produtos            | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 |   |
|---------------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|---|
| Floor Mats          | - | X | X | - | X | - | - | - | - | X  | -  | X  | -  | -  | X  | -  | X  | -  | -  | -  | -  | -  |   |
| Grade               | X | X | X | X | X | X | X | X | X | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X |
| Gray                | - | - | X | X | - | - | - | X | X | -  | X  | X  | -  | -  | X  | -  | X  | X  | X  | X  | X  | X  |   |
| In Glass            | X | - | - | - | X | - | - | - | X | -  | X  | -  | -  | -  | X  | -  | -  | -  | -  | X  | X  | -  | X |
| InteriorColor       | X | X | X | X | X | X | X | X | X | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X |
| L4                  | - | X | - | X | X | - | - | X | X | X  | -  | -  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X |
| LE                  | - | - | - | X | - | - | X | - | - | -  | -  | -  | -  | -  | -  | -  | X  | -  | -  | -  | X  | -  |   |
| LTD                 | X | - | - | - | - | - | - | - | X | -  | -  | -  | -  | -  | -  | X  | -  | -  | -  | X  | -  | -  |   |
| ModelCode           | X | X | X | X | X | X | X | X | X | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X |
| Mud Flaps           | - | X | X | - | - | - | - | - | X | X  | -  | X  | -  | -  | -  | -  | X  | -  | -  | X  | -  | X  |   |
| Navigation Package  | - | - | - | - | X | - | - | - | - | -  | X  | -  | -  | -  | -  | -  | -  | -  | -  | -  | -  | X  |   |
| Pole                | - | X | X | X | - | X | X | X | - | X  | -  | X  | X  | X  | -  | X  | X  | X  | -  | -  | X  | -  |   |
| PsgrCount           | X | X | X | X | X | X | X | X | X | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X |
| SE                  | - | - | - | - | - | X | - | - | - | X  | -  | -  | -  | X  | -  | X  | -  | X  | -  | -  | -  | -  |   |
| Sienna              | X | X | X | X | X | X | X | X | X | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X |
| Silver Sky Metallic | X | - | - | - | X | - | - | - | X | X  | -  | X  | -  | -  | -  | -  | -  | -  | -  | -  | -  | X  | - |
| Super White         | - | - | X | - | - | X | - | X | - | -  | -  | -  | X  | -  | -  | -  | X  | -  | X  | X  | -  | X  |   |
| Trim                | X | X | X | X | X | X | X | X | X | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X |
| V6                  | X | - | X | - | - | X | X | - | - | -  | X  | X  | -  | -  | -  | -  | -  | -  | -  | -  | -  | -  | - |
| XLE                 | - | - | X | - | X | - | - | - | - | -  | X  | -  | -  | -  | -  | -  | -  | -  | -  | -  | X  | -  | X |
| XLE Premium Package | - | - | - | - | X | - | - | - | - | -  | X  | -  | -  | -  | -  | -  | -  | -  | -  | -  | X  | -  | - |

**Tabela B-6 - Matriz de Cobertura LPS Inventory Gerada por Pacogen**

| Produtos             | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|----------------------|---|---|---|---|---|---|---|---|---|----|----|----|
| Adjustment           | X | X | X | - | - | - | X | X | X | -  | X  | -  |
| Allocation           | X | - | X | - | - | X | X | X | - | -  | X  | -  |
| Assembling           | X | - | - | - | - | - | X | X | - | X  | X  | -  |
| Assembly             | X | - | X | - | - | - | X | X | - | X  | X  | -  |
| AVG                  | - | - | - | X | - | - | - | - | X | -  | X  | X  |
| Bar Code             | X | - | X | - | X | - | X | - | X | -  | -  | -  |
| Catalog Management   | X | X | X | X | X | X | X | X | X | X  | X  | X  |
| Categories           | - | - | - | X | X | - | X | X | - | X  | -  | -  |
| Consignment          | X | - | X | - | - | - | X | - | X | -  | X  | X  |
| Decrease             | X | - | - | X | X | - | X | - | - | -  | -  | -  |
| Expiration Control   | X | - | - | - | - | X | X | - | X | X  | -  | -  |
| FIFO                 | X | X | X | - | X | - | - | X | - | -  | -  | -  |
| Generic              | - | - | X | X | - | - | - | - | - | X  | X  | -  |
| Identification       | X | - | X | - | X | - | X | X | X | X  | -  | X  |
| Increase             | X | - | X | - | - | - | X | X | X | -  | X  | -  |
| Inventory            | X | X | X | X | X | X | X | X | X | X  | X  | X  |
| Inventory Method     | X | X | X | X | X | X | X | X | X | X  | X  | X  |
| Item Associations    | X | - | X | - | X | - | X | X | - | X  | X  | -  |
| Item Classification  | X | X | X | X | X | X | X | X | X | X  | X  | X  |
| Items                | X | X | X | X | X | X | X | X | X | X  | X  | X  |
| LIFO                 | - | - | - | - | - | X | X | - | - | X  | -  | -  |
| Periodic             | X | - | - | X | - | - | X | - | X | -  | -  | -  |
| Perishable           | X | - | - | - | - | X | - | - | X | X  | -  | -  |
| Perpetual            | - | X | X | - | X | X | - | X | - | X  | X  | X  |
| Physical Counting    | X | X | X | X | X | X | X | X | X | X  | X  | X  |
| Raw Material         | X | X | X | - | - | X | X | - | - | -  | X  | X  |
| RFID                 | X | - | X | - | - | - | - | X | X | X  | -  | X  |
| Serial Tracking      | X | - | - | - | - | - | X | X | X | X  | -  | -  |
| Serialized           | X | - | X | - | X | - | X | X | - | -  | -  | X  |
| Services             | - | - | - | - | X | X | X | - | X | X  | -  | -  |
| Stock Management     | X | X | X | X | X | X | X | X | X | X  | X  | X  |
| Tags                 | X | X | X | - | X | X | - | - | X | -  | X  | X  |
| Transactions         | X | X | X | X | X | X | X | X | X | X  | X  | X  |
| Transfer             | X | - | - | - | X | - | X | - | X | -  | X  | -  |
| Units Conversion     | X | X | X | X | X | X | X | X | X | X  | X  | X  |
| Valuation Method     | X | X | X | X | X | X | X | X | X | X  | X  | X  |
| Warehouse Management | X | - | - | - | X | - | X | - | X | X  | X  | -  |

