

UNIVERSIDADE ESTADUAL DE MARINGÁ  
CENTRO DE TECNOLOGIA  
DEPARTAMENTO DE INFORMÁTICA  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

JOÃO CHOMA NETO

**Uma Abordagem Memética para Otimizar Projeto de Linha de  
Produto de Software**

Maringá  
2017

JOÃO CHOMA NETO

**Uma Abordagem Memética para Otimizar Projeto de Linha de  
Produto de Software**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Departamento de Informática, Centro de Tecnologia da Universidade Estadual de Maringá, como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

Orientadora: Prof<sup>a</sup>. Dr<sup>a</sup>. Thelma Elita  
Colanzi Lopes

Coorientador: Prof. Dr. Igor Fábio  
Steinmacher

Maringá  
2017

**Dados Internacionais de Catalogação na Publicação (CIP)**  
**(Biblioteca Central - UEM, Maringá, PR, Brasil)**

C548u Choma Neto, João  
Uma abordagem memética para otimizar projeto de  
linha de produto de software / João Choma Neto. --  
Maringá, 2017.  
172 f. : il., color, figs., tabs.

Orientador: Prof<sup>a</sup>. Dr<sup>a</sup>. Thelma Elita Colanzi  
Lopes.

Co-orientador: Prof. Dr. Igor Fábio Steinmacher.  
Dissertação (mestrado) - Universidade Estadual de  
Maringá, Centro de Tecnologia, Departamento de  
Informática, Programa de Pós-Graduação em Ciência da  
Computação, 2017.

1. Software - Arquitetura de linha de produto. 2.  
Busca multiobjetivo. 3. Algoritmo genético. 4.  
Algoritmo memético. I. Lopes, Thelma Elita, orient.  
II. Steinmacher, Igor Fábio, coorient. III.  
Universidade Estadual de Maringá. Centro de  
Tecnologia. Programa de Pós-Graduação em Ciência da  
Computação. IV. Título.

CDD 21.ed. 005.12

AHS

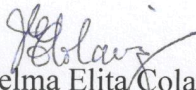
## FOLHA DE APROVAÇÃO

JOÃO CHOMA NETO

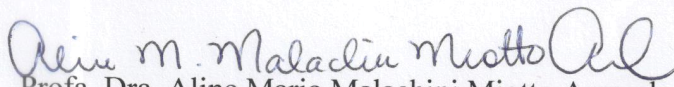
### **Uma abordagem memética para otimizar projeto de linha de produto de software**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Departamento de Informática, Centro de Tecnologia da Universidade Estadual de Maringá, como requisito parcial para obtenção do título de Mestre em Ciência da Computação pela Banca Examinadora composta pelos membros:

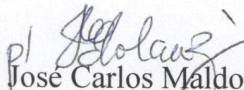
#### BANCA EXAMINADORA



Profa. Dra. Thelma Elita Colanzi Lopes  
Universidade Estadual de Maringá – DIN/UEM



Profa. Dra. Aline Maria Malachini Miotto Amaral  
Universidade Estadual de Maringá – DIN/UEM



Prof. Dr. José Carlos Maldonado  
Universidade de São Paulo – ICMC/USP  
participação por videoconferência

Aprovada em: 31 de julho de 2017.

Local da defesa: Sala 120, Bloco C56, *campus* da Universidade Estadual de Maringá.

Dedico este trabalho a meus pais  
Ivone e Roberto, por toda dedicação,  
paciência e motivação.

## AGRADECIMENTOS

Agradeço primeiramente a Deus pela oportunidade de estudar em uma instituição de excelência e pela força em todos os momentos. Em segundo, agradeço a meus pais, Roberto e Ivone, por todo apoio, paciência e motivação ofertados nos momentos difíceis e felizes desta fase. Além de minha irmã, cunhado e sobrinhos que me animaram nos momentos de dificuldade. Também agradeço a minha noiva por estar sempre ao meu lado.

Agradeço a todos os professores que direta ou indiretamente me auxiliaram neste jornada. Um agradecimento especial vai para minha orientadora Thelma por toda dedicação e paciência na realização deste trabalho e, pela oportunidade oferecida de trabalhar em uma área que me trouxe um fascínio pelo mundo da pesquisa científica.

Por fim, agradeço a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo apoio financeiro concedido a este trabalho.

# Uma Abordagem Memética para Otimizar Projeto de Linha de Produto de Software

## RESUMO

Este trabalho é voltado à aplicação de algoritmos meméticos em projeto de arquitetura de Linha de Produto de Software (LPS). A Arquitetura de Linha de Produto (PLA) é um dos artefatos mais importantes da LPS, pois contém todas as informações necessárias para geração dos produtos da LPS. Construir um projeto de PLA é uma atividade difícil e altamente dependente do arquiteto de software. Dentre os problemas solucionados pela *Search Based Software Engineering* (SBSE), está o problema de otimização de projetos de PLA, que busca encontrar melhores projetos de forma automática utilizando algoritmos de busca multiobjetivo. Neste contexto, a abordagem MOA4PLA foi desenvolvida com objetivo de otimizar princípios básicos de projeto, modularização de características e extensibilidade de LPS de projeto de PLA por meio de algoritmos de busca multiobjetivo. Para automatizar a abordagem MOA4PLA foi desenvolvida a ferramenta OPLA-Tool, onde estão implementados os algoritmos de busca multiobjetivo baseados em algoritmos genéticos (AG). Nesta ferramenta existe um módulo chamado OPLA-Patterns responsável pela aplicação de padrões de projeto durante o processo de otimização. Segundo o mapeamento sistemático realizado, os algoritmos meméticos (AM), que consistem da utilização de AGs com busca local, têm alcançado melhores resultados em problemas de otimização, se comparados ao AG. No entanto, não houve relatos da utilização de algoritmos meméticos para otimização de projeto de PLA. Dessa forma, este trabalho aborda a aplicação de AMs para otimização de projetos de PLAs, adaptando o operador de busca, *Design Pattern Mutation Operator*, proposto no OPLA-Patterns, como operador de busca local. Foram implementadas quatro versões distintas de AM, cada uma com um critério de seleção. Estudos experimentais foram realizados para comparar as soluções obtidas tanto pelo AG como pelas versões do AM. As comparações envolveram análises quantitativa e qualitativa das soluções encontradas. De maneira geral, os estudos quantitativos indicaram que o AM encontrou melhores soluções, em termos de *fitness*, quando comparado ao AG. Já o estudo qualitativo apontou que as soluções obtidas com o AM, no contexto da MOA4PLA, são boas do ponto de vista de arquitetos de software. O AM ainda necessita ser aprimorado com melhorias identificadas durante as etapas deste trabalho.

**Palavras-chave:** Arquitetura de Linha de Produto de Software. Busca multiobjetivo. Algoritmo Genético. Algoritmo Memético.

# A Memetic Approach to Optimize Software Product Line Design

## *ABSTRACT*

This work is focused on the application of memetic algorithms in the Software Product Line (SPL) architecture design. Product Line Architecture (PLA) is one of the most important SPL artifacts since it contains all information needed to generate SPL products. Building a PLA design is a difficult and highly architect-dependent activity. PLA design could be modeled as an optimization problem to be solved by Search Based Software Engineering (SBSE). SBSE aims at automatically obtaining near-optimal solutions using multi-objective search algorithms. In this context, MOA4PLA approach was developed in order to optimize PLA design in terms of basic design principles, feature modularization and SPL extensibility, with the use of multi-objective search algorithms. OPLA-Tool automates MOA4PLA by using multi-objective search algorithms based on genetic algorithms (GA). In this tool there is a module called OPLA-Patterns responsible for applying design patterns during the optimization process. According to the systematic mapping performed, the memetic algorithms (MA), which consist of the use of GAs with local search, have achieved better results in optimization problems when compared with GA. However, there were no report on the use of memetic algorithms for PLA design optimization. Thus, this work deals with the application of MAs to optimize PLA design, adapting the search operator named Design Pattern Mutation Operator proposed in OPLA-Patterns as a local search operator. Four distinct versions of MA were implemented, each one with a different selection criterion. Experimental studies were carried out to compare the solutions obtained by both GA and four versions of MA through quantitative and qualitative analysis. In general, the quantitative studies indicated that the MA found better solutions in terms of fitness when compared with GA. The qualitative study pointed out that the solutions obtained with MA, in the context of MOA4PLA, are good from the software architects point of view. The MA still needs to be refined with improvements identified during the steps of this work.

**Keywords:** Genetic Algorithm. Memetic Algorithm. Multiobjective Search. Product Line Architecture.



## LISTA DE FIGURAS

2.1	Representação da Fronteira de Pareto (Colanzi, 2014). . . . .	24
2.2	Processo da MOA4PLA (Colanzi, 2014). . . . .	28
2.3	Modelo de representação do metamodelo (Colanzi, 2014). . . . .	29
2.4	Representação da Arquitetura da OPLA-Tool (Colanzi, 2014). . .	32
2.5	Funcionamento de satisfação de PS / PS-PLA (Guizzo, 2014). . .	36
4.1	Representação da arquitetura da nova versão da OPLA-Tool. . . .	47
4.2	Classes da OPLA-Tool e do jMetal. . . . .	53
5.1	Atividades realizadas no presente trabalho. . . . .	55
6.1	Projetos Experimentais Exp 01 - Etapa 01. . . . .	65
6.2	Conjunto amostral. . . . .	66
6.3	Projeto Experimental Exp 01 - Etapa 02. . . . .	67
6.4	Exp 01 - Etapa 02 - Gerações 1 - 35 . . . . .	72
6.5	Exp 01 - Etapa 02 - Gerações 36 - 70 . . . . .	73
6.6	Exp 01 - Etapa 02 - Gerações 71 - 100 . . . . .	73
6.7	Exp 01 - Etapa 02 - Reta de regressão. . . . .	74
6.8	Escopo propício à aplicação do padrão <i>Bridge</i> . . . . .	76
6.9	Escopo propício à aplicação do padrão <i>Strategy</i> . . . . .	76
6.10	Padrão <i>Strategy</i> aplicado na solução da MOM obtida pelo NoChoice.	77
6.11	Padrão <i>Strategy</i> aplicado na solução da BANK obtida pelo Bestof2.	78
6.12	$PF_{true}$ AGM. . . . .	79
6.13	$PF_{true}$ BANK. . . . .	80
6.14	Aplicação do padrão <i>Mediator</i> na solução BET do Exp 02 - Etapa 01. . . . .	89
6.15	Aplicação do padrão <i>Strategy</i> na solução MOM do Exp 02 - Etapa 01. . . . .	89
6.16	Exp 03 - Área de atuação dos participantes. . . . .	94
6.17	Exp 03 - Similitude Corpus 1. . . . .	99
6.18	Exp 03 - Nuvem de Palavras Corpus 1. . . . .	100
6.19	Exp 03 - Similitude Corpus 2. . . . .	105
6.20	Exp 03 - Nuvem de Palavras Corpus 2. . . . .	106
6.21	Exp 03 - Similitude Corpus 3. . . . .	108
6.22	Exp 03 - Nuvem de Palavras Corpus 3. . . . .	109

## LISTA DE TABELAS

3.1	Resultados do Mapeamento Sistemático. . . . .	39
3.2	Resumo da Etapa 04 do Mapeamento Sistemático. . . . .	41
5.1	Síntese do método de avaliação. . . . .	58
5.2	Informações das PLAs . . . . .	62
6.1	Exp 01 - Etapa 01 - Teste de Normalidade. . . . .	68
6.2	Exp 01 - Etapa 01 - Tempo de otimização de uma solução em milisegundos. . . . .	69
6.3	Exp 01 - Etapa 01 - Número de soluções não dominadas por algoritmo. . . . .	69
6.4	Exp 01 - Etapa 01 - Média dos valores de HV. . . . .	70
6.5	<i>Fitness</i> das Soluções Original e Ideal. . . . .	70
6.6	Exp 01 - Etapa 01 - Menores EDs por experimento. . . . .	71
6.7	Exp 01 - Etapa 01 - Média dos valores de GD. . . . .	71
6.8	Exp 01 - Etapa 01 - Médias dos valores de IGD. . . . .	71
6.9	Exp 01 - Etapa 03 - PLAs selecionadas - Parte 1 . . . . .	75
6.10	Exp 01 - Etapa 03 - Número de Soluções. . . . .	75
6.11	Exp 02 - Etapa 01 - Teste de Normalidade. . . . .	84
6.12	Exp 02 - Etapa 01 - Resultados do ER. . . . .	84
6.13	Exp 02 - Etapa 01 - Resultados do HV. . . . .	84
6.14	Exp 02 - Etapa 01 - Resultados do IGD. . . . .	85
6.15	Exp 02 - Etapa 01 - <i>Fitness</i> Original e <i>Fitness</i> Ideal. . . . .	85
6.16	Exp 02 - Etapa 01 - Soluções com menor ED. . . . .	85
6.17	Exp 02 - Etapa 02 - Teste de Normalidade. . . . .	86
6.18	Exp 02 - Etapa 02 - Resultados do ER. . . . .	87
6.19	Exp 02 - Etapa 02 - Resultados do HV. . . . .	87
6.20	Exp 02 - Etapa 02 - Resultados do IGD. . . . .	87
6.21	Exp 02 - Etapa 02 - <i>Fitness</i> Original e <i>Fitness</i> Ideal. . . . .	87
6.22	Exp 02 - Etapa 02 - Soluções com menor ED. . . . .	88
6.23	Exp 02 - Etapa 03 - Taxa de aplicação de padrões de projeto no Conjunto de Amostras do Exp 02 - Etapa 01 e 02. . . . .	90
6.24	Exp 03 - Distribuição dos avaliadores. . . . .	92
6.25	Exp 03 - Dendograma Corpus 1. . . . .	95
6.26	Exp 03 - Dendograma Corpus 2. . . . .	101

6.27	Exp 03 - Contabilização de acertos da questão 5 - Corpus 1. . . .	110
6.28	Exp 03 - Contabilização de acertos da questão 3 - Corpus 2. . . .	112
6.29	Exp 03 - Contabilização de acertos da questão 4 - Corpus 2. . . .	113

## LISTA DE SIGLAS E ABREVIATURAS

- ACO:** *Ant Colony Optimization*
- AE:** Algoritmo Evolucionário (*Evolutionary Algorithms*)
- AG:** Algoritmos Genéticos (*Genetic Algorithm*)
- AGM:** Arcade Game Maker
- AM:** Algoritmos Meméticos (*Memetic Algorithm*)
- BANK:** Banking System
- BET:** Bilhetes Eletrônicos para Transporte Urbano
- ED:** Euclidean Distance to the Ideal Solution
- ER:** Error Ratio
- GD:** Generational Distance
- IGD:** Inverse Generational Distance
- HV:** Hypervolume
- LPS:** Linha de Produto de Software
- MOA4PLA:** *Multi-Objective Approach for Product-Line Architecture Design*
- MOM:** Mobile Media
- MS:** Mapeamento Sistemático
- OPLA-Tool:** *Optimization for PLA Tool*
- PLA:** Arquitetura de Linha de Produto (*Product Line Architecture*)
- PS:** *Pattern Application Scope*
- PSO:** *Particle Swarm Optimization*
- PS-PLA:** *Pattern Application Scope in Product Line Architecture*
- SBSE:** *Search Based Software Engineering*

# SUMÁRIO

<b>1</b>	<b>Introdução</b>	<b>14</b>
<b>2</b>	<b>Revisão de literatura</b>	<b>18</b>
2.1	Linha de Produto de Software . . . . .	18
2.1.1	Arquitetura de Linha de Produto . . . . .	21
2.2	<i>Search Based Software Engineering</i> . . . . .	22
2.2.1	Otimização Multiobjetivo . . . . .	23
2.2.2	Algoritmos Genéticos . . . . .	24
2.2.3	Algoritmos Meméticos . . . . .	26
2.3	MOA4PLA . . . . .	27
2.3.1	Modelo de Avaliação . . . . .	30
2.3.2	OPLA-Tool . . . . .	32
2.3.3	Operadores de Busca da MOA4PLA . . . . .	33
<b>3</b>	<b>Trabalhos Relacionados</b>	<b>38</b>
3.1	Resultados do Mapeamento Sistemático . . . . .	39
3.2	Considerações Finais . . . . .	44
<b>4</b>	<b>Abordagem Memética</b>	<b>45</b>
4.1	Algoritmo Proposto . . . . .	46
4.1.1	Algoritmo Memético sem Critério de Escolha (NoChoice) . . . . .	48
4.1.2	Algoritmo Memético First Improvement (Bestof2) . . . . .	48
4.1.3	Algoritmo Memético Best Improvement (Bestof12) . . . . .	49
4.1.4	Algoritmo Memético First Best Improvement (UntilBest) . . . . .	51
4.2	Aspectos de Implementação . . . . .	53
4.3	Considerações Finais . . . . .	54
<b>5</b>	<b>Métodos e Materiais</b>	<b>55</b>
5.1	Método de pesquisa . . . . .	55
5.1.1	Método de avaliação . . . . .	56
5.1.2	Técnicas de análise quantitativa . . . . .	57
5.2	Materiais . . . . .	61
5.2.1	Linguagem e Ferramenta R . . . . .	61
5.2.2	Iramuteq . . . . .	61
5.2.3	PLAs utilizadas . . . . .	62

5.3	Ameaças à Validade . . . . .	62
5.4	Considerações Finais . . . . .	63
<b>6</b>	<b>Resultados e Discussões</b>	<b>64</b>
6.1	Estudo Exploratório - Exp 01 . . . . .	64
6.1.1	Projeto do Estudo . . . . .	64
6.1.2	Resultados . . . . .	67
6.2	Estudo Experimental - Exp 02 . . . . .	81
6.2.1	Projeto do Estudo . . . . .	81
6.2.2	Resultados . . . . .	83
6.3	Estudo Qualitativo - Exp 03 . . . . .	91
6.3.1	Projeto do Estudo . . . . .	91
6.3.2	Resultados . . . . .	94
6.4	Considerações Finais . . . . .	114
<b>7</b>	<b>Conclusão</b>	<b>115</b>
7.1	Dificuldades Encontradas . . . . .	116
7.2	Trabalhos Futuros . . . . .	117
	<b>REFERÊNCIAS</b>	<b>119</b>
<b>A</b>	<b>Protocolo de Mapeamento Sistemático</b>	<b>125</b>
A.1	Planejamento . . . . .	125
A.2	Execução . . . . .	127
<b>B</b>	<b>Documentos utilizados no Exp 03</b>	<b>130</b>
B.1	Documentos . . . . .	130
<b>C</b>	<b>Corpus construídos no Exp 03</b>	<b>163</b>
C.1	Corpus 1 . . . . .	163
C.2	Corpus 2 . . . . .	168
C.3	Corpus 3 . . . . .	172

---

# Introdução

---

A Arquitetura de Linha de Produto de Software (do inglês *Product Line Architecture* (PLA)) contém o projeto comum a todos os produtos de uma Linha de Produto de Software (LPS) (Linden e Rommes, 2007), incluindo todas as variações arquiteturais possíveis nos produtos da LPS. Uma característica da LPS é uma capacidade do sistema que é importante e visível para um usuário (Kang *et al.*, 1990). A PLA é um dos artefatos mais importantes para o sucesso do LPS e um estudo analítico de seu projeto deve ser considerado durante o desenvolvimento da LPS (OliveiraJr *et al.*, 2013a).