**Tabela B-7 - Matriz de Cobertura LPS TankWar Gerada por Pacogen**

| Características \ Produtos | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|----------------------------|---|---|---|---|---|---|---|---|---|----|----|
| AI                         | X | X | X | X | X | X | X | X | X | X  | X  |
| Beschleunigung             | X | - | X | X | - | X | - | - | X | X  | -  |
| Bombe                      | - | - | - | X | X | X | X | X | X | X  | -  |
| China_Type99               | - | - | X | - | - | X | X | - | - | X  | -  |
| DE                         | X | - | - | - | - | X | - | - | - | -  | -  |
| Easy                       | - | X | - | X | X | X | X | X | - | X  | X  |
| Einfrieren                 | - | X | - | - | - | X | X | - | X | -  | -  |
| EN                         | - | X | X | X | X | - | X | X | X | X  | X  |
| Energie                    | X | X | - | - | - | - | X | - | X | -  | X  |
| Explodierem                | X | X | - | X | - | - | X | - | X | -  | X  |
| Feuerkraft                 | X | X | - | - | - | - | X | - | X | X  | -  |
| Fuer_Handy                 | - | - | - | - | - | - | - | - | X | X  | -  |
| Fuer_PC                    | X | - | - | - | - | - | X | X | - | -  | -  |
| Germany_Leopard            | - | X | - | - | - | X | - | - | X | -  | X  |
| GUI                        | X | X | X | X | X | X | X | X | X | X  | X  |
| Handy                      | - | - | X | X | - | - | - | - | X | X  | X  |
| Hard                       | X | - | X | - | - | - | - | - | X | -  | -  |
| Image                      | X | - | - | - | - | - | X | X | X | X  | -  |
| IMG_tool                   | - | - | X | X | - | X | X | - | - | -  | X  |
| Language                   | X | X | X | X | X | X | X | X | X | X  | X  |
| M_240                      | - | X | X | - | - | - | - | X | X | -  | -  |
| M_600                      | X | - | - | X | - | X | - | - | - | X  | -  |
| M_780                      | - | - | - | - | X | - | X | - | - | -  | X  |
| Map                        | X | X | X | X | X | X | X | X | X | X  | X  |
| Mars                       | X | - | - | - | - | - | - | - | X | X  | -  |
| PC                         | X | X | - | - | X | X | X | X | - | -  | -  |
| Platform                   | X | X | X | X | X | X | X | X | X | X  | X  |
| Re_fuer_Handy              | - | - | - | - | - | - | - | - | X | X  | X  |
| Re_fuer_PC                 | X | X | - | - | X | X | - | - | - | -  | -  |
| Record                     | X | X | - | - | X | X | - | - | X | X  | X  |
| Sound_fuer_handy           | - | - | X | X | - | - | - | - | X | -  | X  |
| Sound_fuer_pc              | X | - | - | - | - | X | X | X | - | -  | -  |
| Soundeffekt                | X | - | X | X | - | X | X | X | X | -  | X  |
| Tanks                      | X | X | X | X | X | X | X | X | X | X  | X  |
| TankWar                    | X | X | X | X | X | X | X | X | X | X  | X  |
| Tools                      | X | X | X | X | X | X | X | X | X | X  | X  |
| USA_M1Abrams               | X | - | - | X | X | X | - | X | - | -  | X  |

**Tabela B-8 - Matriz de Cobertura da LPS Fire Alarm Gerada por Pacogen**

| Características \ Produtos    | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|-------------------------------|---|---|---|---|---|---|---|
| Advance 0                     | X | - | - | X | - | - | - |
| Advance 1                     | X | - | - | X | - | - | - |
| Alarm bell                    | - | - | X | - | - | X | X |
| Alarm texts on display        | - | X | X | - | X | - | - |
| Automatic alert               | X | - | - | - | - | - | - |
| Configuration                 | X | X | X | X | X | X | X |
| Conventional 0                | - | X | X | - | X | X | X |
| Conventional 1                | - | X | X | - | X | X | X |
| Devices                       | X | X | X | X | X | X | X |
| Fire Alarm                    | X | X | X | X | X | X | X |
| High-end systems              | X | - | - | X | - | - | - |
| High-speed stinguisher system | - | - | - | X | - | - | - |
| Input                         | X | X | X | X | X | X | X |
| Laser smoke                   | X | - | - | - | - | - | - |
| Low-end systems               | - | X | X | - | X | X | X |
| Output                        | X | X | X | X | X | X | X |
| Smoke sensor                  | - | - | X | - | X | - | X |
| Temperature                   | - | X | X | - | - | X | - |
| Ultra-sensitivelaser          | X | - | - | X | - | - | - |

Tabela B-9 - Matriz de Cobertura LPS Arcade Game Gerada por Pacogen

| Características           | Produtos |   |   |   |   |   |   |   |   |    |    |    |    |
|---------------------------|----------|---|---|---|---|---|---|---|---|----|----|----|----|
|                           | 1        | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| alley bowling             | X        | X | - | X | - | X | - | - | - | -  | -  | -  | -  |
| Animation_Loop            | X        | X | X | X | X | X | X | X | X | X  | X  | X  | X  |
| Arcade_Game               | X        | X | X | X | X | X | X | X | X | X  | X  | X  | X  |
| ball bowling              | X        | X | - | - | - | - | - | X | X | -  | -  | -  | -  |
| Board                     | X        | X | X | X | X | X | X | X | X | X  | X  | X  | X  |
| board bowling             | X        | X | - | - | - | X | - | X | X | -  | -  | -  | -  |
| board_brickles            | -        | - | - | X | - | - | X | - | X | X  | X  | X  | -  |
| board_pong                | X        | - | X | - | X | - | - | X | - | -  | -  | -  | X  |
| bottom_paddle_pong        | X        | X | X | X | - | - | - | - | X | -  | -  | -  | -  |
| bowling_game_menu         | X        | - | - | - | - | X | - | X | - | X  | X  | -  | -  |
| brick_brickles            | -        | X | X | X | - | - | - | X | - | X  | -  | -  | -  |
| brick_pile_brickles       | -        | - | - | X | - | - | - | X | - | -  | X  | X  | -  |
| brickles_game_menu        | -        | X | - | X | - | - | - | - | - | -  | X  | X  | -  |
| ceiling_brickles          | -        | X | X | X | - | - | - | - | - | -  | X  | X  | -  |
| ceiling_pong              | -        | - | X | X | - | - | - | X | - | -  | -  | -  | X  |
| check_previous_best_score | -        | - | X | - | - | X | X | X | - | -  | -  | -  | -  |
| Class_Diagram             | X        | X | X | X | X | X | X | X | X | X  | X  | X  | X  |
| dividing_line_pong        | X        | - | X | X | - | - | - | X | - | -  | -  | -  | -  |
| edge_bowling              | X        | - | - | X | - | - | - | X | - | -  | X  | -  | -  |
| Exit_Game                 | X        | X | X | X | X | X | X | X | X | X  | X  | X  | X  |
| floor_brickles            | -        | - | X | X | - | - | - | X | - | -  | -  | -  | X  |
| floor_pong                | X        | - | - | X | - | - | - | X | - | -  | -  | -  | X  |
| Game_Menu                 | X        | X | X | X | X | X | X | X | X | X  | X  | X  | X  |
| game_sprite               | X        | X | X | X | X | X | X | X | X | X  | X  | X  | X  |
| gutter_bowling            | X        | - | - | X | X | X | - | - | - | -  | -  | -  | -  |
| Initialization            | X        | X | X | X | X | X | X | X | X | X  | X  | X  | X  |
| Install_Game              | X        | X | X | X | X | X | X | X | X | X  | X  | X  | X  |
| lane_bowling              | -        | X | - | - | - | X | - | X | - | -  | X  | -  | -  |
| left_brickles             | -        | X | X | X | - | - | - | - | - | X  | X  | -  | -  |
| left_wall_pong            | X        | X | - | X | - | - | - | - | - | -  | -  | -  | X  |
| List_Game                 | X        | X | X | X | X | X | X | X | X | X  | X  | X  | X  |
| Menu                      | X        | X | X | X | X | X | X | X | X | X  | X  | X  | X  |
| menu_bowling              | X        | X | - | X | X | X | - | X | X | X  | X  | -  | -  |
| menu_brickles             | -        | X | X | X | - | - | X | X | X | X  | X  | X  | -  |
| menu_pong                 | X        | X | X | X | X | - | X | X | X | -  | -  | -  | X  |
| Movable_Sprite            | X        | X | X | X | X | - | X | X | X | -  | -  | -  | X  |
| Paddle                    | X        | X | X | X | - | - | X | X | X | -  | -  | -  | -  |
| pin_bowling               | -        | - | - | X | X | - | - | X | X | -  | -  | -  | -  |
| play_bowling              | X        | X | - | X | X | X | - | X | X | X  | X  | -  | -  |
| play_brickles             | -        | X | X | X | - | - | X | X | X | X  | X  | X  | -  |
| Play_Game                 | X        | X | X | X | X | X | X | X | X | X  | X  | X  | X  |
| play_pong                 | X        | X | X | X | X | - | X | X | X | -  | -  | -  | X  |
| Point                     | X        | X | X | X | X | X | X | X | X | X  | X  | X  | X  |
| pong_game_menu            | -        | - | X | X | X | - | X | X | X | -  | -  | -  | X  |
| puck_core_assets          | X        | - | - | X | - | - | X | X | - | -  | -  | -  | X  |
| puck_supply_brickles      | -        | - | - | X | - | - | X | X | - | -  | X  | -  | -  |
| rack_pins_bowling         | X        | - | - | - | X | X | - | X | - | -  | -  | X  | -  |
| Rectangle                 | X        | X | X | X | X | X | X | X | X | X  | X  | X  | X  |
| right_brickles            | -        | X | X | X | - | - | - | - | - | -  | X  | -  | -  |
| right_wall_pong           | X        | X | X | X | - | - | - | - | - | -  | -  | -  | -  |
| Save_Game                 | X        | X | X | X | X | X | X | X | X | X  | X  | X  | X  |
| save_score                | -        | - | - | X | - | X | X | - | - | -  | -  | -  | -  |
| score_board_pong          | X        | - | - | X | - | - | - | X | - | -  | -  | -  | -  |
| Size                      | X        | X | X | X | X | X | X | X | X | X  | X  | X  | X  |
| sprite_pair               | -        | - | - | X | - | X | - | - | X | -  | -  | -  | -  |
| Stationary_Sprite         | X        | X | X | X | X | X | - | X | - | X  | X  | X  | X  |
| top_paddle_pong           | X        | - | X | X | - | - | X | X | - | -  | -  | -  | -  |
| Uninstall_Game            | X        | X | X | X | X | X | X | X | X | X  | X  | X  | X  |
| Use_Cases                 | X        | X | X | X | X | X | X | X | X | X  | X  | X  | X  |
| Velocity                  | X        | X | X | X | X | - | X | X | X | -  | -  | X  | -  |
| Wall                      | X        | X | X | X | - | - | - | - | - | X  | X  | -  | X  |