O projeto de PLA possibilita prever a qualidade de uma LPS mesmo antes de construí-la. Portanto, é possível realizar uma avaliação estrutural usando métricas. Além das métricas, a avaliação de uma PLA pode levar em consideração fatores econômicos, complexidade e restrições sobre o produto, permitindo a modelagem do projeto de PLA como um problema de otimização multiobjetivo (Colanzi *et al.*, 2014).

A *Search-Based Software Engineering* (SBSE) tenta resolver problemas da engenharia de software por meio da utilização de algoritmos de otimização multiobjetivo (Harman e Mansouri, 2010). Um problema multiobjetivo procura atender a mais de um objetivo envolvendo a otimização simultânea de duas ou mais funções objetivo (Coello *et al.*, 2007).

Concentrando-se em propriedades arquiteturais, os arquitetos precisam desenvolver projetos de PLA considerando medidas arquiteturais diferentes e conflitantes, por exemplo, com alta modularização e baixa difusão de características. Isso faz com que a elaboração de um projeto de uma PLA seja uma tarefa difícil para as pessoas. Este cenário permite modelar o projeto de PLA como um problema de otimização multiobjetivo (Colanzi *et al.*, 2014). Para lidar com esse problema, foi proposta uma abordagem

denominada *Multi-Objective Approach for Product-Line Architecture Design* (MOA4PLA) (Colanzi *et al.*, 2014) para otimizar o projeto de PLA em relação aos princípios básicos de projeto, modularização de características e extensibilidade de LPS, usando algoritmos multiobjetivos baseados em algoritmos genéticos (AG). A ferramenta que automatiza a abordagem MOA4PLA é denominada OPLA-Tool.

Os algoritmos meméticos (AM) foram usados como uma opção de algoritmo de otimização para resolver outros problemas de engenharia de software. AMs são compostos por um AG com busca local (Fraser *et al.*, 2015). O conceito de busca é entendido como um tipo de otimização, portanto, no texto atual, o termo de busca global denota a otimização realizada em uma área de busca cobrindo todo o espaço de soluções, enquanto a busca local diz respeito à otimização de um espaço limitado de soluções.

Estudos existentes aplicam AM no contexto da modelagem de classe (Smith e Simons, 2013) e teste de software (Chawla *et al.*, 2015; Fraser *et al.*, 2015; Harman e McMinn, 2010; Jeya Mala *et al.*, 2013). O AM conseguiu melhores soluções que as obtidas usando AG em todos os trabalhos citados. No entanto, até o fim deste estudo, não foram identificados trabalhos relacionados ao uso de AM para otimização de projeto de PLA.

A hipótese deste trabalho é que: a aplicação de algoritmos meméticos no processo de otimização da MOA4PLA é capaz de obter projetos de PLA melhores do que os projetos obtidos na versão original da MOA4PLA.

Neste contexto, o objetivo geral deste trabalho foi investigar a aplicação de algoritmos meméticos no processo de busca e otimização da MOA4PLA. Como consequência ao objetivo principal, tem-se os seguintes objetivos específicos:

- adaptar o operador *Design Pattern Mutation Operator* desenvolvido no trabalho de Guizzo (2014) como operador de busca local;
- comparar quantitativamente os resultados encontrados na abordagem memética com os resultados da abordagem genética;
- analisar qualitativamente as soluções meméticas, por intermédio de arquitetos de software.

Para o desenvolvimento deste trabalho foi utilizado um método de pesquisa que incorporou a definição do tema de pesquisa, o desenvolvimento do mapeamento sistemático, a definição da proposta da pesquisa, o desenvolvimento da proposta, a definição dos experimentos que avaliaram os algoritmos desenvolvidos, a execução dos experimentos e, por fim, a avaliação dos resultados alcançados.



Durante o desenvolvimento da abordagem memética, foram propostas quatro versões do algoritmo memético: NoChoice, Bestof2, Bestof12 e UntilBest (Capítulo 4), sendo que, cada versão contemplou um critério de seleção de soluções, ou seja, um critério de parada. A abordagem memética foi implementada utilizando o algoritmo NSGA-II como busca global e o operador *Design Pattern Mutation Operator*, desenvolvido por Guizzo *et al.* (2014), como busca local.

Para começar a construir um corpo de conhecimento, foi desenvolvido um estudo exploratório com objetivo de caracterizar a aplicação de AM na otimização de projeto de PLA baseado em busca. Investigou-se duas questões de pesquisa: **QP1:** *O algoritmo memético encontra soluções de melhor qualidade do que as soluções obtidas pelo algoritmo genético no contexto de otimização de projeto de PLA usando a MOA4PLA?* e **QP2:** *Qual abordagem de busca local é mais eficaz no referido contexto?* Os resultados mostraram: (i) qual abordagem de seleção de busca local encontrou melhores resultados e (ii) pontos a serem melhorados na implementação.

Para obter provas para responder a ambas as questões de pesquisa, o estudo exploratório foi dividido em três etapas. Na primeira, realizaram-se experimentos envolvendo as abordagens genética e memética usando projetos de PLA, nos quais a abordagem memética obteve um melhor desempenho. Os resultados foram comparados quantitativamente em termos de indicadores de qualidade multiobjetivos. Na segunda etapa, o estudo exploratório observou o comportamento da busca local durante a evolução das gerações do AM, indicando sua aplicação efetiva em cerca de 30% dos indivíduos durante as evoluções. Finalmente, na terceira etapa do estudo exploratório, foi realizada uma análise qualitativa para avaliar a aplicação de padrões de projeto nas soluções encontradas e, como resultado, foram identificadas aplicações corrompidas.

Para responder à QP2 do estudo exploratório, foi contabilizado o número de vezes que cada algoritmo (NoChoice, Bestof2, Bestof12 e UntilBest) encontrou melhores resultados nos indicadores de qualidade. Os resultados apontaram que o algoritmo Bestof2 obteve melhor desempenho e, também, gastou menos recursos computacionais.

Motivado pelos resultados alcançados no estudo exploratório (Choma *et al.*, 2017), foi realizado um estudo experimental. Neste estudo investigou-se duas questões de pesquisa: **QP1:** *O algoritmo memético encontra soluções de melhor qualidade do que as soluções obtidas pelo algoritmo genético no contexto de otimização de projeto de PLA usando a MOA4PLA?* (mesma questão do estudo exploratório) e **QP3:** *Qual o par de funções objetivo é mais apropriado para avaliar as mudanças realizadas pela busca local nas soluções de projeto de PLA?*

O estudo experimental foi dividido em três etapas: na primeira etapa foi utilizada uma função objetivo para medir princípios básicos de projeto e outra para medir a modularização de características (Colanzi *et al.*, 2014), as mesmas funções objetivo foram usadas em trabalhos relacionados (Guizzo *et al.*, 2014). Uma vez que as funções são compostas pela agregação de métricas, elas podem não ser tão sensíveis às mudanças realizadas por padrões no projeto de PLA. Tendo em conta que os padrões de projeto aplicados pela busca local melhoram a coesão e o acoplamento, realizou-se uma segunda etapa usando uma função objetivo responsável por medir a coesão de uma PLA e outra função para medir o número de elementos arquiteturais que possuem dependências de outras classes. A segunda etapa visa fornecer evidências para responder a questão de pesquisa QP3. Por fim, na terceira etapa realizou-se uma análise qualitativa das soluções encontradas nas duas primeiras etapas. Identificou-se que o segundo par de funções é mais sensível às mudanças causadas pela abordagem memética.

Para atender ao último objetivo deste trabalho foi desenvolvido um estudo qualitativo que envolveu arquitetos de software. O objetivo deste estudo foi avaliar as soluções encontradas pela abordagem memética para responder a seguinte questão de pesquisa: **QP4:** *As soluções encontradas pela OPLA-Tool com o AM são consideradas boas, do ponto de vista de arquitetos de software?* Os resultados indicaram que as soluções encontradas pela abordagem memética são boas do ponto de vista dos arquitetos de software.

O texto está organizado como segue: No Capítulo 2 são definidos os principais conceitos para uma melhor compreensão do trabalho. Já o Capítulo 3 apresenta resultados encontrados no desenvolvimento do MS para investigar trabalhos relacionados. O Capítulo 4 expõe os algoritmos desenvolvidos. O Capítulo 5 apresenta o método de pesquisa, ferramentas e método de avaliação utilizados. No Capítulo 6 explica todos os resultados encontrados nos estudos empíricos desenvolvidos. E, por fim, o Capítulo 7 encerra com as conclusões e contribuições alcançadas, bem como, os trabalhos futuros identificados.

---

## Revisão de literatura

---

Neste Capítulo são apresentados os conceitos necessários para compreensão da proposta do trabalho de dissertação.

### 2.1 Linha de Produto de Software

A maneira de se produzir artefatos e produtos foi alterada com o tempo, onde produtos artesanais com características ajustadas a clientes individuais foram sendo alterados para características genéricas voltadas a um grupo de clientes com as mesmas necessidades. Neste contexto, o reuso é uma base comum da tecnologia proporcionando a ideia de produção em massa, onde um produto de interesse comum pode ser repassado para vários clientes ao mesmo tempo.

A abordagem de produção em massa é trabalhada em inúmeras áreas de construção ou geração de produtos e, não estando de fora, a área de desenvolvimento de software iniciou sua utilização. Combinando uma plataforma comum para um domínio de clientes, surgiu a abordagem de desenvolvimento conhecida como Linha de Produto de Softwares ou Família de Produto de Software (Pohl *et al.*, 2005).

Como qualquer mudança na forma de trabalho, as novas práticas de desenvolvimento em massa da engenharia de software, ocorreram em função de aspectos econômicos. No entanto, a forma de reutilização deve ser planejada com antecedência, pois a criação dos artefatos da LPS interfere no seu ciclo de vida. É necessário um investimento na criação e adaptação dos artefatos da LPS para que os mesmos possam ser reutilizados. Quando uma LPS é criada e cresce, quanto maior o número de produtos, menores serão os custos

de criação, ou seja, quanto maior a LPS vai se tornando, menor é o custo para geração de novos produtos. Essa redução de custo é mais clara se comparada ao desenvolvimento individual de software, onde o atraso com retrabalho na construção de sistemas do mesmo domínio, aumenta o custo de desenvolvimento (Pohl *et al.*, 2005).

O sucesso ou não de um produto pode se dar pelo tempo gasto para geração de um sistema final ao usuário. Dessa forma, o tempo está diretamente relacionado ao custo do produto oferecido ao cliente. A engenharia de LPS pode consumir um tempo considerável no início da construção de uma plataforma de domínio específico (núcleo de artefatos), no entanto, esse tempo é compensado com a geração de um conjunto de produtos. Além disso, permite que o tempo de vida do software seja maior em função da constante atualização dos artefatos na LPS (Pohl *et al.*, 2005).

A garantia de um bom funcionamento dos produtos gerados pela LPS é dada pela validação constante do núcleo de artefato. Quando um novo artefato é inserido ou modificado no núcleo, o conjunto de artefatos é novamente validado, garantindo com isso a estabilidade dos novos produtos da LPS. O constante trabalho de validação permite a detecção de erros e correções em nível global (Garzon-Rodriguez *et al.*, 2015).

Se um projeto de software é desenvolvido e organizado tendo como objetivo o reuso de seus artefatos, o custo de manutenção e extensão desse projeto será reduzido (Coplien, 1999). A separação clara de partes comuns das partes variáveis do sistema, garante que o tempo de desenvolvimento e as taxas de erro sejam reduzidas (Pohl *et al.*, 2005).

O termo característica, também denotado por *feature*, é de grande importância para a LPS uma vez que, representa uma aptidão ou funcionalidade do sistema que é clara e visível ao usuário (Kang *et al.*, 1990). Variabilidade é o termo que denota o modelo de características, entendida como técnica de decisão tardia. Este modelo relaciona as características da LPS e permite distinguir cada um de seus produtos (Pohl *et al.*, 2005). As variabilidades permitem o reuso dos projetos e crescimento da LPS (Schmid e Verlage, 2002).

Os tipos de variabilidades que normalmente ocorrem em LPS são: (Linden *et al.*, 2007):

**Variabilidade Comum:** é uma característica que pode ser comum a todos os produtos da LPS, pode ser implementada como parte do núcleo de artefatos.

**Variabilidade Opcional:** é uma característica que pode ser comum a alguns, mas não a todos os produtos da LPS, deve estar explicitada como variabilidade e modelada de uma forma que esteja disponível apenas para produtos identificados.

**Variabilidade Específica:** é uma característica que pode ser somente de um único produto, mas que, no entanto, a plataforma deve conter.

Uma variabilidade é formada pelos seguintes elementos (Linden *et al.*, 2007):

**Ponto de Variação:** é o ponto onde existem diferenças no produto final, ou seja, é o ponto onde a decisão é tomada tardiamente, gerando assim produtos distintos.

**Variante:** são as possibilidades existentes que satisfazem um ponto de variação, definem as diferenças entre os produtos da LPS, ou seja, são as opções das características.

**Restrições entre variantes:** descrevem o relacionamento de duas ou mais variantes para resolver um ponto de variação. Existem dois tipos de relacionamentos: a variante requerente, essa quando é selecionada exige a seleção de outra variante e, a variante excludente, ou seja, a seleção dela proíbe a seleção de outras variantes.

As variantes podem ser dos seguintes tipos (Linden *et al.*, 2007):

- Variante Opcional: variante que pode ou não fazer parte do produto.
- Variante Obrigatória: variante que está em todos os produtos da LPS.
- Variante Alternativa: é uma variante específica a uma variabilidade ou ponto de variação e pode possuir propriedades de cardinalidade.

Um grande número de variabilidades pode trazer grandes problemas caso não haja controle e organização. O gerenciamento de variabilidades é uma tarefa crítica da LPS, pois nelas estão contidos todos os pontos de variação dos produtos da LPS, determinando seu sucesso ou não (Linden *et al.*, 2007).

O desenvolvimento de uma LPS é formado por duas atividades principais: a engenharia de domínio e a engenharia de aplicação (SEI, 2016). Na engenharia de domínio são desenvolvidos o projeto da LPS e o núcleo de artefatos (Schmid e Verlage, 2002), inclusive, o projeto da arquitetura da LPS (Product Line Architecture - PLA). É vantajoso interpor a PLA entre o modelo de características e os produtos gerados pela linha, pois a PLA apresenta as preocupações de implementação dos produtos de acordo ao modelo de características (Harman *et al.*, 2014). Como a PLA é um dos artefatos mais importantes da LPS (Oliveira Jr *et al.*, 2013b), na próxima seção serão discutidos os principais conceitos que norteiam a PLA.

### 2.1.1 Arquitetura de Linha de Produto

A arquitetura de um software monolítico bem como a de uma LPS é caracterizada por reunir as decisões de projeto, a organização dos componentes, as características de implementação, além de permitir a extensibilidade do projeto (Schmid e Verlage, 2002). Os projetos de PLAs abordam mais propriedades do que as arquiteturas de projetos de software monolíticos, ou seja, os projetos de PLAs se tornaram uma evolução das arquiteturas desses tipos de softwares. A PLA dispõe de todas as possíveis soluções de projeto dos produtos que a LPS pode gerar o que permite um elevado grau de reusabilidade dos componentes (Schmid e Verlage, 2002).

Uma PLA representa uma arquitetura que abstrai todas as características da LPS, isto é, nela estão todas as possíveis variações arquiteturais dos projetos de cada um dos produtos da LPS. Na PLA estão presentes todas as variabilidades da LPS, todos os pontos de variações e as modelagens das variantes. (Schmid e Verlage, 2002).

Pode-se entender que a PLA é um dos artefatos mais importantes para o sucesso de uma LPS, e o estudo analítico sobre o seu projeto deve ser considerado durante o desenvolvimento de LPSs (OliveiraJr *et al.*, 2013b). A importância da análise da PLA está no fato de que é possível prever a sua qualidade antes mesmo da geração dos produtos da linha, bem como, possíveis riscos ao projeto (Colanzi, 2014).

Os critérios de qualidade podem ser: os observáveis na execução do software como desempenho e, os critérios não observáveis durante a execução do software como reusabilidade. No critério de qualidade deve-se estar presente e, de forma clara, as características de variabilidade e flexibilidade que garantem a geração dos produtos da LPS (Etxeberria e Sagardui, 2005).

OliveiraJr *et al.* (2007) divide a avaliação de LPS em três grupos: (i) Avaliação de atributos de qualidade, (ii) avaliação estrutural e (iii) definição e avaliação de escopo de LPS. Segundo Colanzi (2014) a avaliação estrutural de uma PLA é realizada por meio de métricas que avaliam os componentes arquiteturais, bem como, os critérios de qualidade. Essa análise arquitetural pode envolver muito mais do que somente métricas, ela pode considerar a complexidade dos produtos, as restrições impostas à eles e fatores econômicos da LPS. Dessa forma, entende-se que o problema de avaliação de PLA não é um problema simples, ele deve considerar múltiplos objetivos além encontrar o melhor equilíbrio entre eles. Tais considerações motivaram a aplicação de algoritmos multiobjetivo para otimização de projetos de PLAs com base no uso de métricas para avaliar componentes e critérios de qualidade de LPSs.

## 2.2 Search Based Software Engineering

Nesta Seção é apresentado o campo de estudo denominado Engenharia de Software Baseada em Busca (*Search Based Software Engineering* - SBSE), além de apresentar as meta-heurísticas a serem utilizadas no presente trabalho.

Segundo Harman e Mansouri (2010) o termo SBSE denota uma otimização baseada em busca para resolver problemas da Engenharia de Software. Essa abordagem permite encontrar automaticamente soluções próximas do ideal. Pode-se listar algumas áreas onde se pode aplicar a SBSE, a saber: teste de software, projeto de software, modelagem, engenharia de requisitos, re-engenharia e projeto estrutural.

Um aspecto que se deve levar em consideração ao se trabalhar com SBSE é a modelagem dos problemas. Um problema deve ser modelado de forma a ser um problema de busca, ou seja, com um espaço de soluções (espaço de busca) e uma função objetivo. É no espaço de busca que estão todas as soluções possíveis. A função objetivo avalia a qualidade das soluções encontradas (Colanzi, 2014). Existem outros princípios para a utilização de SBSE de forma completa e eficaz, no entanto Harman *et al.* (2012) afirmam que há dois princípios essenciais e que, a partir deles, é possível iniciar os trabalhos de otimização. O primeiro princípio é a escolha da representação do problema, a sua forma deve permitir que um algoritmo de busca trabalhe efetivamente na otimização de soluções. O segundo princípio, é uma clara definição da função objetivo para que as soluções encontradas satisfaçam corretamente a otimização.

Quando um problema é afetado por diferentes fatores, mais de uma função objetivo é utilizada. Ao considerar mais de uma função objetivo é necessário encontrar o melhor *trade-off*, isto é, encontrar uma solução que atenda todas as funções objetivos da melhor forma possível (Harman *et al.*, 2012).

Devido à natureza estocástica dos algoritmos utilizados na SBSE, é preciso se basear em dados de testes estatísticos para inferir conclusões. Por intermédio de experimentos e testes de hipóteses pode-se responder as questões científicas desenvolvidas em SBSE (Harman *et al.*, 2012).

Alguns algoritmos podem ser utilizados em SBSE, como: *Simulated Annealing*, Algoritmos Genéticos e *Hill Climbing*, *Particle Swarm Optimization* (PSO) e *Ant Colony Optimization* (ACO). Vários deles possuem versões para tratar problemas multiobjetivos. Esses algoritmos são estudados pois alguns problemas da engenharia de software almejam encontrar soluções de forma automática satisfazendo mais de uma função objetivo (Harman e Mansouri, 2010). A próxima seção define os principais conceitos de otimização multiobjetivo.

### 2.2.1 Otimização Multiobjetivo

O problema de otimização multiobjetivo pode ser entendido como um problema com duas ou mais funções objetivo, isto é, ocorre uma otimização simultânea de interesses dependentes ou independentes, cooperativos ou concorrentes (Coello *et al.*, 2007). A existência de mais de uma função objetivo faz com que o problema não tenha uma única solução ótima, mas sim, um conjunto de soluções favoráveis à resposta do problema (Coello *et al.*, 2007).

Os problemas multiobjetivos da Engenharia de Software, normalmente, são afetados por fatores concorrentes ou conflitantes: duas características podem ser inversamente proporcionais, dessa forma, ao otimizar uma a outra é diretamente prejudicada. O que a otimização multiobjetivo procura é encontrar um balanceamento de interesses, um melhor *trade-off*, que satisfaça de forma considerável mais de um objetivo otimizando a solução de forma global (Harman *et al.*, 2012).

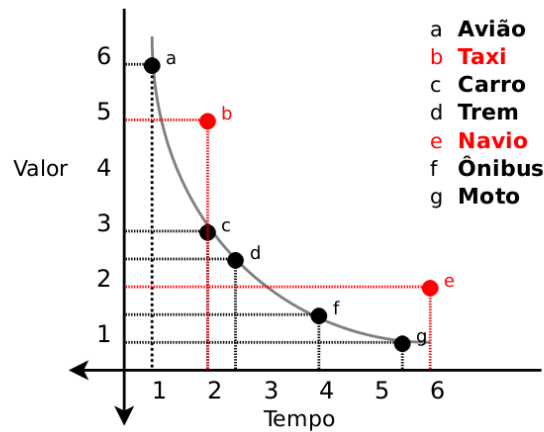
Trabalhos recentes abordando otimização multiobjetivo têm utilizado a Fronteira de Pareto (Pareto *et al.*, 1927) para obtenção das soluções. A Fronteira de Pareto possibilita encontrar um conjunto de soluções de melhor *trade-off* mesclando mais de uma função objetivo. Na Fronteira de Pareto existem os conceitos de soluções dominadas e soluções não dominadas. As soluções dominadas são as soluções que são superadas, em termos de fitness, por melhores soluções. As soluções não-dominadas são as que estão na fronteira e não são dominadas por nenhuma outra solução, apresentando os melhores valores de fitness para cada objetivo, representando as melhores soluções encontradas até o momento (Coello *et al.*, 2007).

Para esclarecer a ideia exposta acima, a Figura 2.1 apresenta um problema multiobjetivo. Pretende-se escolher um meio de transporte considerando a minimização de dois objetivos: o tempo e o custo. As soluções dominadas são as soluções (*b* e *e*), e as soluções não dominadas formam o conjunto de possíveis soluções (*a*, *c*, *d*, *f* e *g*). As soluções dominadas, estão nessa condição, pois existem outras soluções que obtiveram valores de otimização melhores considerando os objetivos.

Ao aplicar algoritmos de otimização multiobjetivo é possível encontrar um conjunto de soluções considerando os objetivos independentemente. Considerando essa abordagem, nas próximas subseções serão discutidos os algoritmos de busca que podem ser utilizados com múltiplos objetivos e que são empregados no presente trabalho de dissertação: o Algoritmo Genético e o Algoritmo Memético.



**Figura 2.1:** Representação da Fronteira de Pareto (Colanzi, 2014).



## 2.2.2 Algoritmos Genéticos

Algoritmos Genéticos (AG) (Goldberg *et al.*, 1989) fazem parte de uma categoria de algoritmos evolutivos que se caracterizam por utilizar mecanismos de evolução semelhantes à evolução biológica natural das espécies, inspirada no conceito de seleção natural de Darwin. A evolução por si só é um processo de otimização, uma vez que, sempre o melhor indivíduo sobreviverá durante a evolução restando somente os mais aptos. Diferente de outros algoritmos de otimização, os algoritmos evolutivos não utilizam de conhecimento prévio para solucionar o problema, mas sim, acumulam conhecimento durante as evoluções (Goldberg *et al.*, 1989).

Inicialmente os AGs foram propostos por John Holland em 1975 simulando sistemas genéticos computacionalmente. Esses algoritmos realizam uma otimização global utilizando a ideia de seleção natural explorando o histórico da evolução para analisar pontos favoráveis à uma geração de melhor desempenho (Goldberg *et al.*, 1989). Um algoritmo genético trabalha de forma iterativa explorando o conhecimento conquistado a cada iteração e, devido a sua estrutura genética, pode possuir várias versões se adaptando às necessidades do problema proposto.

O funcionamento do AG pode ser explicado de forma simplificada: inicialmente é gerada de forma aleatória uma população inicial e durante o processo de evolução cada indivíduo é avaliado por uma função de *fitness*. A função de *fitness*, também denominada função objetivo, recebe um valor que é sua aptidão, representando o quão bom esse indivíduo é no ambiente em que está. Os indivíduos mais aptos são mantidos e os outros são descartados. Os indivíduos selecionados podem sofrer alterações em suas características através de operadores de mutação ou cruzamento, formando assim, a

próxima população. Todas as etapas do processo: seleção, possível mutação e cruzamento e, por fim, a criação da nova geração, são repetidas por um número finito de vezes ou até encontrar uma solução ótima (Rezende, 2003).

Em suma o processo de AG pode ser visto como:

**Passo 01:** Geração da população inicial até um número de indivíduos  $x$ . A geração é dada com a utilização dos operadores de mutação escolhidos aleatoriamente. Uma cópia da PLA original faz parte dessa primeira população.

**Passo 02:** As funções de *fitness* selecionadas pelo arquiteto realizam a avaliação dos indivíduos da população.

**Passo 03:** Os melhores indivíduos são selecionados.

**Passo 04:** O operador de cruzamento é aplicado.

**Passo 05:** Os operadores de mutação são aplicados.

**Passo 06:** Os melhores indivíduos, restantes até este momento, substituem os piores formando próxima geração.

O processo do Passo 02 ao 06 se repete até que o número de gerações seja alcançado.

O AG pode sofrer diferentes adaptações e manter as características básicas da abordagem genética. NSGA-II (*Non-dominated Sorting Genetic Algorithm*) é uma adaptação de AG para tratar problemas multiobjetivos.

O algoritmo NSGA-II foi desenvolvido por Deb *et al.* (2002). Este algoritmo ordena sua população em várias fronteiras não-dominadas. Inicialmente todos os indivíduos estão na fronteira e, a cada geração os indivíduos pais e filhos são ordenados. Uma nova fronteira é formada com os indivíduos não-dominados e, isso acontece a cada geração até que todos os indivíduos estejam classificados em uma fronteira, dessa forma, vão existir várias fronteiras. É na última fronteira a ser formada que restarão os indivíduos não-dominados, estes, sendo os de melhor *trade-off*. Outra medida de ordenação é realizada, a medida de distância da multidão, com o objetivo de preservar a diversidade de soluções. Essa segunda medida calcula a distância de uma solução em relação aos seus vizinhos de mesma fronteira. A distância da multidão privilegia os indivíduos mais afastados no espaço de busca. O operador de seleção do NSGA-II utiliza a técnica torneio, onde usa as medidas de fronteira e distância da multidão como critérios.

Como relatado na Seção 2.2, a representação é de grande importância quando se utilizam algoritmos de busca para solucionar problemas. Assim, quando se utiliza AG é

importante representar o problema de acordo com as peculiaridades do método de busca, codificando cada um dos indivíduos, funções de *fitness* e os operadores utilizados. É comum em problemas de otimização denominar um indivíduo pelo termo solução.

A próxima subseção descreve os Algoritmos Meméticos que são o foco de estudo deste trabalho.

### 2.2.3 Algoritmos Meméticos

O algoritmo memético (AM) é uma extensão do AG onde é inserida uma busca local durante ou depois do processo de busca global (Fraser *et al.*, 2015). AMs também podem ser encontrados com outras denotações: algoritmos evolucionários (AE) híbridos, busca local genética, AE Baldwinian ou AE Lamarckian (Molina *et al.*, 2010). Um dos primeiros AM desenvolvidos foi em 1988 utilizando AG e *Simulated Annealing*, considerado um modelo tradicional de AM (Moscato e Cotta, 2003).

O AM é um algoritmo híbrido que utiliza uma busca global do AG juntamente com uma busca local. Esta estrutura permite aos indivíduos da população global uma possibilidade de otimização local, mas esse recurso pode ser usado ou não. Utilizar ou não a otimização local vai depender de restrições que mensuram a viabilidade da sua aplicação. A otimização local pode ser utilizada depois de um número  $x$  de iterações, ou a cada iteração, também pode ser aplicada logo após a reprodução, ainda, pode-se definir um número finito de vezes para sua utilização (Fraser *et al.*, 2015).

O AG por vezes pode perder características locais relevantes que a busca global desconsidera ou que devido aos saltos que acontecem durante suas iterações, algumas soluções próximas não são alcançadas. O AG pode demorar a alcançar uma solução boa devido a sua visão global, já com a utilização de uma busca local integrada, esse problema é solucionado, uma vez que ela explora a vizinhança de forma imediata. Este contexto de otimização permite encontrar boas soluções em um número de iterações menor. No entanto, o fato de integrar duas otimizações acarreta um maior tempo de processamento, mas esse fato é compensado com o crescimento do conjunto de soluções avaliadas (Fraser *et al.*, 2015).

O AM permite contornar os problemas encontrados tanto em algoritmos de busca local quanto nos de busca global. Os operadores da busca global favorecem a busca local à não ficar presa em ótimos locais e a otimização local beneficia a global ao alcançar vizinhos de forma imediata. Segundo estudos realizados por Harman e McMinn (2010), os AMs alcançaram um melhor desempenho do que a otimização global e local separadamente. No trabalho de Liaskos e Roper (2008), também se confirmam os resultados benéficos

do uso de AM que apresentam melhora na cobertura de soluções para casos de teste . O trabalho de Ishibuchi *et al.* (2003) utiliza o hibridismo do AM em uma otimização multiobjetivo e relata o efeito positivo trazido pelo AM melhorando a convergência das soluções utilizando a fronteira de Pareto.

Os vários tipos de algoritmos de busca local tem formas particulares de execução, no entanto, geralmente a busca local faz uma varredura nos vizinhos da solução candidata avaliando-os por meio de funções objetivo. Este tipo de busca pode ficar facilmente presa em ótimos locais e uma solução para esse problema é reiniciar a busca com novos valores. Já os algoritmos de busca global se comparados aos de busca local são mais eficientes mas menos eficazes, devido a busca por ótimos globais (Fraser *et al.*, 2015).

Existem diferentes abordagens para selecionar uma solução na busca local: First Improvement, Best Improvement e First-Best Improvement. First Improvement (Russell e Norvig, 2003) consiste em aplicar o operador de busca local uma vez e esta solução vai para a próxima geração, independentemente de ter uma qualidade melhor ou pior do que a solução atual. Best Improvement (Ochoa *et al.*, 2010) vai para a vizinhança da solução atual e seleciona o melhor vizinho. First-Best Improvement é a união das duas abordagens de pesquisa anteriores, na qual, é realizada uma busca na vizinhança que se encerra quando a primeira melhor solução é encontrada.

No Capítulo 3 são apresentados mais trabalhos que envolvem o AM e na próxima seção apresenta-se a abordagem MOA4PLA, a qual utiliza algoritmos multiobjetivos para otimização de projetos de PLA.

## 2.3 MOA4PLA

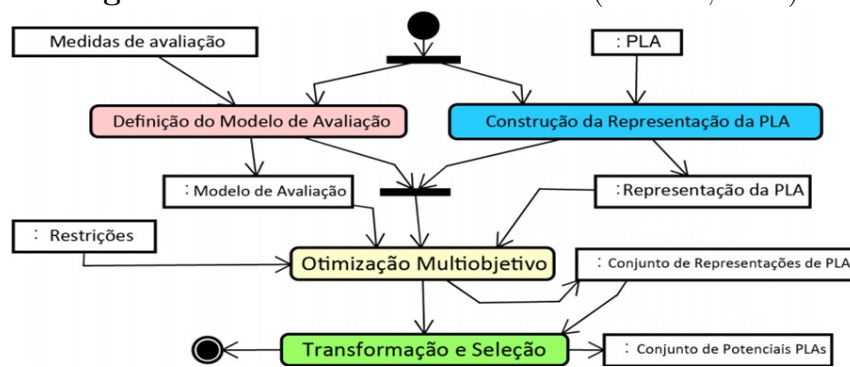
A MOA4PLA (*Multi-Objective Approach for Product-Line Architecture Design*) é uma abordagem desenvolvida por Colanzi (2014) que utiliza algoritmos de otimização multiobjetivos com o intuito de avaliar e aprimorar o projeto de PLA no que se refere a princípios básicos de projeto, modularização de características e extensibilidade de LPS. A aplicação dessa abordagem resulta um conjunto de soluções com melhores *trade-off* entre os objetivos otimizados. Assim, é permitido ao arquiteto de software escolher qual solução é a melhor para empregar em um determinado domínio de LPS. A abordagem MOA4PLA viabiliza a manipulação de PLAs por meio de um metamodelo de representação e utiliza funções objetivo para uma avaliação quantitativa de projetos de PLA.

A abordagem MOA4PLA permite utilizar diferentes algoritmos de busca multiobjetivos na otimização de projetos de PLA, independentemente do algoritmo utilizado, uma vez que, um problema pode ser resolvido de forma eficiente conduzido por diversos algoritmos

de busca. O objetivo é realizar uma análise estrutural do projeto de PLA por meio de métricas e de forma automatizada, tal análise se justifica pela dificuldade encontrada ao otimizar o projeto em vários aspectos simultaneamente, necessitando de um equilíbrio de interesses (Colanzi, 2014).

O processo da MOA4PLA pode ser visto na Figura 2.2. Este processo é dividido em quatro atividades principais: (i) construção da representação da PLA, (ii) definição do modelo de avaliação, (iii) otimização multiobjetivo e (iv) transformação e seleção. Segundo Colanzi (2014) as atividades apresentadas funcionam da seguinte forma:

**Figura 2.2:** Processo da MOA4PLA (Colanzi, 2014).

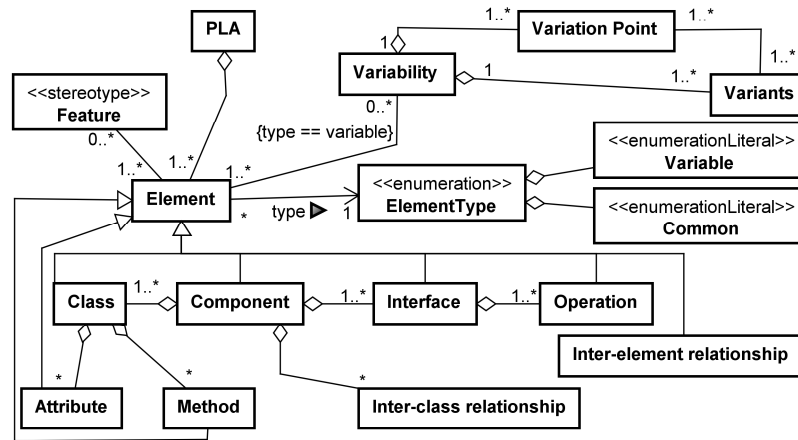


(i) **Construção da Representação da PLA:** O projeto de PLA é modelado em um diagrama de classes contendo elementos básicos: classes, atributos, métodos, interfaces, componentes, operações e seus relacionamentos. Este diagrama deve conter a descrição das variabilidades da LPS apresentando os pontos de variação e variantes e, por fim, apresentar os estereótipos para identificação de características associadas aos elementos arquiteturais. Como saída desta atividade, o diagrama de classes é convertido em um metamodelo interpretado pela ferramenta. O metamodelo está apresentado na Figura 2.3.

De maneira geral, um arquivo no formato XMI contém o projeto da PLA PLA original que é dado como entrada para a abordagem MOA4PLA. Durante o processamento o XMI é interpretado para gerar a representação da PLA no metamodelo. Cada elemento do XMI é identificado e instanciado como um objeto do metamodelo de acordo com seu tipo. Dessa forma, a representação da PLA conterá vários objetos para representar os elementos arquiteturais, seus relacionamentos, variabilidades e recursos associados a cada elemento arquitetural.

A classe *PLA* representa o projeto de PLA, esta classe é composta por objetos do tipo *Element* e todos os elementos arquiteturais identificados no XMI herdam

**Figura 2.3:** Modelo de representação do metamodelo (Colanzi, 2014).



da classe *Element*. Ao passo que o código XMI é interpretado, cada elemento é instanciado de acordo à seu tipo (*Class*, *Component*, *Inteface*, *Variability*, ...). Depois de instanciados todos os elementos, vai existir uma estrutura de objetos representando o projeto de PLA e, é essa estrutura que é utilizada pelos algoritmos de busca.

- (ii) **Definição do Modelo de Avaliação:** A saída dessa atividade é um modelo de avaliação formado pelos objetivos escolhidos pelo arquiteto. Um objetivo representa uma função de *fitness*. Essa função é composta por métricas: convencionais (coesão, acoplamento, tamanho e elegância do projeto), de extensibilidade de LPS e, por fim, métricas dirigidas a características. As função de *fitness* modeladas até o momento são descritas na Subseção 2.3.1.
- (iii) **Otimização Multiobjetivo:** Esta atividade trabalha com três entradas: as duas primeiras são as saídas das atividades (i) e (ii) e, a terceira são as restrições que deverão ser utilizadas no processo de otimização. Nesta atividade são utilizados os algoritmos de busca multiobjetivos. Os operadores de busca utilizados pelos algoritmos são apresentados na Subseção 2.3.3. A saída gerada é um conjunto de soluções alternativas para a PLA original. As soluções são as PLAs que se enquadraram às restrições estabelecidas e obtiveram melhores resultados nas funções de *fitness*, ou seja, os melhores *trade-off* entre os objetivos.
- (iv) **Transformação e Seleção:** Como entrada para essa atividade tem-se a saída da atividade anterior: o conjunto de soluções encontradas. As soluções são convertidas para um modelo legível para o arquiteto de software, ou seja, para um diagrama

de classes com todas as características necessárias para representação gráfica. O arquiteto realiza a escolha de uma solução oferecida, a que julgar mais adequada ao problema.

### 2.3.1 Modelo de Avaliação

O modelo de avaliação original da MOA4PLA incluía 4 funções objetivos: CM, FM, Ext e Eleg, todas a serem minimizadas (Colanzi, 2014). Devido a necessidade de um modelo de avaliação mais preciso, no trabalho de Santos *et al.* (2015) foram desenvolvidas seis novas funções (ACOMP, ACLASS, TAM, COE, DC e EC) e no trabalho de Delgado *et al.* (2017) foram desenvolvidas outras sete funções (LCC, CS, SD, CV, RCC, SV e TV).