**Tabela B-10 - Matriz de Cobertura da LPS Sienna Van Gerada por Pacogen**

| Produtos            | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|---------------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|
| 2Wd                 | X | - | - | - | X | - | - | - | - | -  | -  | -  | -  | -  | X  | -  | X  | -  | X  | -  |
| 5312                | X | - | - | - | - | - | - | - | - | -  | -  | -  | X  | -  | -  | -  | -  | -  | -  | -  |
| 5328                | - | X | X | X | X | X | X | X | X | X  | X  | X  | -  | X  | X  | X  | X  | X  | X  | X  |
| 7PSGR               | X | X | X | X | X | X | X | - | X | X  | X  | X  | -  | X  | -  | X  | -  | X  | -  | X  |
| 8PSGR               | - | - | - | - | - | - | - | X | - | -  | -  | -  | X  | -  | X  | -  | X  | -  | X  | -  |
| Accessories         | X | X | - | - | X | X | X | - | - | X  | X  | -  | X  | X  | -  | -  | -  | X  | -  | -  |
| Antenna             | X | X | X | X | X | X | X | X | X | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  |
| AWD                 | - | X | X | X | - | X | X | X | X | X  | X  | X  | X  | X  | -  | X  | -  | X  | -  | X  |
| Base                | X | - | - | - | - | - | - | - | - | -  | X  | X  | -  | -  | -  | -  | -  | -  | -  | -  |
| Bisc                | - | - | - | X | - | X | - | X | - | -  | -  | X  | X  | -  | -  | -  | X  | -  | -  | -  |
| Blizzard Pearl      | - | - | - | X | X | - | X | X | X | -  | -  | -  | X  | -  | -  | X  | -  | -  | X  | -  |
| Charcoal            | - | - | - | - | - | - | - | - | X | -  | X  | -  | -  | -  | X  | -  | -  | -  | -  | -  |
| Color               | X | X | X | X | X | X | X | X | X | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  |
| Derived             | X | X | X | X | X | X | X | X | X | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  |
| Drive               | X | X | X | X | X | X | X | X | X | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  |
| Engine              | X | X | X | X | X | X | X | X | X | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  |
| ExteriorColor       | X | X | X | X | X | X | X | X | X | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  |
| Floor Mats          | X | X | - | - | - | X | X | - | - | -  | X  | -  | X  | X  | -  | -  | -  | -  | -  | -  |
| Grade               | X | X | X | X | X | X | X | X | X | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  |
| Gray                | X | X | X | - | X | - | X | - | - | X  | -  | -  | -  | X  | -  | X  | -  | X  | X  | X  |
| In Glass            | - | - | - | - | - | X | X | - | - | X  | -  | -  | -  | -  | -  | -  | X  | X  | X  | X  |
| InteriorColor       | X | X | X | X | X | X | X | X | X | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  |
| L4                  | - | - | X | X | X | X | X | - | X | X  | X  | X  | X  | X  | -  | X  | -  | X  | -  | X  |
| LE                  | - | - | X | - | X | - | - | X | - | -  | -  | -  | -  | X  | -  | -  | -  | -  | -  | -  |
| LTD                 | - | - | - | - | - | - | X | - | - | X  | -  | -  | -  | -  | -  | -  | X  | -  | -  | X  |
| ModelCode           | X | X | X | X | X | X | X | X | X | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  |
| Mud Flaps           | X | X | - | - | - | - | - | - | - | X  | X  | -  | X  | X  | -  | -  | -  | X  | -  | -  |
| Navigation Package  | - | - | - | - | - | X | - | - | - | -  | -  | -  | -  | -  | -  | -  | -  | -  | X  | -  |
| Pole                | X | X | X | X | X | - | - | X | X | -  | X  | X  | X  | X  | X  | X  | -  | -  | -  | -  |
| PogrCount           | X | X | X | X | X | X | X | X | X | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  |
| SE                  | - | - | - | X | - | - | - | - | X | -  | X  | -  | -  | -  | X  | X  | -  | -  | -  | -  |
| Sienna              | X | X | X | X | X | X | X | X | X | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  |
| Silver Sky Metallic | X | - | X | - | - | X | - | - | - | X  | X  | -  | -  | -  | -  | -  | X  | -  | -  | -  |
| Super White         | - | X | - | - | - | - | - | - | - | -  | X  | -  | X  | X  | -  | -  | X  | -  | X  | -  |
| Trim                | X | X | X | X | X | X | X | X | X | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  |
| V6                  | X | X | - | - | - | - | X | - | - | -  | -  | -  | -  | -  | X  | -  | X  | -  | X  | -  |
| XLE                 | - | X | - | - | - | X | - | - | - | -  | -  | -  | -  | -  | -  | -  | -  | X  | X  | -  |