Cada uma das funções objetivo são formadas por métricas de avaliação já consolidadas. As métricas convencionais utilizadas mensuram princípios básicos de projeto como coesão e acoplamento, existem métricas específicas a LPS como métricas dirigidas a características e de extensibilidade de PLA (Colanzi, 2014).

Na sequência são apresentadas as funções objetivo que compõe o modelo de avaliação da MOA4PLA:

**CM(*pla*):** seu objetivo é alcançar soluções com alta coesão, baixo acoplamento e alta reusabilidade. A função é constituída pela soma de várias métricas convencionais (Colanzi, 2014).

**FM(*pla*):** seu objetivo é alcançar um projeto com alta modularização de características, é formada pela soma de métricas dirigidas a características (Colanzi, 2014).

**Ext(*pla*):** seu objetivo é medir a extensibilidade de uma PLA (Colanzi, 2014).

**Eleg(*pla*):** seu objetivo é mensurar a elegância do projeto de PLA, formada pela soma das métricas de elegância (Colanzi, 2014).

**ACOMP(*pla*):** seu objetivo é medir o acoplamento entre componentes de uma PLA (Santos *et al.*, 2015).

**ACLASS(*pla*):** seu objetivo é medir o número de elementos arquiteturais que tem dependência de outras classes do projeto mais o número de elementos dos quais cada classe depende (Santos *et al.*, 2015).

**TAM(*pla*):** seu objetivo é medir o tamanho das operações de uma classe (Santos *et al.*, 2015).

**COE(*pla*):** seu objetivo é medir a coesão do projeto de PLA em termos de relacionamento interno entre classes de componentes mais o número de características associadas a cada componente (Santos *et al.*, 2015).

**DC(*pla*):** seu objetivo é medir a difusão de características, isso acontece somando-se o número de elementos arquiteturais associados a cada característica (Santos *et al.*, 2015).

**EC(*pla*):** seu objetivo é medir a interação entre características, mensurando as dependências existentes entre as características (Santos *et al.*, 2015).

**LCC(*pla*):** seu objetivo é medir a falta de coesão baseada em recursos, somando o número de características avaliadas por cada componente do PLA (Delgado *et al.*, 2017).

**CS(*pla*):** seu objetivo é medir o tamanho do componente em termos de operações (métodos) que são exigidas por outros serviços ou componentes (Delgado *et al.*, 2017).

**SD(*pla*):** seu objetivo é medir a similaridade de um projeto de PLA (Delgado *et al.*, 2017).

**CV(*pla*):** seu objetivo é medir quanto forte é a combinação de variabilidade considerando as dependências entre os pontos de variabilidade do projeto de PLA (Delgado *et al.*, 2017).

**RCC(*pla*):** seu objetivo é medir o acoplamento dos componentes do projeto de PLA, somando o número de relações entre as interfaces (Delgado *et al.*, 2017).

**SV(*pla*):** seu objetivo é medir a variabilidade estrutural do projeto de PLA (Delgado *et al.*, 2017).

**TV(*pla*):** seu objetivo é medir a variabilidade total do projeto de PLA (Delgado *et al.*, 2017).

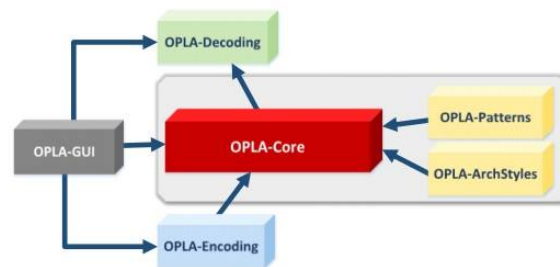
A próxima subseção apresenta a ferramenta que implementa a abordagem da MOA4PLA, automatizando todas as suas atividades.



### 2.3.2 OPLA-Tool

A ferramenta OPLA-Tool (*Optimization for PLA Tool*), automatiza a abordagem MOA4PLA (Colanzi, 2014). A Figura 2.4 representa a arquitetura da OPLA-Tool. A ferramenta, atualmente, está dividida em seis módulos: OPLA-GUI, OPLA-Encoding, OPLA-Decoding (Féderle *et al.*, 2015), OPLA-Core (Colanzi *et al.*, 2014), OPLA-Patterns (Guizzo *et al.*, 2014) e OPLA-ArchStyles (Mariani *et al.*, 2015).

**Figura 2.4:** Representação da Arquitetura da OPLA-Tool (Colanzi, 2014).



Em seguida apresenta-se a descrição de cada um dos módulos (Colanzi, 2014) :

**OPLA-GUI:** módulo de interface gráfica para interação direta ao usuário, neste módulo são informados a PLA que será otimizada, qual algoritmo multiobjetivo será utilizado, bem como, as funções de *fitness* e operadores de busca (Seção 2.3.3) disponíveis, tudo a critério do arquiteto. Esse módulo se comunica com três módulos (OPLA-Encoding, OPLA-Core e OPLA-Decoding), utilizando serviços disponibilizados em cada um.

**OPLA-Encoding:** Este módulo reconhece PLAs modeladas na ferramenta *Papyrus*<sup>1</sup>. O módulo converte o projeto de PLA em um metamodelo que pode ser utilizado pelo OPLA-Core.

**OPLA-Core:** é o principal módulo da ferramenta, ele tem como entrada o metamodelo gerado pelo módulo OPLA-Encoding e na sequência executa o processo evolutivo e de busca por meio de algoritmos de busca multiobjetivos. As funções de *fitness* e operadores utilizados já foram escolhidos no módulo OPLA-GUI. Como saída da otimização retorna um conjunto de soluções, ainda representado segundo o metamodelo.

**OPLA-Decoding:** Este módulo recebe como entrada a saída do módulo OPLA-Core. Ele decodifica as soluções e as converte em uma representação gráfica legível ao

<sup>1</sup>Mais informações sobre a ferramenta *Papyrus* - <https://eclipse.org/papyrus/>

arquiteto, modeladas em diagramas de classes no formato do *Papyrus*. Essas soluções, ainda, podem ser utilizadas como entrada para um novo processo de busca na OPLA-Tool.

**OPLA-Patterns:** Este módulo dispõe de um operador de aplicação de padrões de projeto denominado *Design Pattern Mutation Operator*. Este operador é responsável por analisar se uma determinada região da solução apresenta um escopo propício para aplicação de um padrão de projeto e, em caso positivo, realiza a aplicação.

**OPLA-ArchStyles:** módulo com operadores de mutação para preservar os estilos arquiteturais aplicados ao projeto de PLA.

Os dois módulos OPLA-Patterns e OPLA-ArchStyles são acoplados ao OPLA-Core podendo trabalhar em conjunto com seus operadores ou substituí-los, de acordo as configurações definidas pelo arquiteto.

Os algoritmos de busca multiobjetivos implementados até o momento na OPLA-Tool são dois: NSGA-II (Seção 2.2.2) e PAES Knowles e Corne (2000). Já, os operadores de busca implementados são apresentados a seguir.

### 2.3.3 Operadores de Busca da MOA4PLA

Esta seção apresenta os operadores de busca definidos pela abordagem MOA4PLA. Os operadores fundamentais (Colanzi, 2014) estão implementados no módulos OPLA-Core e o operador para aplicação de padrões de projeto (Guizzo, 2014) está implementado no módulo OPLA-Patterns.

#### Operadores Fundamentais

Colanzi (2014) propôs seis operadores fundamentais para a MOA4PLA os quais são abordados a seguir:

*Move Method:* Move um método escolhido aleatoriamente de uma classe para outra, tais classes também são escolhidas aleatoriamente. Uma associação bidirecional é adicionada entre as classes selecionadas, caso não exista. O operador deve respeitar as seguintes restrições para a classe de origem: não deve fazer parte de uma hierarquia de herança, não pode fazer parte de uma variante ou ponto de variação, deve ter mais de um método e, não pode ser a classe de destino.

*Move Attribute:* Move um atributo escolhido aleatoriamente de uma classe para outra, essas classes também são escolhidas aleatoriamente. Caso não exista, uma associação

bidirecional é adicionada. O operador deve respeitar as seguintes restrições para a classe de origem: não deve fazer parte de uma hierarquia de herança, deve ter mais de um atributo, não pode fazer parte de uma variante ou ponto de variação e, não pode ser a classe de destino.

*Add Class:* Uma nova classe é adicionada. O operador escolhe aleatoriamente uma classe e move um atributo ou método para uma nova classe. Se as classes estiverem no mesmo componente então é adicionada uma associação bidirecional, caso a classe adicionada faça parte de outro componente então é adicionada uma classe de interface no componente de origem. O operador deve respeitar as restrições de que a classe de origem não deve fazer parte de uma hierarquia de herança, deve ter mais de um atributo e método e, não pode fazer parte de um ponto de variação ou variante e, os componentes das interfaces devem ser da mesma camada arquitetural.

*Move Operation:* Move uma operação selecionada aleatoriamente entre duas interfaces também aleatórias. Se os componentes forem diferentes então uma interface destino é adicionada no componente original. Algumas restrições devem ser consideradas: o operador só move uma operação entre componentes da mesma camada arquitetural, a interface de destino não pode ser a mesma da origem e, as interfaces devem ter mais de uma operação.

*Add Manager Package:* Um novo pacote é criado, com isso, uma nova interface é criada para esse pacote. A nova interface recebe uma operação aleatória de uma interface de outro pacote também aleatório. Os componentes devem pertencer a mesma camada arquitetural.

*Feature-Driven-Operator:* O operador dirigido a características tem como objetivo modularizar uma característica entrelaçada em outras em um mesmo componente. Um componente é selecionado aleatoriamente e caso ele tiver interfaces relacionadas a mais de uma característica então, é selecionada uma dessas características para ser modularizada em um novo componente. Ao modularizar uma característica, um novo componente é criado e, todos os elementos arquiteturais associados a ela são movidos ao novo componente.

## **Operador de Aplicação de Padrões de Projeto**

Esta subseção faz referência ao operador desenvolvido no trabalho de Guizzo (2014) para aplicação de padrões de projeto ao projeto de PLA. O autor propôs um operador de mutação para permitir a aplicação automática de padrões de projeto em SBSE, chamado Design Pattern Mutation Operator. O autor identificou que se um projeto de PLA

possuir características entrelaçadas, os operadores fundamentais são mais eficientes, já se as características estiverem modularizadas, então a utilização do operador de aplicação de padrões de projeto tem melhores resultados.

O *Design Pattern Mutation Operator* aplica padrões de projeto em uma PLA mas, para que este operador seja utilizado, o projeto de PLA deve ser favorável a sua aplicação. O escopo avaliado deve satisfazer as particularidades necessárias para que o padrão de projeto seja aplicado e o padrão não deve inserir anomalias na arquitetura. Nesse operador a aplicação de um padrão de projeto é totalmente automatizada e sem interferências do arquiteto, a não ser, na escolha dos objetivos a serem otimizados. Como saída o operador retorna uma solução adicionando, nela, os estereótipos específicos do padrão de projeto.

O operador seleciona aleatoriamente um conjunto de classes e interfaces e analisa o escopo da PLA. Tal aspecto unido à seleção aleatória dos padrões de projeto, mantém a aleatoriedade do processo evolutivo. Caso o escopo permita o emprego de um padrão de projeto então o processo de aplicação é realizado. Em seguida é realizada uma análise para verificar se não há outro padrão de projeto empregado, caso exista, o atual padrão é corrigido, caso contrário uma nova instância é criada.

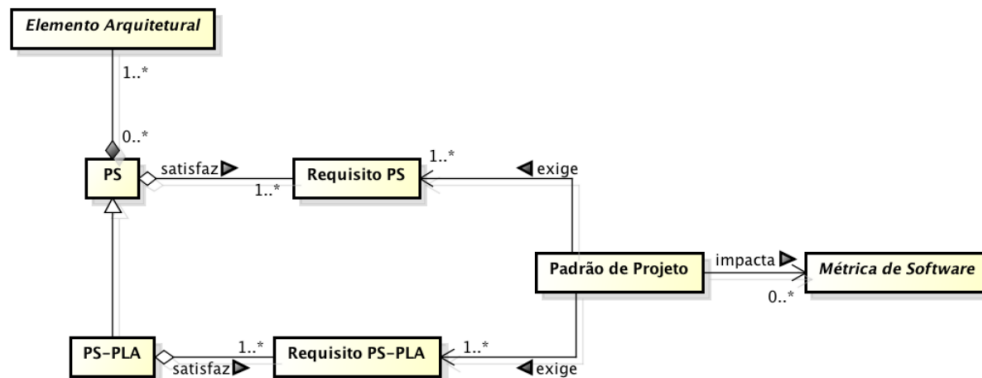
Existem dois escopos para a aplicação de um padrão de projeto. Caso o escopo satisfaça os requisitos mínimos para aplicação de um padrão de projeto então ela contém um Escopo de Aplicação do Padrão (*Pattern Application Scope* - (PS)), este escopo é composto somente de elementos arquiteturais convencionais, ou seja, sem elementos específicos de projeto de LPS. Em um segundo caso, pode existir um Escopo de Aplicação do Padrão em Arquitetura de Linha de Produto de Software (*Pattern Application Scope in Product Line Architecture* - (PS-PLA)), além de satisfazer os requisitos do PS, ele necessita conter elementos arquiteturais específicos de PLA como variabilidades e pontos de variação. Dessa forma se um escopo é um PS-PLA então ele obrigatoriamente é um PS.

Um padrão de projeto pode ser aplicado em um projeto de PLA mesmo que o escopo não seja PS-PLA, neste formato, sua aplicação produz efeitos positivos sobre métricas convencionais mas não sobre métricas específicas de LPS. Para que a aplicação de um padrão de projeto surta efeito sobre as métricas específicas de LPS, o projeto de PLA deve conter um escopo PS-PLA. A explanação dada até o momento fica mais clara na Figura 2.5, dessa forma, caso um elemento arquitetural satisfaça os requisitos de escopo o padrão de projeto pode ser aplicado.

Na Figura 2.5 um Elemento Arquitetural pode conter um escopo PS, caso ele satisfaça os requisitos do PS, o padrão de projeto pode ser aplicado e, tal aplicação impactada em Métricas de Software. Em um segundo caso, o Elemento Arquitetural pode conter

um escopo PS-PLA, herdando um escopo PS, caso o Elemento satisfaça as restrições do PS-PLA então o padrão de projeto pode ser aplicado impactando em Métricas de Software.

**Figura 2.5:** Funcionamento de satisfação de PS / PS-PLA (Guizzo, 2014).



Os padrões de projeto do catálogo GoF se dividem em três propósitos: criacional, estrutural e comportamental. Para serem utilizados na abordagem MOA4PLA os padrões de projeto deveriam permitir sua aplicação de forma automática e sem interferência do arquiteto de software, por esse motivo, Guizzo (2014), em seu trabalho, selecionou três padrões de projeto, da tabela GoF, os quais podem ser aplicados de forma automatizada e sem supervisão de um arquiteto de software: *Strategy*, *Bridge* e *Mediator*. Esses padrões de projeto serão apresentados na sequência.

**Padrão de Projeto *Strategy*:** Este padrão permite que o algoritmo varie independentemente de seus usuários, para isso, ele encapsula famílias de algoritmos fazendo-os intercambiáveis (GAMMA *et al.*, 2000). O projeto de PLA deve satisfazer os escopos PS«*Strategy*» e PS-PLA«*Strategy*», assim, um elemento arquitetural deve ter uma relação de uso ou dependência de pelo menos duas classes e/ou interfaces que compõem uma família de algoritmos, o elemento deve ser um ponto de variação e as classes relacionadas, as variantes. Esse padrão de projeto beneficia tanto métricas convencionais quanto específicas de LPS, especialmente a métrica de extensibilidade e beneficia a métrica de difusão de características, no entanto, não influencia métricas relacionadas a interação de características de LPS e coesão baseada em características.

**Padrão de Projeto *Bridge*:** Este padrão cria uma abstração de sua implementação, permitindo que ambos possam variar independentemente um do outro (GAMMA *et al.*, 2000). O projeto de PLA deve satisfazer os escopos PS«*Bridge*» e PS-PLA«*Bridge*», assim, um elemento arquitetural deve ter uma relação de uso ou dependência de pelo menos duas classes e/ou interfaces que compõem uma família de algoritmos e devem

estar associados a pelo menos uma característica em comum. O elemento também deve ser um ponto de variação e as classes relacionadas, as variantes. O *Bridge* favorece o acoplamento e a coesão da arquitetura, e beneficia métricas convencionais e especialmente a de extensibilidade, beneficia as métricas de difusão de características mas não favorece as métricas relacionadas a interação entre características.

**Padrão de Projeto *Mediator*:** Este padrão permite uma acoplamento fraco evitando que os objetos se refiram explicitamente entre si, para isso ele cria um objeto que encapsula as iterações de um conjunto de objetos, dessa forma as interações podem variar independentemente. O projeto de PLA deve satisfazer o escopo PS«*Mediator*», assim, deve existir dois elementos arquiteturais associados a uma mesma característica possuindo pelo menos dois relacionamentos. Dentre as métricas favorecidas, a métrica de acoplamento é a mais beneficiada, entretanto, apesar de não ser seu objetivo, o *Mediator* favorece a métrica de extensibilidade.

O próximo capítulo trata dos trabalhos relacionados ao foco de pesquisa do presente trabalho e apresenta os resultados encontrados na realização do mapeamento sistemático.

---

## Trabalhos Relacionados

---

Foi realizado um mapeamento sistemático (MS) a fim de identificar publicações científicas que envolvam estudos relacionados a utilização de algoritmos meméticos em LPS e/ou projeto de software, permitindo com isso, a aquisição de conceitos necessários para a construção e aprimoramento do trabalho de dissertação, além de permitir uma visão geral do estado da arte do tema investigado.

As buscas do MS foram realizadas nas bibliotecas: ACM, IEEE, Scopus e Springer. O MS teve o objetivo de reunir trabalhos relacionados que respondam às seguintes questões:

- Quais são os problemas relacionados ao projeto de software solucionados com a utilização de algoritmos meméticos?
- Como os algoritmos meméticos estão sendo utilizados no contexto de linhas de produto de software?

O MS foi desenvolvido por meio de quatro etapas:

**Etapa 01:** Busca das palavras chaves e sinônimos nas seções: título, *abstract* e palavras-chaves por meio das ferramentas de filtragem oferecidas pelas bibliotecas digitais selecionadas.

**Etapa 02:** Leitura do título e *abstract* para identificar a relevância dos estudos encontrados no contexto da pergunta do mapeamento sistemático, utilizando os critérios de seleção listados no Apêndice A.1.

**Etapa 03:** Nos trabalhos selecionados na Etapa 02, foi realizada a leitura da introdução e conclusão para sanar possíveis dúvidas quanto a contribuição dos trabalhos para o mapeamento.

**Etapa 04:** Estudos categorizados como relevantes na Etapa 03 foram lidos na íntegra, buscando-se informações relacionadas à pesquisa. Um estudo é categorizado relevante uma vez que ele contenha pelo menos um critério de inclusão e nenhum critério de exclusão.

No Apêndice A é apresentado o protocolo do MS e, a seguir, os resultados obtidos.

### 3.1 Resultados do Mapeamento Sistemático

A Etapa 01 foi realizada encontrando um total de 242 trabalhos relacionados de alguma forma com a *string* de busca pesquisada. Com a execução da Etapa 02 restaram somente 44 trabalhos. Em função de existirem dúvidas quanto ao conteúdo apresentado em alguns trabalhos, foi necessária a realização da Etapa 03, a qual reduziu o número de trabalhos para cinco. Na Etapa 04 os cinco trabalhos, considerados mais pertinentes a responder as questões do MS, foram lidos na íntegra. Com o fim da Etapa 04, somente 2 trabalhos foram selecionados como relevantes para o MS. A Tabela 3.1 mostra de forma quantitativa o desenvolvimento de cada uma das etapas.

**Tabela 3.1:** Resultados do Mapeamento Sistemático.

<b>Bibliotecas Consultadas</b>	<b>Etapa 01</b>	<b>Etapa 02</b>	<b>Etapa 03</b>	<b>Etapa 04</b>
<b>ACM</b>	21	10	3	2
<b>IEEE</b>	78	6	1	0
<b>Scopus</b>	57	13	1	0
<b>Springer</b>	86	13	0	0
<b>Totais</b>	242	44	5	2

A seguir apresenta-se uma breve discussão sobre os cinco trabalhos restantes da Etapa 03, pois, apesar de somente dois trabalhos serem selecionados no final da Etapa 04, em cada um dos cinco trabalhos é possível ter uma visão diferenciada de como os algoritmos meméticos podem ser empregados. Os dois trabalhos selecionados na Etapa 04 apresentam como os algoritmos meméticos estão sendo utilizados em projeto de software, bem como, quais problemas eles resolvem. A Tabela 3.2 exprime uma visão resumida dos cinco trabalhos, a coluna (i) identifica os autores, a coluna (ii) apresenta o problema a ser



solucionado, a coluna (iii) informa qual algoritmo foi utilizado, a coluna (iv) explica como o algoritmo foi utilizado, a coluna (v) mostra se o trabalho foi selecionado ou não na Etapa 04 e, por último, a coluna (vi) informa se o AM solucionou o problema de forma positiva.

O trabalho de Smith e Simons (2013) investiga a utilização de buscas locais em um quadro memético, ou seja, trabalhando juntamente com buscas globais, com objetivo de melhorar as soluções candidatas tanto na qualidade quanto na eficiência da busca de base populacional. Esse trabalho compara a utilização de Algoritmos Evolucionários (*Evolutionary Algorithms* - AE) e Otimização por Colônia de Formigas (*Ant Colony Optimisation* - ACO), e procura responder duas questões principais:

- 1 - Existe diferença no desempenho da busca global quando aplicado à modelagem de classe? Se sim, quais tipos de problemas e por quê?
- 2 - A adição de uma busca local, para criação de um algoritmo memético, melhora o desempenho da busca global? E, se há diferenças no desempenho de diferentes algoritmos de busca global?

O trabalho de Smith e Simons (2013) apresenta as estruturas dos algoritmos: AE e ACO. No entanto o foco de atenção foi o AE por se aproximar do AG. O AE adota uma com probabilidade de mutação de 0.2 selecionando de forma uniforme e aleatória um conjunto de valores permitidos. O valor é então multiplicado por  $l$  e o operador é aplicado para saber o número de mutações que serão realizadas. um dos três operadores é selecionado e aplicado no número de vezes encontrado na multiplicação por  $l$ , onde,  $l$  foi definido por um conjunto de dez taxas de mutação 0,001, 0,002, 0,01, 0,02, 0,1, 0,2, 1, 2, MIN (0,25  $l$ , 5), MIN (0,25  $l$ , 10). O operador de busca local, descrito no trabalho move atributos ou métodos entre as classes e, os movimentos considerados são os que permitem soluções aceitáveis. Duas funções de *fitness* foram utilizadas: *Coupling Between Objects measure* responsável por medir o acoplamento entre classes e, *Numbers Among Classes* responsável por medir a elegância das classes.

**Tabela 3.2:** Resumo da Etapa 04 do Mapeamento Sistemático.

(i) Trabalho	(ii) Problema	(iii) Algoritmos Utilizados	(iv) Estratégia de Implementação	(v) Selecionado na Etapa 04?	(vi) O Algoritmo Memético foi benéfico?
(Smith e Simons, 2013)	Modelagem de classes	Algoritmo Evolucionário e ACO	Inserção de busca local dentro do algoritmo evolucionário e do ACO	Sim. Inclusão item 5	Sim. A busca local favoreceu soluções em grandes populações e com grande número de restrições
(Chawla <i>et al.</i> , 2015)	Geração de testes a algoritmos	Algoritmo Genético e PSO	Desenvolvimento de um algoritmo híbrido entre as duas abordagens, global e local	Não. Exclusão item 7	Sim. Minimizou o tempo de geração de testes e maximizou o ramo de cobertura e detecção de falhas
(Jeya Mala <i>et al.</i> , 2013)	Procura testar o impacto de componentes em um software	Algoritmo Genético	O algoritmo de busca local foi inserido nos operadores de cruzamento e mutação. A busca local tenta corrigir problemas do AG.	Sim. Inclusão item 5	Sim. Os resultados encontrados pelo algoritmo memético superam os do AG.
(Garzon-Rodriguez <i>et al.</i> , 2015)	Melhorar a ferramenta Goldberry-GA	Algoritmo Genético modificado, onde o processo de geração é acelerado.	Essa nova versão da ferramenta permite inserir metaheurísticas durante o processo de otimização para melhorar os resultados retornados.	Não. Exclusão item 7	Não informa.
(Bai <i>et al.</i> , 2010)	Otimizar arquitetura de aquedutos	Algoritmo Genético	Não informa	Não. Exclusão item 7	Sim. A otimização das arquiteturas de aquedutos reduziram em 20% os custos de engenharia.

Por fim, Smith e Simons (2013) apresentam conclusões favoráveis ao uso de algoritmos meméticos. As soluções apresentadas pelo AE com busca local foram melhores do que as do ACO com busca local. Apesar de tais resultados, o trabalho refutou a hipótese de que a inserção de uma busca local iria reduzir o impacto de escolha da heurística de busca global, em razão de a busca local não apresentar resultados favoráveis em todos os experimentos realizados. O trabalho apresentado acima foi incluído na Etapa 04 por satisfazer os critérios de seleção.

No trabalho de Chawla *et al.* (2015) é apresentado um hibridismo de AG e Otimização por Enxame de Partículas (PSO - *Particle Swarm Optimization*). Esse formato de algoritmo permite alguns benefícios como: melhor tempo de execução e complexidade de espaço, fácil implementação, melhores soluções são armazenadas, estabilidade e qualidade de solução, maior velocidade de convergência e maior probabilidade de se encontrar um ótimo global. Esta abordagem memética cumpriu com êxito os objetivos de minimizar o tempo de execução, maximizar a cobertura de ramos e aumentar a detecção de falhas. No entanto, este estudo é voltado para a geração de testes de algoritmos e, por essa razão, ele não satisfaz os critérios de seleção sendo eliminado na Etapa 04.

O trabalho de Jeya Mala *et al.* (2013) apresenta o algoritmo genético em conjunto com uma busca local com o intuito de analisar ótimos globais e locais numa abordagem que testa o impacto de componentes em uma arquitetura classificando em: catastrófico, crítico, marginal e menor. De acordo com o impacto, uma lista de componentes críticos é gerada, esses componentes são testados por casos de testes já estabelecidos. A busca se baseia na avaliação dos valores das mutações. O trabalho se divide em duas fases: na primeira os componentes críticos são identificados por meio de análise de impacto baseado em teste de mutação e, na segunda fase, é proposto um quadro de componentes críticos baseados em casos de teste de geração e otimização utilizando o algoritmo híbrido de AG e busca local (HGA). O HGA consegue superar o AG na otimização dos casos de teste.

Jeya Mala *et al.* (2013) apresentam alguns problemas encontrados no AG que podem ser resolvidos com o uso do hibridismo, ou seja, AM. Os problemas encontrados no AG são: (i) o AG convergir para um ótimo local antes de convergir para um ótimo global, (ii) o AG convergir precocemente para uma solução que pode ser inválida para dados posteriores e, (iii) o AG não conseguir convergir para solução ótima quando em algum momento, a função de *fitness* apresenta um valor muito alto ou muito baixo. Durante o trabalho os autores apresentam como o HGA foi projetado e como o hibridismo foi utilizado nos operadores de cruzamento e mutação. Neste trabalho o algoritmo memético é utilizado para a geração de casos de teste os quais analisam o impacto de componentes em arquiteturas de software, neste caso o algoritmo memético está resolvendo um problema

em arquitetura de software de forma indireta e, por esse motivo, ele foi aceito na Etapa 04.

O trabalho de Garzon-Rodriguez *et al.* (2015) apresenta o desenvolvimento de uma ferramenta chamada Goldberry-GA, uma nova abordagem de algoritmo genético onde é acelerado o processo de geração da população. O trabalho descreve a utilização de uma boa arquitetura e padrões de projeto, só que no entanto, é referente ao desenvolvimento da própria aplicação Goldberry-GA. Esse novo modelo de AG pode utilizar algumas metaheurísticas entre elas buscas locais. Pela razões descritas, o trabalho de Garzon-Rodriguez *et al.* (2015) não satisfaz os critérios de seleção e, assim, não foi selecionado na Etapa 04.

O quinto e último trabalho da Etapa 03, desenvolvido por Bai *et al.* (2010), utiliza um hibridismo de algoritmo genético com busca local para otimizar de projetos de arquitetura de aquedutos em larga escala. Com a utilização do algoritmo memético, o sistema de otimização de arquiteturas de aquedutos conseguiu alcançar soluções ótimas, o que reduziu os custos de engenharia em 20%. Este trabalho relata o sucesso do uso de algoritmos meméticos, no entanto por não ser utilizado para otimização de projeto de software ele não satisfaz os critérios de seleção e, por esse motivo, não foi selecionado na Etapa 04.

Com a análise dos trabalhos realizada até o momento, é possível perceber a utilidade e versatilidade da utilização de algoritmos meméticos para resolver problemas de otimização. Muito embora, no início, o MS tenha conseguido refinar um número considerável de trabalhos com possibilidade de estarem relacionados com seu propósito, somente dois trabalhos conseguiram responder de forma satisfatória os questionamentos do MS. Os trabalhos de Smith e Simons (2013) e Jeya Mala *et al.* (2013) respondem as questões principais do MS como segue:

- Quais são os problemas relacionados ao projeto de software solucionados com a utilização de algoritmos meméticos?

O trabalho de Smith e Simons (2013) utiliza o AM com objetivo de otimizar a modelagem de classes em relação às métricas de coesão e acoplamento. Ele utiliza o operador de mutação do AG para realizar as trocas de métodos e atributos entre as classes e mede o quão perto da solução ótima a nova configuração de modelagem está e, procura acelerar a busca de soluções com a utilização da busca local.

Já, no trabalho de Jeya Mala *et al.* (2013), o AM é utilizado para testar o impacto de componentes em uma arquitetura de software, se dividindo em duas fases: a primeira onde a mutação é baseada na análise de impacto e, a segunda, baseada na geração e otimização de componentes críticos.

- Como os algoritmos meméticos estão sendo utilizados no contexto de linhas de produto de software?

Os dois trabalhos selecionados na Etapa 04 não envolvem linha de produto de software.

Com o fim do MS não foi possível responder a questão: “Como os algoritmos meméticos estão sendo utilizados no contexto de linhas de produto de software?”. Com isso entende-se que considerando este MS não há, até o momento, estudos que relacionem a utilização de AM no contexto de LPS bem como à sua arquitetura. Este fato deixa evidente a importância de uma investigação sobre aplicação de AMs no projeto de PLA para que seja possível entender se a abordagem memética em PLA é favorável ou não, se comparada com os resultados já encontrados utilizando a abordagem genética, tendo em vista os resultados promissores do uso de AM para resolver outros tipos de problemas.

## 3.2 Considerações Finais

Este capítulo abordou o levantamento do estado da arte sobre a utilização de AMs para resolver problemas relacionados a projetos de LPS. Como não foram encontrados trabalhos associados a esse contexto, foi identificada uma lacuna de pesquisa. A lacuna se refere a utilização de AMs para otimização de projetos de PLA. Tal fato embasou a motivação inicial do estudo em se trabalhar com AMs no contexto da abordagem MOA4PLA.

Assim, o próximo capítulo discorre sobre o desenvolvimento da abordagem memética para tratar o problema de otimização de projeto de PLA no contexto da abordagem MOA4PLA.

---

## Abordagem Memética

---

Este capítulo tem como objetivo descrever a abordagem memética proposta no presente trabalho.

Como já destacado anteriormente, Colanzi (2014) propôs a abordagem chamada MOA4PLA que têm como objetivo a utilização de algoritmos de busca multiobjetivos para avaliar e aprimorar o projeto de PLA nos princípios básicos de projeto, modularização de características e extensibilidade de projeto de PLA.

Guizzo (2014) propôs um operador de mutação para permitir a aplicação automática de padrões de projeto em SBSE, chamado *Design Pattern Mutation Operator*. Uma análise de viabilidade mostrou que três padrões de projeto, do catálogo GoF (Ó Cinnéide e Nixon, 2001), podem ser aplicados automaticamente: *Strategy*, *Bridge* e *Mediator*. Os resultados encontrados foram satisfatórios, indicando que o uso do *Design Pattern Mutation Operator* permite melhores soluções se comparado com as soluções obtidas usando apenas os operadores fundamentais (Guizzo, 2014).

Além disso, os experimentos de Guizzo (2014) identificaram que se um projeto de PLA possuir características difusas, os operadores fundamentais são mais eficientes, já se as características estiverem modularizadas, então a utilização do *Design Pattern Mutation Operator* tem melhores resultados. A utilização em conjunto dos operadores fundamentais com o operador de aplicação de padrões de projeto selecionados aleatoriamente, alcançou soluções melhores, e isso, trouxe indícios de que a utilização conjunta dos operadores é mais viável.

Nos trabalhos de Guizzo (2014) e Colanzi (2014) foram utilizados algoritmos em uma abordagem genética. Essa abordagem apresenta uma otimização global que em grandes populações pode acarretar elevados custos de tempo e processamento.

Nos trabalhos apresentados na Seção 3.1 percebe-se que a utilização de uma abordagem memética pode trazer benefícios à otimização, como no trabalho de Smith e Simons (2013), Jeya Mala *et al.* (2013), Fraser *et al.* (2015) e Chawla *et al.* (2015), onde a utilização do AM melhorou a qualidade e velocidade da otimização.

Até o final desse trabalho, não haviam sido identificados relatos sobre o uso de uma abordagem memética na otimização de projeto de PLA. Esta lacuna de pesquisa identificada motivou a investigação da aplicação de AMs para otimização de projeto de PLA. Outro fator relevante está na realização de uma análise qualitativa das soluções encontradas, uma vez que, nos trabalhos de Colanzi (2014) e Guizzo (2014) foram realizadas, somente, análises quantitativas das soluções.

Com base no exposto, a hipótese deste trabalho é que: a aplicação de algoritmos meméticos no processo de otimização da MOA4PLA é capaz de obter projetos de PLA melhores do que os projetos obtidos na versão original da MOA4PLA. A abordagem memética bem como seus aspectos são apresentados a seguir.

## 4.1 Algoritmo Proposto

O algoritmo memético é uma extensão do AG onde é inserida uma busca local durante ou após o processo de busca global (Russell e Norvig, 2003). Esta estrutura permite aos indivíduos da população global a possibilidade de otimização local (Fraser *et al.*, 2015).

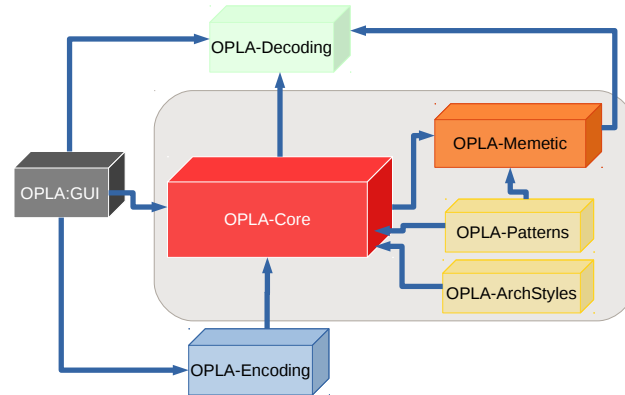
O AG com melhor desempenho usado na OPLA-Tool é o NSGA-II (Colanzi e Vergilio, 2014). Neste contexto, foi implementado um algoritmo memético modificando a estrutura necessária da OPLA-Tool para tratar o operador *Design Pattern Mutation Operator* (Seção 2.3.3) como operador de busca local no algoritmo NSGA-II (Seção 2.2.2).

Foram realizadas modificações na rotina do NSGA-II para que fosse possível inserir a busca local. Com isso, um novo módulo foi desenvolvido para a ferramenta OPLA-Tool denominado OPLA-Memetic. Este módulo contém o AM e interage com os módulos OPLA-Core e OPLA-Patterns. A Figura 4.1 apresenta a configuração da nova arquitetura da OPLA-Tool na qual o módulo OPLA-Memetic foi adicionado entre os módulos OPLA-Core e OPLA-Patterns.

O módulo OPLA-Memetic segue as mesmas atividades da MOA4PLA. No entanto, a principal mudança é que, necessariamente após a aplicação de um operador de mutação, o operador de busca local é executado. Caso exista um escopo adequado para aplicar um padrão de projeto então, o padrão é aplicado.

O pseudo-código apresentado no Algoritmo 1 mostra como o NSGA-II foi adaptado para incluir a busca local. Dois pais ( $p_1$  e  $p_2$ ) são selecionados (linhas 3 e 4). Utilizando os

**Figura 4.1:** Representação da arquitetura da nova versão da OPLA-Tool.



pais, o operador de crossover obtém dois filhos (linha 5). Os *filhos* são mutados (linhas 6 e 7) e, então, a busca local é aplicada sobre cada filho (linhas 8 e 9). Os *filhos* resultantes desse processo seguem para uma nova população (linhas 10 e 11).

---

**Algoritmo 1** Algoritmo Memético - Adaptação do NSGA-II

---

MEMETIC()

```

1  while (MaximoAvaliacoes) do
2      for s = 0 to Populacao.size/2 do
3          pais1 = POPULACAO(s)
4          pais2 = POPULACAO(s + 1)
5          filhos[] = CROSSOVER(p1, p2)
6          filhos[1] = MUTATION(filhos[1])
7          filhos[2] = MUTATION(filhos[2])
8          filhos[1] = LOCALSEARCH(filhos[1])
9          filhos[2] = LOCALSEARCH(filhos[2])
10         NovaPopulacao = filhos[1]
11         NovaPopulacao = filhos[2]
12         s += 2
13     for end
14     Populacao = SELECTION(NovaPopulacao)
15 while end

```

---

O operador de busca local seleciona de forma aleatória o conjunto de classes e interfaces que terão seu escopo de aplicação analisado, dessa forma, não há garantias que o algoritmo encontrará especificamente o conjunto favorável à aplicação de padrão de projeto. Logo, quando utiliza-se o termo busca local aplicada, entende-se que houve uma tentativa de



aplicação e caso exista o escopo para tal, então o padrão de projeto é aplicado, caso contrário a tentativa encerra e a rotina do algoritmo prossegue.