| XLE Premium Package | - | - | - | - | - | X | - | - | - | -  | -  | -  | -  | -  | -  | -  | -  | X  | X  | -  |

## ANEXO C - Tabelas de Frequência e Ocorrência

**Tabela C-1 - Frequência e Ocorrência das t-Tuplas da LPS Inventory**

| Tuplas  | Frequência Pacogen | Ocorrência Pacogen | Frequência ICPL | Ocorrência ICPL |
|---|--------------------|--------------------|-----------------|-----------------|
| Catalog Management, Item Associations                 | 7                  | 0.4                | 7               | 0.5             |
| Catalog Management, Item Associations, Services       | 3                  | 0.2                | 3               | 0.2             |
| Identification, RFID, Bar Code                        | 3                  | 0.2                | 3               | 0.2             |
| Inventory Method, Periodic                            | 4                  | 0.2                | 4               | 0.3             |
| Inventory Method, Perpetual                           | 9                  | 0.6                | 9               | 0.6             |
| Inventory, Catalog Management                         | 12                 | 1                  | 13              | 1               |
| Inventory, Item Classification                        | 12                 | 1                  | 13              | 1               |
| Inventory, Stock Management                           | 12                 | 1                  | 13              | 1               |
| Item Classification, Categories                       | 5                  | 0.3                | 5               | 0.3             |
| Item Classification, Tags                             | 9                  | 0.6                | 9               | 0.6             |
| Items, Assembly                                       | 6                  | 0.4                | 6               | 0.4             |
| Items, Generic  | 4                  | 0.2                | 4               | 0.3             |
| Items, Perishable                                     | 4                  | 0.2                | 4               | 0.3             |
| Items, Raw Material                                   | 8                  | 0.5                | 8               | 0.6             |
| Items, Serialized                                     | 6                  | 0.4                | 6               | 0.4             |
| Stock Management, Transactions                        | 12                 | 1                  | 13              | 1               |
| Transactions, Adjustment                              | 6                  | 0.4                | 6               | 0.4             |
| Transactions, Allocation                              | 6                  | 0.4                | 6               | 0.4             |
| Transactions, Assembling                              | 5                  | 0.3                | 5               | 0.3             |
| Transactions, Consignment                             | 6                  | 0.4                | 6               | 0.4             |
| Transactions, Decrease                                | 5                  | 0.3                | 5               | 0.3             |
| Transactions, Expiration Control                      | 5                  | 0.3                | 5               | 0.3             |
| Transactions, Increase                                | 4                  | 0.2                | 4               | 0.3             |
| Transactions, Serial Tracking                         | 5                  | 0.3                | 5               | 0.3             |
| Transactions, Transfer                                | 7                  | 0.4                | 7               | 0.5             |
| Transactions, Warehouse Management                    | 6                  | 0.4                | 6               | 0.4             |
| Units Conversion, Physical Counting, Valuation Method | 12                 | 1                  | 13              | 1               |
| Valuation Method, AVG                                 | 4                  | 0.2                | 4               | 0.3             |
| Valuation Method, FIFO                                | 5                  | 0.3                | 5               | 0.3             |
| Valuation Method, LIFO                                | 4                  | 0.2                | 4               | 0.3             |

**Tabela C-2 - Frequência e Ocorrência das t-Tuplas da LPS TankWar**

| Tuplas                        | Frequência Pacogen | Ocorrência Pacogen | Frequência ICPL | Ocorrência ICPL |
|-------------------------------|--------------------|--------------------|-----------------|-----------------|
| AI, Easy                      | 8                  | 0.7                | 10              | 0.6             |
| AI, Hard                      | 3                  | 0.2                | 5               | 0.3             |
| Gui, Fuer_Handy               | 3                  | 0.2                | 4               | 0.2             |
| Gui, Fuer_PC                  | 3                  | 0.2                | 4               | 0.2             |
| Gui, Image                    | 5                  | 0.4                | 8               | 0.5             |
| Gui, IMG_tool                 | 5                  | 0.4                | 5               | 0.3             |
| Language, DE                  | 2                  | 0.1                | 4               | 0.2             |
| Language, EN                  | 9                  | 0.8                | 11              | 0.7             |
| Map, M_240                    | 4                  | 0.3                | 5               | 0.3             |
| Map, M_600                    | 4                  | 0.3                | 5               | 0.3             |
| Map, M_780                    | 3                  | 0.2                | 5               | 0.3             |
| Platform, Handy               | 5                  | 0.4                | 8               | 0.5             |
| Platform, PC                  | 6                  | 0.5                | 7               | 0.4             |
| Record, Re_fuer_Handy         | 3                  | 0.2                | 4               | 0.2             |
| Record, Re_fuer_PC            | 4                  | 0.3                | 4               | 0.2             |
| Soundeffekt, Sound_fuer_handy | 4                  | 0.3                | 5               | 0.3             |
| Soundeffekt, Sound_fuer_pc    | 4                  | 0.3                | 4               | 0.2             |
| Tanks, China_Type99           | 4                  | 0.3                | 5               | 0.3             |
| Tanks, Germany_Leopard        | 4                  | 0.3                | 6               | 0.4             |
| Tanks, USA_M1Abrams           | 6                  | 0.5                | 9               | 0.6             |
| TankWar, Explodierem          | 6                  | 0.5                | 7               | 0.4             |
| TankWar, Platform             | 11                 | 1                  | 15              | 1               |
| TankWar, Language             | 11                 | 1                  | 15              | 1               |
| TankWar, GUI                  | 11                 | 1                  | 15              | 1               |
| TankWar, Map                  | 11                 | 1                  | 15              | 1               |

**Tabela C-2 - Frequência e Ocorrência das t-Tuplas da LPS TankWar (cont.)**

| Tuplas                | Frequência Pacogen | Ocorrência Pacogen | Frequência ICPL | Ocorrência ICPL |
|-----------------------|--------------------|--------------------|-----------------|-----------------|
| TankWar, Tools        | 11                 | 1                  | 15              | 1               |
| TankWar, AI           | 11                 | 1                  | 15              | 1               |
| TankWar, Tanks        | 11                 | 1                  | 15              | 1               |
| TankWar, Record       | 7                  | 0.6                | 8               | 0.5             |
| TankWar, Soundeffekt  | 8                  | 0.7                | 9               | 0.6             |
| Tools, Beschleunigung | 3                  | 0.2                | 5               | 0.3             |
| Tools, Bombe          | 7                  | 0.6                | 8               | 0.5             |
| Tools, Einfrieren     | 4                  | 0.3                | 5               | 0.3             |
| Tools, Energie        | 5                  | 0.4                | 7               | 0.4             |
| Tools, Feuerkraft     | 5                  | 0.4                | 6               | 0.4             |
| Tools, Mars           | 3                  | 0.2                | 5               | 0.3             |