Quatro versões da busca local foram implementadas para OPLA-Memetic: NoChoice, Bestof2, Bestof12 e UntilBest. A primeira e a segunda versões implementadas são baseadas na abordagem First Improvement, com a diferença de que, na segunda versão, a melhor solução dentre a global e local é escolhida. A terceira versão usa a abordagem Best Improvement e a quarta versão usa First-Best Improvement. As subseções a seguir apresentam as versões do AM implementadas.

#### 4.1.1 Algoritmo Memético sem Critério de Escolha (NoChoice)

Esta versão de AM utiliza a abordagem First Improvement. O AM aplica a busca local logo após a aplicação do operador de mutação da busca global e, sem qualquer critério de comparação de *fitness*, a solução corrente prossegue para a próxima geração no NSGA-II.

O Algoritmo 2 representa a busca local NoChoice. A variável  $s$  é a solução encontrada pelo operador de busca local e, sem qualquer avaliação,  $s$  é retornada.

---

##### Algoritmo 2 LocalSearch - NoChoice

---

LOCALSEARCH(*solution*)

```
1  $s = \text{OPERATORLOCALSEARCH}(\text{Solution})$ 
2 return ( $s$ )
```

---

#### 4.1.2 Algoritmo Memético First Improvement (Bestof2)

A abordagem First Improvement também foi utilizada nesta versão de AM. O operador de mutação é aplicado e, na sequência, a busca local. No entanto, é realizada uma avaliação entre as soluções encontradas pela busca global e pela busca local, na qual, a melhor solução segue para a próxima geração.

O Algoritmo 3 mostra o funcionamento do Bestof2. A solução da busca global é salva na variável  $sOri$  (linha 1). A solução obtida após a aplicação da busca local é atribuída à variável  $sLoc$  (linha 2). As duas soluções  $sOri$  e  $sLoc$  são avaliadas e a melhor é retornada (linha 3).

O método de avaliação é apresentado no Algoritmo 4. O método *theBest* consiste em comparar os valores de *fitness* das soluções das buscas global e local, escolhendo a solução que tem o menor somatório (linhas 5 a 8), uma vez que, uma otimização por minimização

---

**Algoritmo 3** LocalSearch - Bestof2
 

---

```

LOCALSEARCH(solution)
1  sOri = solution
2  sLoc = OPERATORLOCALSEARCH(solution)
3  sSelect = THEBEST(sOri, sLoc)
4  return (sSelect)

```

---

é realizada. O somatório consiste na soma dos valores obtidos por cada uma das funções objetivo (linhas 2 e 3).

---

**Algoritmo 4** theBest
 

---

```

THEBEST(solution1, solution2)
1  for f = 0 to numeroDeObjetivos do
2      somatorio1 =+ solution1.getFitness(f)
3      somatorio2 =+ solution2.getFitness(f)
4  for end
5  if (somatorio2 <= somatorio1) then
6      return solution2
7  else
8      return solution1

```

---

### 4.1.3 Algoritmo Memético Best Improvement (Bestof12)

A terceira versão de AM utiliza a abordagem Best Improvement, criando uma vizinhança de soluções encontradas pela busca local. A busca local é aplicada logo após o operador de mutação da busca global mas, nessa abordagem, a busca local é repetida doze vezes sobre a solução da busca global.

O Bestof12 está representado no Algoritmo 5. A implementação atual do AM trabalha sobre um processador com quatro cores. Três buscas locais foram instanciadas para cada núcleo usando paginação realizada pelo sistema operacional, o que resulta em 12 vizinhos (linhas 1 a 16). A melhor solução da vizinhança é selecionada pelo método *TheBest12* (linha 18).

O método de avaliação *TheBest12* é baseado no método *TheBest*, e está apresentado no Algoritmo 6. Neste método, 12 soluções são comparadas por meio do somatório das funções de *fitness* utilizadas. A função da linha 16 ordena e retorna a solução com menor soma.

---

**Algoritmo 5** LocalSearch - Bestof12

---

LOCALSEARCH(*solution*)

```
1  Core0:
2  sLoc0 = OPERATORLOCALSEARCH(solution)
3  sLoc1 = OPERATORLOCALSEARCH(solution)
4  sLoc2 = OPERATORLOCALSEARCH(solution)
5  Core1:
6  sLoc3 = OPERATORLOCALSEARCH(solution)
7  sLoc4 = OPERATORLOCALSEARCH(solution)
8  sLoc5 = OPERATORLOCALSEARCH(solution)
9  Core2:
10 sLoc6 = OPERATORLOCALSEARCH(solution)
11 sLoc7 = OPERATORLOCALSEARCH(solution)
12 sLoc8 = OPERATORLOCALSEARCH(solution)
13 Core3:
14 sLoc9 = OPERATORLOCALSEARCH(solution)
15 sLoc10 = OPERATORLOCALSEARCH(solution)
16 sLoc11 = OPERATORLOCALSEARCH(solution)
17
18 sSelect = THEBEST12(sLoc0,sLoc1,sLoc2,sLoc3,sLoc4,sLoc5,sLoc6+
19 +,sLoc7,sLoc8,sLoc9,sLoc10,sLoc11)
20
21 return (sSelect)
```

---

---

**Algoritmo 6** theBest12
 

---

```

THEBEST(sol1, sol2, sol3, sol4, sol5, sol6, sol7, sol8, sol9, sol10, sol11)
1  for  f = 0 to numeroDeObjetivos  do
2      soma0 =+ sol0.getFitness(f)
3      soma1 =+ sol1.getFitness(f)
4      soma2 =+ sol2.getFitness(f)
5      soma3 =+ sol3.getFitness(f)
6      soma4 =+ sol4.getFitness(f)
7      soma5 =+ sol5.getFitness(f)
8      soma6 =+ sol6.getFitness(f)
9      soma7 =+ sol7.getFitness(f)
10     soma8 =+ sol8.getFitness(f)
11     soma9 =+ sol9.getFitness(f)
12     soma10 =+ sol10.getFitness(f)
13     soma11 =+ sol11.getFitness(f)
14
15  for end
16  sSelect = ORDENASOMATORIOSSELECONAMENORSOLUCAO(soma0,soma1+
17  +soma2,soma3,soma4,soma5,soma6,soma7,soma8,soma9,soma10,soma11)
18  return  sSelect

```

---

#### 4.1.4 Algoritmo Memético First Best Improvement (UntilBest)

A última versão do AM funciona com a abordagem First Best Improvement cujo pseudo-código é apresentado no Algoritmo 7. O algoritmo repete a busca local para mesma solução da busca global até alcançar uma melhoria na soma dos *fitness* das funções objetivo utilizadas (linhas 2 a 8), ou seja, até encontrar a primeira melhoria na vizinhança. Para fazer isso, a comparação entre a solução encontrada pela busca global (*sOri*) e o vizinho (*sLoc*) é realizada pelo método denominado *theBestUntil* (linha 4), apresentado no Algoritmo 8. Assim que uma solução melhor for encontrada, a solução *sLoc* é retornada.

O método de avaliação *theBestUntil*, Algoritmo 8, compara os somatórios de *fitness* das funções objetivo utilizadas e retorna o melhor somatório (linhas 5 a 8).

Devido ao *Design Pattern Mutation Operator* (busca local) precisar de um escopo adequado para aplicar um padrão de projeto, existem casos em que, apesar de muitas tentativas de melhora realizada pela busca local, nenhuma melhoria ocorre. Para evitar esse problema, o número de tentativas para alcançar uma solução melhor foi limitado a

---

**Algoritmo 7** LocalSearch - UntilBest

---

LOCALSEARCH(*solution*)

```
1  sOri = solution
2  while (best == false) do
3      sLoc = OPERATORLOCALSEARCH(solution)
4      best = THEBESTUNTIL(sOri, sLoc)
5      contador ++
6      if (contador > 15) then
7          return sOri
8  while end
9  return sLoc
```

---

---

**Algoritmo 8** theBestUntil

---

THEBESTUNTIL(*solution1*, *solution2*)

```
1  for f = 0 to numeroDeObjetivos do
2      somatorio1 += solution1.getFitness(f)
3      somatorio2 += solution2.getFitness(f)
4  for end
5  if (somatorio2 < somatorio1) then
6      return true
7  else
8      return false
```

---

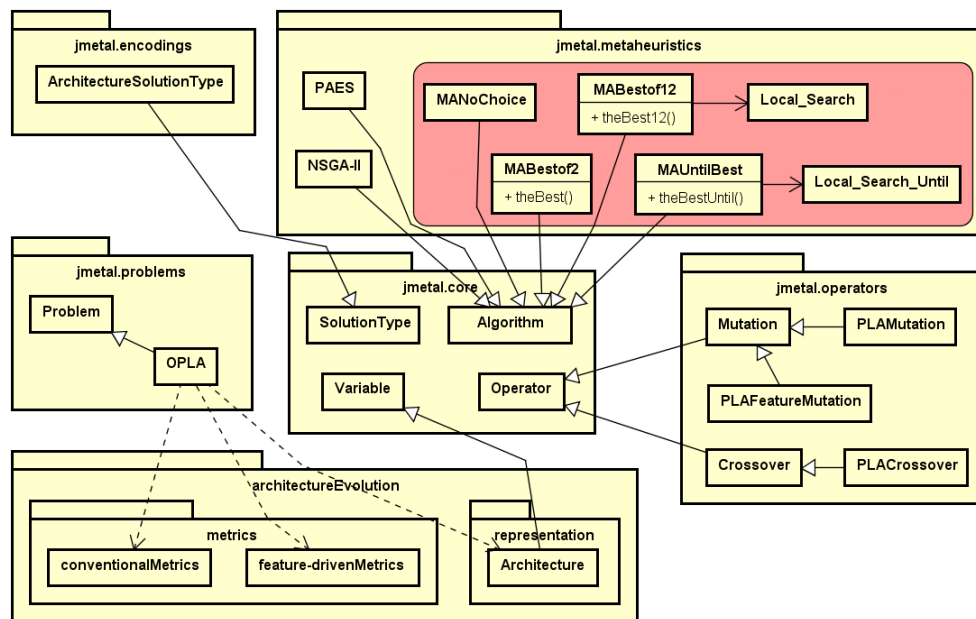
quinze. Se não houver melhora, a solução encontrada pela busca global (*sOri*) sobrevive para próxima geração.

## 4.2 Aspectos de Implementação

Esta seção apresenta os aspectos envolvidos na implementação dos AMs. Uma análise de código foi realizada em cinco módulos que compõem a ferramenta OPLA-Tool (OPLA-Core, OPLA-Decoding, OPLA-Encoding, OPLA-Patterns e OPLA-GUI) e na sequência foi desenvolvida a implementação da abordagem memética.

A Figura 4.2 mostra as principais classes existentes na OPLA-Tool. Na região rosa estão as classes criadas no desenvolvimento da abordagem memética.

Figura 4.2: Classes da OPLA-Tool e do jMetal.



Após o estudo da implementação atual, foi identificado que o AM seria implementado na classe NSGA-II. Por consequência, foram criadas quatro novas classes denominadas: *MANoChoice*, *MABestof2*, *MABestof12* e *MAUntilBest*. Nessas classes foi inserido o operador de busca local logo após o operador de mutação global, seguindo a implementação do pseudo-código apresentado no Algoritmo 1. A classe NSGA-II está no pacote *jmetal.metaheuristics*. Logo, as quatro novas classes estão no mesmo pacote, como pode ser visto da Figura 4.2.

As quatro versões do AM (NoChoice, Bestof2, Bestof12 e UntilBest) se diferenciam pelo critério de parada do operador de busca local. O algoritmo *MANoChoice* parava na primeira solução encontrada e a inseria na próxima geração sem qualquer avaliação.

Para o algoritmo Bestof2 foi implementado, na classe *MABestof2*, um método chamado *theBest* que comparava a solução encontrada pelo operador de busca global com a solução encontrada pelo operador de busca local, conforme o pseudo-código apresentado no Algoritmo 4.

O algoritmo Bestof12 utilizou o paralelismo na busca local. Para isso, foi utilizada uma classe nativa do Java chamada *Executors*, a partir da qual foi usado o método *newFixedThreadPool()* sendo instanciadas doze *threads*. Uma classe chamada *Local\_Search* foi criada com objetivo de executar o operador de busca local, para isso, a classe recebia uma cópia da solução do operador global para executar a busca. Desse modo, cada *thread* recebia uma chamada para a classe *Local\_Search* e assim foram executadas doze buscas locais para uma mesma solução de forma paralela. E, por fim, foi implementado um método chamado *theBest12* para selecionar qual a melhor solução da vizinhança.

Como o algoritmo NSGA-II lida com duas soluções a cada interação do laço de repetição, foi utilizada a mesma estrutura de paralelismo do *MABestof12* para chamar a busca local de forma simultânea para as duas soluções. A busca local foi implementada numa classe chamada *Local\_Search\_Until* conforme o pseudo-código do Algoritmo 7. Como houve a necessidade de comparar as soluções da busca global e local a cada tentativa do operador de busca local, o método *theBestUntil* (Algoritmo 8) foi implementado também na classe *Local\_Search\_Until*.

Atualmente a ferramenta OPLA-Tool disponibiliza dois algoritmos de otimização: NSGA-II e PAES. Com o desenvolvimento da abordagem memética a rotina da ferramenta não foi alterada, somente, foram inseridos os quatro novos algoritmos: NoChoice, Bestof2, Bestof12 e UntilBest.

### 4.3 Considerações Finais

Neste capítulo foi exposta a proposta de desenvolvimento da abordagem memética para otimização de projeto de PLA no contexto da MOA4PLA. Quatro algoritmos meméticos foram desenvolvidos: NoChoice, Bestof2, Bestof12 e UntilBest, cada um com um critério de parada distinto.

Em razão da abordagem memética ainda não ter sido investigada em outros trabalhos, foi necessário elaborar um método de pesquisa. Dessa forma, o próximo capítulo apresenta os materiais e métodos utilizados no desenvolvimento do presente trabalho.

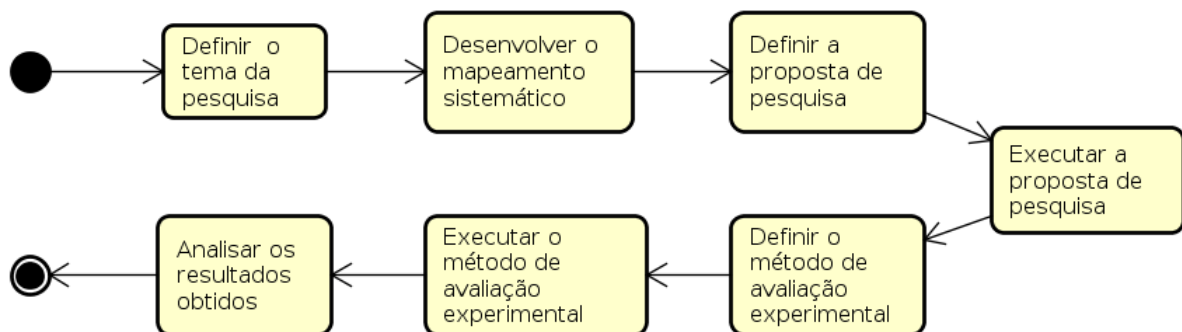
## Métodos e Materiais

Este capítulo aborda o método utilizado para condução do presente trabalho, bem como, os materiais necessários. Ainda, apresenta as PLAs utilizadas nos estudos empíricos e expõe as ameaças a validade dos experimentos.

### 5.1 Método de pesquisa

Para atingir o objetivo proposto a sequência de atividades, descritas a seguir, foi realizada. A Figura 5.1 apresenta a sequência dessas atividades.

**Figura 5.1:** Atividades realizadas no presente trabalho.



**Definir o tema da pesquisa:** a utilização de AMs em problemas de otimização estão encontrando resultados promissores, este fato fez com que fosse proposto uma abordagem memética para ser utilizada na MOA4PLA.



**Desenvolver o mapeamento sistemático:** a proposta foi definida com base no estado da arte sobre o tema de algoritmos meméticos, o qual, não apresentou nenhum trabalho que relaciona otimização de projetos de PLA com a utilização da abordagem memética.

**Definir a proposta de pesquisa:** a abordagem genética encontrou boas soluções com a utilização de todos os operadores de mutação fundamentais juntamente com o operador de aplicação de padrões de projeto. Isso fez com que fosse proposta uma abordagem memética para utilizar operador *Design Pattern Mutation Operator* como busca local, além, dos operadores de mutação fundamentais (Move Method, Move Attribute, Add Class, Move Operation, Add Manager Package and Feature-Driven Operator) na busca global, transformando o algoritmo NSGA-II em um algoritmo memético.

**Executar a proposta de pesquisa:** foi implementado o algoritmo memético em quatro diferentes versões: NoChoice, Bestof2, Bestof12 e UntilBest.

**Definir o método de avaliação experimental:** a avaliação consistiu em definir os experimentos que deveriam ser desenvolvidos para avaliar os algoritmos propostos, além de, comparar seu desempenho em relação ao algoritmo genético original. Foi necessário, também, definir os indicadores de qualidade e os testes estatísticos para avaliação.

**Executar o método de avaliação experimental:** foram desenvolvidos três experimentos sendo dois quantitativos subdividido em três etapas e um qualitativo. O experimento qualitativo foi desenvolvido com auxílio de especialistas. Os três experimentos foram desenvolvidos com objetivo de avaliar a abordagem memética.

**Analisar os resultados obtidos:** através de testes estatísticos foi possível analisar os resultados quantitativos trazidos pela abordagem memética, além de possibilitar a comparação com a abordagem genética. Já os resultados qualitativos foram interpretados subjetivamente com base na proposta trabalhada.

### 5.1.1 Método de avaliação

Para avaliação dos AMs desenvolvidos, foi proposto um conjunto de experimentos com o objetivo de analisar a qualidade das soluções encontradas pela abordagem memética, avaliar o desempenho dos AMs e, comparar quantitativamente a abordagem memética e genética.

A Tabela 5.1 sintetiza o método de avaliação. A primeira coluna apresenta o nome do experimento, a segunda coluna os objetivos do experimento, a terceira coluna qual o tipo de experimento foi realizado, a quarta coluna os algoritmos utilizados no experimento,

a quinta coluna as PLAs utilizadas no experimento, a sexta coluna as funções objetivo utilizadas no experimento e, por fim, a sétima coluna os dados analisados.

A princípio foi desenvolvido um estudo exploratório (Exp 01) com as quatro versões do algoritmo memético tendo como objetivo de analisar o comportamento da abordagem memética sobre o contexto de otimização de projeto de PLA usando a abordagem MOA4PLA. Na sequência, comparar os resultados encontrados pela abordagem memética com os resultados da abordagem genética. E, no fim, identificar qual a versão de AM foi mais eficaz.

Com a versão de AM mais eficaz identificada, foi desenvolvido um estudo experimental (Exp 02) com objetivo de comparar a qualidade das soluções entre a abordagem memética e genética. E, ainda neste experimento, foram utilizados dois pares de funções distintas, para identificar qual par de funções objetivo é mais apropriado para avaliar as mudanças realizadas pela busca local.

Por fim, um estudo qualitativo (Exp 03) foi realizado com auxílio de arquitetos de software. Neste experimento foram avaliadas algumas soluções encontradas no Exp 02.

### 5.1.2 Técnicas de análise quantitativa

Esta seção apresenta as técnicas utilizadas para analisar os resultados encontrados nos diversos experimentos realizados neste trabalho de dissertação, dessa forma, apresenta como e quais indicadores de qualidade e testes estatísticos foram utilizados.