**Tabela C-3 - Frequência e Ocorrência das t-Tuplas da LPS Fire Alarm**

| Tuplas   | Frequência Pacogen | Ocorrência Pacogen | Frequência ICPL | Ocorrência ICPL |
|--|--------------------|--------------------|-----------------|-----------------|
| Configuration, High-end systems                        | 5                  | 0.7                | 5               | 0.5             |
| Configuration, Low-end systems                         | 5                  | 0.7                | 5               | 0.5             |
| Conventional0, Alarm Bell                              | 3                  | 0.4                | 3               | 0.3             |
| Conventional0, Alarm bell, Alarm texts on display      | 1                  | 0.1                | 1               | 0.1             |
| Fire-Alarm System, Input                               | 7                  | 1                  | 10              | 1               |
| Input, Advancel, Ultra-sensitivelaser                  | 3                  | 0.4                | 3               | 0.3             |
| Advancel, Ultra-sensitivelaser                         | 3                  | 0.4                | 3               | 0.3             |
| Input, Conventionall                                   | 3                  | 0.4                | 3               | 0.3             |
| Conventionall, Temperature                             | 3                  | 0.4                | 3               | 0.3             |
| Input, Smoke sensor                                    | 3                  | 0.4                | 3               | 0.3             |
| Output, Advance0                                       | 5                  | 0.7                | 5               | 0.5             |
| Output, Automatic alert, High-speed stinguisher system | 1                  | 0.1                | 1               | 0.1             |

**Tabela C-4 - Frequência e Ocorrência das t-Tuplas da LPS Arcade Game**

| Tuplas  | Frequência Pacogen | Ocorrência Pacogen | Frequência ICPL | Ocorrência ICPL |
|---|--------------------|--------------------|-----------------|-----------------|
| Class_diagram, game_menu                            | 13                 | 1                  | 18              | 1               |
| Class_diagram, game_sprite                          | 13                 | 1                  | 18              | 1               |
| Class_diagram, menu, bowling                        | 9                  | 0.6                | 10              | 0.5             |
| Class_diagram, menu, brickles                       | 10                 | 0.7                | 13              | 0.7             |
| Class_diagram, menu, pong                           | 10                 | 0.7                | 13              | 0.7             |
| Class_diagram, puck supply brickles                 | 5                  | 0.3                | 5               | 0.2             |
| Game_menu, bowling_game_menu                        | 5                  | 0.3                | 6               | 0.3             |
| Game_menu, brickles_game_menu                       | 5                  | 0.3                | 7               | 0.3             |
| Game_menu, pong_game_menu                           | 7                  | 0.5                | 9               | 0.5             |
| Game_sprite, board, bowling_board                   | 5                  | 0.3                | 5               | 0.2             |
| Game_sprite, board, brickles_board                  | 7                  | 0.5                | 9               | 0.5             |
| Game_sprite, board, pong_board                      | 6                  | 0.4                | 8               | 0.4             |
| Game_sprite, movable_sprite                         | 8                  | 0.5                | 11              | 0.6             |
| Game_sprite, rectangle, point                       | 13                 | 1                  | 18              | 1               |
| Game_sprite, rectangle, size                        | 13                 | 1                  | 18              | 1               |
| Game_sprite, sprite_pair                            | 4                  | 0.2                | 4               | 0.2             |
| Game_sprite, stationary_sprite, brick_brickles      | 6                  | 0.4                | 6               | 0.3             |
| Game_sprite, stationary_sprite, brick_pile_brickles | 5                  | 0.3                | 5               | 0.2             |
| Game_sprite, stationary_sprite, ceiling_brickles    | 5                  | 0.3                | 5               | 0.2             |
| Game_sprite, stationary_sprite, ceiling_pong        | 5                  | 0.3                | 5               | 0.2             |
| Game_sprite, stationary_sprite, dividing_line       | 5                  | 0.3                | 5               | 0.2             |
| Game_sprite, stationary_sprite, edge                | 4                  | 0.2                | 4               | 0.2             |
| Game_sprite, stationary_sprite, floor_brickles      | 5                  | 0.3                | 6               | 0.3             |
| Game_sprite, stationary_sprite, floor_pong          | 5                  | 0.3                | 5               | 0.2             |
| Game_sprite, stationary_sprite, gutter              | 4                  | 0.2                | 4               | 0.2             |
| Game_sprite, stationary_sprite, lane                | 4                  | 0.2                | 4               | 0.2             |
| Game_sprite, stationary_sprite, rack_of_pins        | 5                  | 0.3                | 5               | 0.2             |
| Game_sprite, stationary_sprite, score_board         | 3                  | 0.2                | 4               | 0.2             |
| Movable_sprite,velocity, bowling_ball               | 4                  | 0.2                | 5               | 0.2             |
| Movable_sprite,velocity, paddle, bottom_paddle_pong | 5                  | 0.3                | 5               | 0.2             |