#### Indicadores de Qualidade

Os indicadores de qualidade permitem avaliar o desempenho de algoritmos multiobjetivos e, com isso, é possível comparar os resultados de diferentes algoritmos. Alguns conjuntos de soluções comumente são utilizados:  $PF_{approx}$ ,  $PF_{known}$  e  $PF_{true}$ .

A  $PF_{approx}$  é formada pelas soluções não dominadas encontradas por uma execução de um algoritmo. A  $PF_{known}$  é a fronteira formada pelas soluções não dominadas dentre todas as  $PF_{approx}$  obtidas por um dado algoritmo. Alguns dos indicadores de qualidade necessitam da Fronteira de Pareto Real ( $PF_{true}$ ), no entanto, em problemas reais a  $PF_{true}$  não é conhecida. Neste caso, a  $PF_{true}$  é formada pelas soluções não dominadas da união das  $PF_{known}$  encontradas por todos os algoritmos avaliados (Yoo e Harman, 2007; Zitzler *et al.*, 2003).

Para mensurar o desempenho dos algoritmos, foram utilizados cinco diferentes indicadores de qualidade multiobjetivos, apresentados na sequência.

**Tabela 5.1:** Síntese do método de avaliação.

Nome do Experimento	Objetivo do Experimento	Tipo do Experimento	Algoritmos Utilizados	PLAs Utilizadas	Funções Objetivo Utilizadas	Dados Analisados
Estudo Exploratório (Exp 01)	(i) Investigar o comportamento do AM; (ii) Comparar seus resultados com os alcançados pelo AG; (iii) Definir qual versão de AM é mais eficaz no contexto da MOA4PLA.	Quantitativo e Qualitativo	Genético, Bestof2, Bestof12, NoChoice e UntilBest	AGM, BANK, MOM e BET	(CM, FM)	Fitness das funções objetivo, tempo de execução, indicadores de qualidade e soluções encontradas
Estudo Experimental (Exp 02)	(i) Comparar os resultados obtidos pelos algoritmos memético e genético; (ii) Analisar qual par de funções é mais apropriado para avaliar as mudanças realizadas pela busca local.	Quantitativo e Qualitativo	Bestof2 e Genético	AGM, BANK, MOM e BET	(CM, FM) e (COE, ACLASS)	Fitness das funções objetivo, indicadores de qualidade e soluções encontradas
Estudo Qualitativo (Exp 03)	(i) Avaliar qualitativamente algumas soluções encontradas no Exp 02.	Qualitativo	Bestof2	AGM, BANK e MOM	(CM, FM) e (COE, ACLASS)	Soluções encontradas no Exp 02 e respostas dos avaliadores

### Hypervolume

O Hypervolume (HV) mede a área de cobertura que uma PF exerce sobre o espaço de objetivos (Coello *et al.*, 2007). Comparando duas  $PF_{approx}$  sempre que uma  $PF_{approx}$  domina completamente a outra, o HV da primeira será maior do que o HV da segunda. O indicador calcula o volume da região entre um ponto de referência e uma  $PF_{approx}$ . Esse ponto é escolhido de forma que seja dominado por todas as  $PF_{approx}$ . Ao se trabalhar com um problema de minimização define-se um ponto maior que a pior solução alcançada (Zitzler *et al.*, 2001).

Quanto maior o valor de HV, maior será a área de cobertura, refletindo assim, uma melhor fronteira. Como pode ser visto na Equação 5.1.2, o cálculo é a união dos volumes alcançados por todas as soluções encontradas. Para facilitar o cálculo, os valores utilizados foram normalizados e, o valor utilizado como ponto de referência foi 1,01.

$$\bigcup_i vol_i : vec_i \in PF_{know}$$

### *Euclidean Distance to the Ideal Solution*

O indicador de *Euclidean Distance to the Ideal Solution* (ED) não é um indicador de qualidade, em vez disso, é usado como uma medida para ajudar o tomador de decisão a selecionar uma solução dentre todas as fornecidas. A ED encontra a solução mais próxima dos melhores objetivos. Para isso a solução ideal tem o valor mínimo de cada objetivo, considerando um problema de minimização. Dessa forma, a ED mensura a distância entre uma solução N e a solução ideal em um espaço de soluções (Zeleny e Cochrane, 1973).

### Generational Distance

O indicador de qualidade *Generational Distance* (GD) é uma medida de erro usada para examinar a convergência de um algoritmo para a  $PF_{true}$ . O GD calcula a distância existente entre a  $PF_{known}$  e a  $PF_{true}$  com objetivo de identificar o quanto a  $PF_{known}$  está próxima da fronteira ideal (Van Veldhuizen e Lamont, 1998). Para cada solução na  $PF_{known}$  calcula-se a ED da solução mais próxima na  $PF_{true}$ , e por fim, uma média desses valores é realizada para encontrar o valor de GD (Van Veldhuizen e Lamont, 1998).

### Inverse Generational Distance

O *Inverse Generational Distance* (IGD) é um indicador baseado no GD, mas com objetivo de avaliar a média da ED da  $PF_{true}$  a  $PF_{known}$ , ou seja, o inverso de GD (Van Veldhuizen e Lamont, 1998).

### Error Ratio

O *Error Ratio* (ER) é uma medida de erro para relatar o número finito de membros da  $PF_{known}$  que não estão na  $PF_{true}$ . Se eles não estão na  $PF_{true}$ , o algoritmo errou ou talvez

não tenha convergido. Quanto menor o ER melhor é o desempenho do algoritmo pois, um número maior de membros da  $PF_{known}$  foram encontrados na  $PF_{true}$  (Van Veldhuizen, 1999).

Para avaliar os resultados encontrados pelos indicadores de qualidade foram utilizados dois testes estatísticos: *Shapiro Wilk* e *Friedman*, apresentados na seção seguinte.

## Testes Estatísticos

### Teste de Normalidade

O teste de *Shapiro Wilk* foi utilizado para averiguar se os conjuntos amostrais têm distribuição normal (Royston, 1982). Este teste foi aplicado com 95 % de confiança ( $p\text{-value} < 0,05$ ) para as seguintes hipóteses:

- $H_{0shapiro}$  : A amostra tem distribuição normal;
- $H_{1shapiro}$  : A amostra não tem distribuição normal.

### Teste de *Wilcoxon*

O teste de *Wilcoxon* é um teste não-paramétrico cuja hipótese nula ( $H_{0wilcoxon}$ ) indica que a diferença entre os pares segue uma distribuição simétrica em torno de zero e, cuja hipótese alternativa ( $H_{1wilcoxon}$ ) indica que a diferença entre os pares não segue uma distribuição simétrica em torno de zero (Arcuri e Briand, 2011). Realizando o teste com 95% de confiança, encontrando  $p\text{-value} < 0,05$ , é possível refutar  $H_{0wilcoxon}$  indicando que há diferença estatística entre as abordagens comparadas.

- $H_{0wilcoxon}$  : Não há diferença estatística entre as amostras A e B;
- $H_{1wilcoxon}$  : Há diferença estatística entre as amostras A e B.

### Teste de *Friedman*

O teste de *Friedman* é uma extensão do teste de *Wilcoxon*. Ele é utilizado para três ou mais amostras experimentais com objetivo de comparar dados amostrais vinculados. Os valores numéricos dos dados não são utilizados diretamente, mas sim, as posições ocupadas por eles e, dessa forma, um teste de igualdade das somas dos grupos de resultados é realizado. Por esse teste de igualdade é possível interpretar se há ou não diferença entre as amostras.

Com 95% de confiança,  $H_{0friedman}$  indica que não há diferença estatística entre as amostras e, como hipótese alternativa,  $H_{1friedman}$  indica que há diferença estatística entre as amostras.

- $H_{0friedman}$  : Não há diferença estatística entre as amostras A, B, ..., N;
- $H_{1friedman}$  : Há diferença estatística entre as amostras A, B, ..., N.

Todos os testes estatísticos (*Shapiro Wilk*, *Wilcoxon* e *Friedman*) foram utilizados com 95% de confiança, ou seja,  $p\text{-value} < 0,05$ .

## 5.2 Materiais

Esta seção apresenta os materiais utilizados nos estudos experimentais desenvolvidos neste trabalho. Todos os experimentos foram rodados na ferramenta OPLA-Tool (Seção 2.3.2). Na sequência estão outros materiais que auxiliaram os experimentos e a análise de resultados.

### 5.2.1 Linguagem e Ferramenta R

R é uma linguagem para computação estatística que oferece um ambiente para geração de gráficos. É um projeto GNU permite modelagens lineares e não-lineares, testes estatísticos, agrupamentos entre outras opções estatísticas, sendo dessa forma, altamente extensível (R, 2017). Os testes estatísticos utilizados no presente trabalho foram aplicados utilizando essa ferramenta na versão 3.2.3 (2015-12-10).

### 5.2.2 Iramuteq

O software IRAMUTEQ (Interface de R pour lês Analyses Multidimensionnelles de Textes et de Questionnaires) é uma ferramenta francesa que permite análises estatísticas de textos. Desenvolvido por Ratinaud (2017) é um programa livre que se ancora na ferramenta R.

Para análises dos conteúdos textuais é necessário a construção de um *corpus* textual composto por um conjunto de textos que se pretende analisar. A análise é realizada por meio de métodos de classificação. Os métodos utilizados neste trabalho foram: classificação hierárquica descendente, análise de similitude e nuvem de palavras.

A ferramenta Iramuteq permite realizar a Classificação Hierárquica Descendente (CHD), método proposto por Reinert (1990). Esse método permite identificar classes entre os segmentos de texto. Os segmentos que possuem um vocabulário semelhante são agrupados e separados dos segmentos com vocabulário distinto. Um dendograma é utilizado para ilustrar os relacionamentos existentes entre cada classe (Ratinaud, 2017).

A Análise de similitude baseia-se na teoria dos grafos e permite identificar conexidade entre as palavras auxiliando na identificação da estrutura de representação (Ratinaud, 2017).

Outra classificação disponibilizada pela ferramenta Iramuteq é a Nuvem de Palavras. Em função da frequência de cada palavra, presente no *corpus* textual, um grupo de palavras é gerado e organizado graficamente (Ratinaud, 2017). Dessa forma, quanto maior a palavra na nuvem maior a sua frequência no *corpus*.

Neste trabalho a ferramenta Iramuteq foi utilizada na versão 0.7 alpha 2.

### 5.2.3 PLAs utilizadas

Em todos os experimentos foi necessário a utilização de PLAs como variável independente. Neste contexto, foram utilizados quatro projetos de PLA: Arcade Game Maker (AGM) é uma LPS que inclui três jogos Brickles, Bowling e Pong, desenvolvida pelo Software Engineering Institute (SEI) (Clements e Northrop, 2007); Banking System (BANK) é uma LPS criada por (Gomaa, 2011) com a finalidade de gerenciar sistemas bancários; Bilhetes Eletrônicos de Transporte Urbano (BET) de (Donegan e Masiero, 2007) é uma LPS utilizada para gerenciar o transporte urbano; Mobile Media (MOM) para controle de mídia em dispositivos moveis (Contieri Jr *et al.*, 2011). A quantidade de elementos arquiteturais por PLA está apresentada na Tabela 5.2.

**Tabela 5.2:** Informações das PLAs

PLA	#Componentes	#Interfaces	#Classes	#Features	#Variabilidades
AGM	9	14	30	11	5
BANK	4	5	25	16	3
MOM	8	15	14	14	7
BET	56	30	115	18	8

## 5.3 Ameaças à Validade

Esta seção apresenta as ameaças à validade consideradas nos experimentos realizados. O tamanho e a diversidade das PLAs utilizadas nos experimentos é uma ameaça em razão de que apenas 4 PLAs foram usadas, onde, apenas uma delas é uma LPS real (BET). No entanto, a dificuldade de encontrar projetos reais com o detalhamento necessário para realizar os experimentos é um fator limitante para o presente trabalho. No que diz respeito à diversidade, os projetos de PLA possuem diferentes tamanhos e são de domínios distintos minimizando esta ameaça.

Outra ameaça está relacionada aos algoritmos serem não-determinísticos. Para mitigar esta ameaça, foram realizadas 15 repetições do Exp 01 - Etapa 01 e 30 repetições dos experimentos Exp 02 - Etapas 01 e 02, como sugerido em (Fraser *et al.*, 2015). Foi utilizado também indicadores de qualidade, geralmente utilizados nos trabalhos de SBSE.

A adoção do mesmo tamanho de população, o mesmo número de gerações independentemente do tamanho de PLA e as funções de *fitness* CM e FM são outras ameaças. Contudo, os trabalhos de Guizzo (2014) e Colanzi (2014) utilizaram as mesmas configurações e alcançaram resultados satisfatórios. Outras configurações podem ser realizadas em estudos futuros.

Uma ameaça evidente na realização do experimento Exp 03 está na dificuldade de encontrar um número considerável de avaliadores especializados em projetos de PLA. Ainda, a maioria dos participantes foi entrevistado remotamente o que pode causar um viés nas respostas ao passo que não foi possível apresentar os projetos impressos o que facilitaria a visão geral da arquitetura.

## 5.4 Considerações Finais

Esse capítulo descreveu os métodos e materiais utilizados para alcançar os objetivos do presente trabalho.

O método de pesquisa apresentado englobou a organização do método de avaliação, bem como, a técnica de análise quantitativa, expondo os indicadores de qualidade e testes estatísticos multiobjetivo. Por fim, foram apresentadas os projetos de PLA utilizados, além das ameaças a validade dos experimentos.

O próximo capítulo apresenta os projetos dos estudos experimentais e, também, os resultados alcançados nas suas execuções.



---

## Resultados e Discussões

---

Este capítulo apresenta três estudos empíricos (Exp 01, Exp 02 e Exp 03) para teste e avaliação da abordagem memética. Os projetos dos estudos desenvolvidos e os resultados obtidos nas suas execuções são apresentados nas próximas seções. Ainda, são respondidas as questões de pesquisa levantadas em cada experimento.

### 6.1 Estudo Exploratório - Exp 01

#### 6.1.1 Projeto do Estudo

O primeiro experimento (Exp 01) foi um estudo exploratório desenvolvido com objetivo de analisar o desempenho da abordagem memética para otimização de projeto de PLA. Para isso foram desenvolvidas duas questões de pesquisa: **QP1:** *O algoritmo memético encontra soluções de melhor qualidade do que as soluções obtidas pelo algoritmo genético no contexto de otimização de projeto de PLA usando a MOA4PLA?* **QP2:** *Qual abordagem de busca local é mais eficaz no referido contexto?*

O Exp 01 foi dividido em três etapas sendo a primeira uma análise quantitativa dos resultados encontrados pelos indicadores de qualidade, a segunda etapa foi uma contabilização da aplicação de padrões de projeto nas soluções do AM e, a terceira etapa analisa qualitativamente as soluções alcançadas pela abordagem memética.

#### Exp 01 - Etapa 01

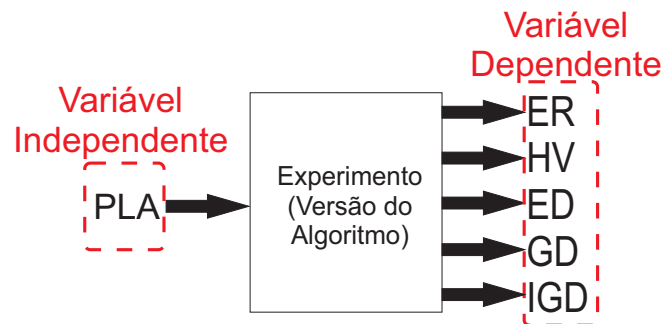
A primeira etapa do Exp 01 (Exp 01 - Etapa 01) teve como objetivo comparar quantitativamente o desempenho das abordagens memética e genética com a seguinte hipótese: a

abordagem memética tem melhor desempenho do que a abordagem genética com relação aos indicadores ER, HV, ED, GD e IGD.

O Exp 01 - Etapa 01 utilizou a ferramenta OPLA-Tool com as seguintes configurações: 15 execuções do algoritmo NSGA-II, com todos os operadores de mutação fundamentais com probabilidade de mutação de 0.9; sem a utilização do Crossover; tamanho da população igual a 100; e o número de avaliações de *fitness* igual a 30000. O número de avaliações de *fitness* foi utilizado como critério de parada. As funções objetivo utilizadas foram CM e FM (Seção 2.3.1). A variável independente é um projeto de PLA e as variáveis dependentes são os indicadores de qualidade ER, HV, ED, GD e IGD, como pode ser visto da Figura 6.1.

O Exp 01 - Etapa 01 foi repetido para cada versão de AM (NoChoice, Bestof2, Bestof12 e UntilBest) e para o AG.

**Figura 6.1:** Projetos Experimentais Exp 01 - Etapa 01.

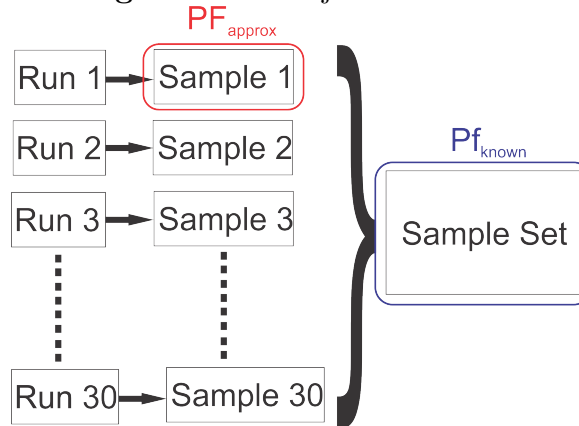


A Figura 6.2 representa como cada uma das versões do Exp 01 foi considerada. Um experimento Exp 01 foi repetido 15 vezes para cada variável independente utilizada, cada repetição forma uma  $PF_{approx}$  com as soluções não dominadas. A união das 15 amostras de resultados forma um conjunto amostral que é uma  $PF_{known}$  com as soluções não dominadas dentre todas as soluções encontradas em todas as rodadas. Cada uma das versões do Exp 01 utilizou as quatro PLAs (AGM, BANK, BET e MOM) como variável independente, uma por vez. Com essa estruturação vão existir quatro conjuntos amostrais para cada versão de algoritmo, onde, cada conjunto refere-se a uma PLA, resultando em 20 conjuntos amostrais. São nesses conjuntos que são aplicados os indicadores de qualidade.

Os 20 conjuntos amostrais do Exp 01 - Etapa 01 foram avaliados por cinco indicadores de qualidade: HV, ED, GD, IGD, e ER, descritos na Seção 5.1.2.

Como hipótese nula ( $H_0$ ) não há diferença estatística entre os resultados dos indicadores de qualidade das abordagens memética e genética, na hipótese alternativa  $H_1$  os resultados dos indicadores de qualidade da abordagem genética supera a abordagem

**Figura 6.2:** Conjunto amostral.



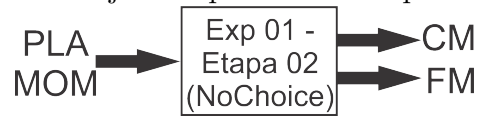
memética e, a hipótese alternativa  $H_2$  os resultados dos indicadores de qualidade da abordagem memética supera a abordagem genética.