**Tabela C-4 - Frequência e Ocorrência das t-Tuplas da LPS Arcade Game (cont.)**

| Tuplas   | Frequência Pacogen | Ocorrência Pacogen | Frequência ICPL | Ocorrência ICPL |
|--|--------------------|--------------------|-----------------|-----------------|
| Movable_sprite,velocity, paddle, top_paddle_pong | 5                  | 0.3                | 5               | 0.2             |
| Movable_sprite,velocity, pin_bowling             | 4                  | 0.2                | 5               | 0.2             |
| Movable_sprite,velocity,puck_core_assets         | 5                  | 0.3                | 6               | 0.3             |
| Use_Cases, check_previous_best_score             | 5                  | 0.3                | 5               | 0.2             |
| Use_Cases, exit_game                             | 13                 | 1                  | 18              | 1               |
| Use_Cases, install_game                          | 13                 | 1                  | 18              | 1               |
| Use_Cases, play_game                             | 13                 | 1                  | 18              | 1               |
| Use_Cases, play_game, animation loop             | 13                 | 1                  | 18              | 1               |
| Use_Cases, play_game, initialization             | 13                 | 1                  | 18              | 1               |
| Use_Cases, play_game, play_bowling               | 9                  | 0.6                | 10              | 0.5             |
| Use_Cases, play_game, play_brickles              | 10                 | 0.7                | 13              | 0.7             |
| Use_Cases, play_game, play_pong                  | 10                 | 0.7                | 13              | 0.7             |
| Use_Cases, save_game                             | 13                 | 1                  | 18              | 1               |
| Use_Cases, save_score                            | 4                  | 0.2                | 5               | 0.2             |
| Use_Cases,list_game                              | 13                 | 1                  | 18              | 1               |
| Use_Cases,uninstall_game                         | 13                 | 1                  | 18              | 1               |

**Tabela C-5 - Frequência e Ocorrência das t-Tuplas da LPS Sienna Van**

| Tuplas                              | Frequência Pacogen | Ocorrência Pacogen | Frequência ICPL | Ocorrência ICPL |
|-------------------------------------|--------------------|--------------------|-----------------|-----------------|
| Acessories, Floor Mats              | 7                  | 0.3                | 7               | 0.3             |
| Acessories, Mud Flaps               | 7                  | 0.3                | 8               | 0.3             |
| Antena, InGlass                     | 7                  | 0.3                | 8               | 0.3             |
| Antena, Pole                        | 13                 | 0.6                | 14              | 0.6             |
| Color, ExteriorColor, InteriorColor | 20                 | 1                  | 22              | 1               |
| Drive, 2Wd                          | 5                  | 0.2                | 6               | 0.2             |
| Drive, AWD                          | 15                 | 0.7                | 16              | 0.7             |
| Engine, L4                          | 14                 | 0.7                | 16              | 0.7             |
| Engine,V6                           | 6                  | 0.3                | 6               | 0.2             |
| ExteriorColor, Blizzard Pearl       | 8                  | 0.4                | 8               | 0.3             |
| ExteriorColor, Silver Sky Metallic  | 6                  | 0.3                | 6               | 0.2             |
| ExteriorColor, Super White          | 6                  | 0.3                | 8               | 0.3             |
| Grade, Base                         | 3                  | 0.1                | 4               | 0.1             |
| Grade, LE                           | 4                  | 0.2                | 4               | 0.1             |
| Grade, LTD                          | 4                  | 0.2                | 4               | 0.1             |
| Grade, SE                           | 5                  | 0.2                | 5               | 0.2             |
| Grade, XLE                          | 4                  | 0.2                | 5               | 0.2             |
| Grade, XLE, Navigation Package      | 2                  | 0.1                | 3               | 0.1             |
| Grade, XLE, XLE Premium Package     | 3                  | 0.1                | 3               | 0.1             |
| InteriorColor, Bisc                 | 6                  | 0.3                | 6               | 0.2             |
| InteriorColor, Charcoal             | 3                  | 0.1                | 3               | 0.1             |
| InteriorColor, Gray                 | 11                 | 0.5                | 13              | 0.5             |
| ModelCode, 5312                     | 2                  | 0.1                | 3               | 0.1             |
| ModelCode, 5328                     | 18                 | 0.9                | 19              | 0.8             |
| PsgrCount, 7PSGR                    | 15                 | 0.7                | 17              | 0.7             |
| PsgrCount, 8PSGR                    | 5                  | 0.2                | 5               | 0.2             |
| Sienna, Acessories                  | 10                 | 0.5                | 11              | 0.5             |
| Sienna, Derived, Antena             | 20                 | 1                  | 22              | 1               |
| Sienna, Drive                       | 20                 | 1                  | 22              | 1               |
| Sienna, Engine                      | 20                 | 1                  | 22              | 1               |
| Sienna, Grade                       | 20                 | 1                  | 22              | 1               |
| Sienna, ModelCode                   | 20                 | 1                  | 22              | 1               |
| Sienna, PsgrCount                   | 20                 | 1                  | 22              | 1               |
| Sienna, Trim                        | 20                 | 1                  | 22              | 1               |