- $H_0 : ER_{MA} = ER_{GA}$  and  $HV_{MA} = HV_{GA}$  and  $ED_{MA} = ED_{GA}$  and  $GD_{MA} = GD_{GA}$  and  $IGD_{MA} = IGD_{GA}$ ;
- $H_1 : ER_{MA} > ER_{GA}$  and  $HV_{MA} < HV_{GA}$  and  $ED_{MA} > ED_{GA}$  and  $GD_{MA} > GD_{GA}$  and  $IGD_{MA} > IGD_{GA}$ ;
- $H_2 : ER_{MA} < ER_{GA}$  and  $HV_{MA} > HV_{GA}$  and  $ED_{MA} < ED_{GA}$  and  $GD_{MA} < GD_{GA}$  and  $IGD_{MA} < IGD_{GA}$ ;

### Exp 01 - Etapa 02

A segunda etapa do Exp 01 (Exp 01 - Etapa 02) foi realizada com o intuito de monitorar o comportamento das aplicações de padrões de projeto, realizadas pela busca local, no decorrer do processo de evolução das gerações no MA.

O Exp 01 - Etapa 02 utilizou a abordagem memética do algoritmo NSGA-II na versão NoChoice. As seguintes configurações foram utilizadas: 1 execução do algoritmo NSGA-II, com todos os operadores de mutação fundamentais com probabilidade de mutação de 0.9; sem a utilização do Crossover; tamanho da população igual a 100; e o número de avaliações de *fitness* igual a 10.000. O número de avaliações de *fitness* foi utilizado como critério de parada. O projeto experimental pode ser visto na Figura 6.3 onde foi utilizada a PLA MOM como variável independente e como variáveis dependentes as funções objetivo CM e FM. Foram armazenadas cada uma das 10.000 soluções avaliadas, com objetivo de averiguar a presença de padrões de projeto e assim, mensurar a taxa de aplicação durante a evolução.

**Figura 6.3:** Projeto Experimental Exp 01 - Etapa 02.

### Exp 01 - Etapa 03

A terceira etapa do Exp 01 (Exp 01 - Etapa 03) teve como objetivo avaliar qualitativamente as soluções encontradas pela abordagem memética no Exp 01 - Etapa 01. A análise procurou identificar possíveis inconsistências no projeto e regiões propícias a aplicação de padrões. As soluções consideradas para este estudo foram as que apresentaram algum padrão aplicado ao projeto de PLA ou as soluções com menor ED em relação às outras soluções.

O Exp 01 é composto por três etapas. Cada etapa com seus respectivos resultados estão apresentadas na sequência <sup>1</sup>.

## 6.1.2 Resultados

### Exp 01 - Etapa 01

O teste de normalidade de *Shapiro Wilk* e os indicadores de qualidade (ER, HV, ED, GD e IGD) foram aplicados nos *fitness* das soluções que compõe os 20 conjuntos amostrais encontrados pelo Exp 01 - Etapa 01. Com exceção da PLA BET no AG, que apresentou distribuição normal, todas as amostras obtiveram distribuição não normal. Para esse contexto o teste de *Friedman* foi o mais indicado. Os resultados do teste de normalidade são apresentados na Tabela 6.1.

Não foi possível realizar o teste de normalidade para a amostra encontrada pelo AG utilizando a PLA BANK pois, ela retornou valores de *fitness* idênticos para todas as rodadas sendo eles: CM = 141,5 e FM = 294. Comparando-se com o valor de *fitness* original (CM = 157,0 e FM = 326,0) fica claro a otimização, no entanto, o AM Bestof2 encontrou uma solução que supera a do AG, com CM = 141,5 e FM = 145,0, indicando que a solução encontrada pelo AG não é ótima.

A Tabela 6.2 apresenta o tempo de otimização de uma solução de projeto de PLA em milissegundos. Era esperado que o tempo de execução do algoritmo memético superasse o tempo do algoritmo genético. Ao executar as versões do AM Bestof12 e UntilBest para

<sup>1</sup>O pacote experimental que engloba os projetos de PLA, os resultados dos indicadores de qualidade, dados estatísticos e as soluções obtidas estão disponíveis em <https://github.com/Oplamemetic/experimentalpackage/>

Tabela 6.1: Exp 01 - Etapa 01 - Teste de Normalidade.

Versão do Algoritmo	Variável Independente	Variáveis Dependentes			
		CM		FM	
		<i>Shapiro Wilk</i>			
		<i>p-value</i>	Condição	<i>p-value</i>	Condição
Genético	AGM	0,02842	Não-normal	0.009651	Não-normal
	BANK	—	Não Aplicado	—	Não Aplicado
	MOM	0,004414	Não-normal	7,789e-07	Não-normal
	BET	1,471e-07	Não-normal	0,06863	Normal
Bestof2	AGM	1,625e-06	Não-normal	1,858e-10	Não-normal
	BANK	1,408e-06	Não-normal	1,102e-09	Não-normal
	MOM	1,015e-05	Não-normal	1,916e-11	Não-normal
	BET	5,705e-14	Não-normal	2,336e-14	Não-normal
Bestof12	AGM	9,879e-10	Não-normal	6,805e-07	Não-normal
	BANK	1,459e-06	Não-normal	1,104e-10	Não-normal
	MOM	8,98e-05	Não-normal	2,454e-09	Não-normal
	BET	—	Interrompido	—	Interrompido
NoChoice	AGM	4,819e-08	Não-normal	2,11e-11	Não-normal
	BANK	4,711e-05	Não-normal	8,316e-06	Não-normal
	MOM	2,19e-06	Não-normal	1,16e-10	Não-normal
	BET	3,869e-11	Não-normal	1,497e-11	Não-normal
UntilBest	AGM	1,298e-09	Não-normal	1,382e-10	Não-normal
	BANK	1,176e-05	Não-normal	1,634e-10	Não-normal
	MOM	0,02084	Não-normal	5,216e-10	Não-normal
	BET	—	Interrompido	—	Interrompido

a PLA BET (a maior PLA do experimento), detectou-se que seria inviável terminar suas execuções. Para realizar 30.000 avaliações o AM Bestof12 levaria aproximadamente 10,7 dias por rodada, para 15 rodadas 161,5 dias, já o AM UntilBest levaria cerca de 22,6 dias por rodada, para 15 rodadas seriam gastos 339,7 dias. Por essa razão, para a PLA BET foram utilizadas somente as amostras encontradas na execução dos algoritmos Bestof2, NoChoice e Genético.

A Tabela 6.3 apresenta a quantidade de soluções que compõem o  $PF_{true}$  do Exp 01 bem como a cardinalidade do  $PF_{known}$  obtido pelos algoritmos. Essa tabela apresenta ainda o *Error Ratio* (ER) de cada algoritmo. O  $PF_{known}$  e o ER estão indicados como  $PF_k$  e ER, respectivamente. O valor do menor ER está destacado em negrito. Os dados dessa tabela indicaram que a abordagem memética, com o algoritmo Bestof2, encontrou uma maior diversidade de soluções, pois acabou por cobrir uma maior área no espaço de

**Tabela 6.2:** Exp 01 - Etapa 01 - Tempo de otimização de uma solução em milisegundos.

PLA	Bestof2	Bestof12	NoChoice	UntilBest
AGM	161,9	999,5	124,4	1.651,5
BANK	65,3	244,8	34,6	412,0
MOM	197,5	836,4	98,6	1.889,9
BET	2.468,06	31.023,8	2.670	65.223,2

**Tabela 6.3:** Exp 01 - Etapa 01 - Número de soluções não dominadas por algoritmo.

PLA	#PF <sub>true</sub>	Genético		Bestof2		Bestof12		NoChoice		UntilBest	
		#PF <sub>k</sub>	ER	#PF <sub>k</sub>	ER	#PF <sub>k</sub>	ER	#PF <sub>k</sub>	ER	#PF <sub>k</sub>	ER
AGM	16	12	0,82	12	<b>0,18</b>	6	0,80	10	0,67	16	1
BANK	3	1	1	3	<b>0</b>	4	0,5	3	1	5	1
MOM	12	7	<b>0,42</b>	8	0,62	4	0,75	5	0,6	6	0,66
BET	13	13	0,69	10	<b>0,2</b>	—	—	6	0,75	—	—

soluções, apresentando menor ER em três das quatro PLAs utilizadas (AGM, BANK e BET).

Um caso especial aconteceu com a PLA BANK. O AG encontrou somente uma solução não dominada durante todas as rodadas. Já, os algoritmos da abordagem memética conseguiram encontrar um maior número de soluções, com mais de uma solução na PF<sub>known</sub>, contudo, somente os algoritmos Bestof2 e Bestof12 conseguiram soluções na PF<sub>true</sub> indicando um ER de valores 0 e 0,5, respectivamente.

O indicador de HV foi aplicado sobre as amostras normalizadas encontradas pelos algoritmos Genético, Bestof2, Bestof12, NoChoice e UntilBest. A Tabela 6.4 apresenta somente as médias de HV encontradas por algoritmo e na última coluna apresenta os resultados do teste de *Friedman*. As maiores médias de HV estão identificadas em negrito, pois representam a versão do algoritmo que obteve melhor rendimento.

O algoritmo Bestof2 obteve o maior valor de HV nas PLAs AGM e BET, já para as PLAs BANK e MOM, o algoritmo Bestof12 se destacou. Dessa forma, os algoritmos Bestof2 e Bestof12 encontraram melhores resultados em todas as PLAs utilizadas, superando assim, o AG em termos de HV.

Como pode ser observado na Tabela 6.4 há diferença estatística entre os algoritmos nas PLAs BANK e BET. Neste caso os algoritmos Bestof12 e Bestof2 obtiveram melhor desempenho, respectivamente.

O indicador de ED foi aplicado na PF<sub>true</sub> de todos os algoritmos do Exp 01 - Etapa 01. A Tabela 6.5 apresenta os valores de *fitness* original e ideal de cada PLA utilizada no Exp 01. A solução ideal pode não existir no conjunto de soluções, uma vez que ela é composta pelos menores valores das funções CM e FM.

**Tabela 6.4:** Exp 01 - Etapa 01 - Média dos valores de HV.

PLA	Genético	Bestof2	Bestof12	NoChoice	UntilBest	<i>p-value</i>
AGM	0,200845	<b>0,227553</b>	0,213305	0,209683	0,220646	0,28025
BANK	0,108785	0,271241	<b>0,316004</b>	0,195562	0,185855	<b>1,29e-09</b>
MOM	0,425223	0,437447	<b>0,447239</b>	0,382441	0,358863	0,428092
BET	0,223165	<b>0,245368</b>	—	0,162444	—	<b>0,001273</b>

**Tabela 6.5:** *Fitness* das Soluções Original e Ideal.

PLA	Solução Original	Solução Ideal
AGM	(89,14; 758,0)	(39,08; 554,0)
BANK	(157,0; 326,0)	(141,5; 271,0)
MOM	(72,14; 1122,0)	(60,08; 807,0)
BET	(461,02; 1486,0)	(312,02; 1343,0)

A Tabela 6.6 apresenta os valores de ED das soluções, por algoritmo, que mais se aproximaram da solução ideal. O menor valor de ED, por PLA, está realçado em negrito. A última coluna da Tabela 6.6 apresenta os valores encontrados pelo teste de *Friedman*.

Observando os menores valores de EDs alcançados fica evidente que o algoritmo Bestof2 apresentou uma melhor performance em todas as PLAs e, o NoChoice empatou com o Bestof2 nas PLAs AGM e BET. Contudo, o teste de *Friedman*, mostrou diferença estatística apenas para as PLAs AGM, BANK e BET. Nestes três casos, os algoritmos com melhor desempenho foram: Bestof2 e Bestof12, respectivamente. Em geral, pode-se dizer que os resultados indicaram que a abordagem memética era igual ou ligeiramente superior às soluções encontradas pelo AG.

O próximo indicador de qualidade aplicado foi o GD. A Tabela 6.7 apresenta as médias dos valores de GD para todos os algoritmos do Exp 01 - Etapa 01. A última coluna apresenta os valores obtidos com a aplicação do teste de *Friedman*.

Na Tabela 6.7, é possível identificar que o algoritmo Bestof2 encontra melhores resultados nas PLAs AGM e BET, já o Bestof12 nas PLAs BANK e MOM. Contudo, somente as PLAs BANK e BET apresentaram *p-value* < 0,05 indicando diferença estatísticas entre as amostras. O algoritmo Bestof2 superou o AG nesses dois casos.

A Tabela 6.8 apresenta as médias dos valores do indicador IGD por algoritmo. Na última coluna estão os valores de *p-value* encontrados pelo teste de *Friedman*.

O algoritmo Bestof2 obteve melhor desempenho na PLA BET, o Bestof12 nas PLAs BANK e MOM, e o UntilBest na AGM. Como no indicador GD, apenas as PLAs BANK e BET apresentaram *p-value* < 0,05 indicando diferença estatística, onde o algoritmo Bestof2 superou o AG.

**Tabela 6.6:** Exp 01 - Etapa 01 - Menores EDs por experimento.

PLA	Genético	Bestof2	Bestof12	NoChoice	UntilBest	<i>p-value</i>
AGM	268,312601	<b>51,0</b>	130,146071	<b>51,0</b>	57,428216	<b>0,003453</b>
BANK	149,656272	<b>14,0</b>	36,619666	54,083269	36,619666	<b>0,03837</b>
MOM	36,124029	<b>25,316541</b>	42,106359	37,0	36,674241	0,1847
BET	762,0	<b>162,000425</b>	—	<b>162,000425</b>	—	<b>0,0366</b>

**Tabela 6.7:** Exp 01 - Etapa 01 - Média dos valores de GD.

PLA	Genético	Bestof2	Bestof12	NoChoice	UntilBest	<i>p-value</i>
AGM	8,262073	<b>7,328262</b>	11,001264	7,574039	7,869226	0,5872
BANK	19,104973	7,840005	<b>6,289115</b>	17,577093	15,126942	<b>0,000042</b>
MOM	11,568652	9,484651	<b>9,023561</b>	10,802115	10,253804	0,7113
BET	10,307010	<b>7,025358</b>	—	21,678729	—	<b>0,003698</b>

**Tabela 6.8:** Exp 01 - Etapa 01 - Médias dos valores de IGD.

PLA	Genético	Bestof2	Bestof12	NoChoice	UntilBest	<i>p-value</i>
AGM	20,675618	19,455419	20,163540	19,708956	<b>18,790651</b>	0,2548
BANK	50,718176	43,639684	<b>39,874537</b>	44,345701	48,206297	<b>0,0000036</b>
MOM	3,781270	4,122948	<b>3,724295</b>	4,188424	4,171765	0,297
BET	53,477759	<b>48,773528</b>	—	59,169005	—	<b>0,007699</b>

Analisando as  $PF_{knows}$  dos algoritmos Genético, Bestof2, Bestof12, NoChoice e UntilBest do Exp 01 - Etapa 01, nenhuma ocorrência de padrões foi encontrada entre todas as soluções não dominadas. Isso levou a outro estudo para investigar se os padrões de projeto foram aplicados durante o processo de otimização e, a taxa de aplicação dos padrões. Os resultados deste estudo são apresentados na próxima subseção.

## Exp 01 - Etapa 02

O Exp 01 - Etapa 02 analisou a taxa de aplicação de padrões de projeto nas soluções geradas durante o processo de evolução de 100 gerações. Para facilitar a leitura dos resultados encontrados as 100 gerações são apresentadas em três gráficos. Cada padrão de projeto, sua combinação ou a não aplicação de padrão foi representada por um número, ficando da seguinte maneira:

1: *Strategy*

2: *Bridge*

3: *Mediator*



4: *Mediator e Strategy*

5: *Strategy e Bridge*

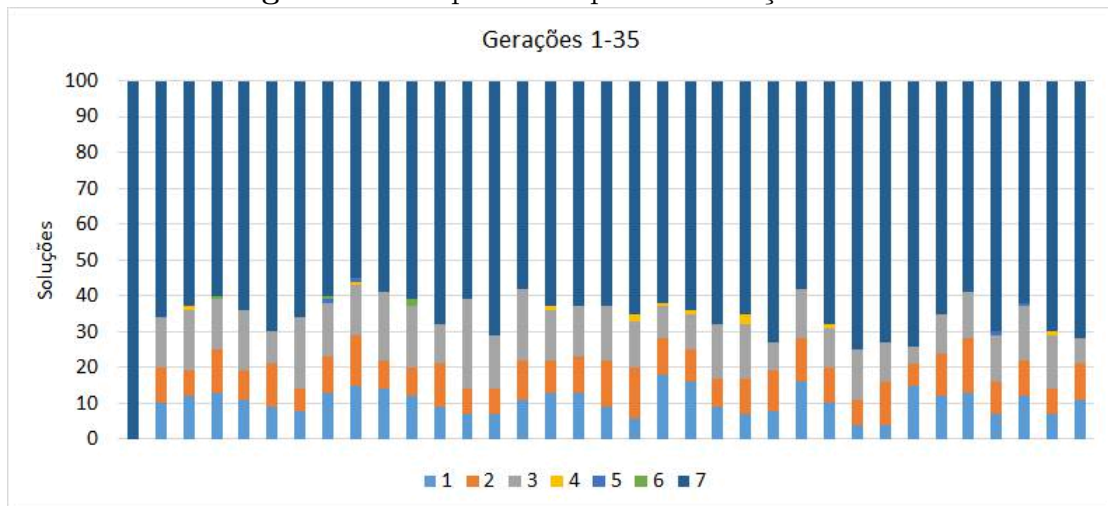
6: *Mediator e Bridge*

7: Nenhum padrão de projeto aplicado na PLA

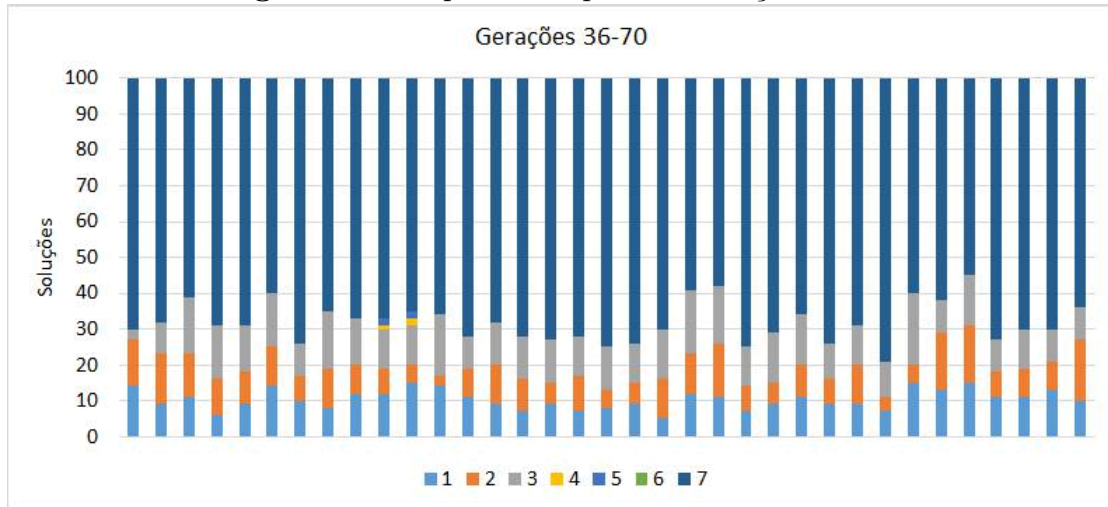
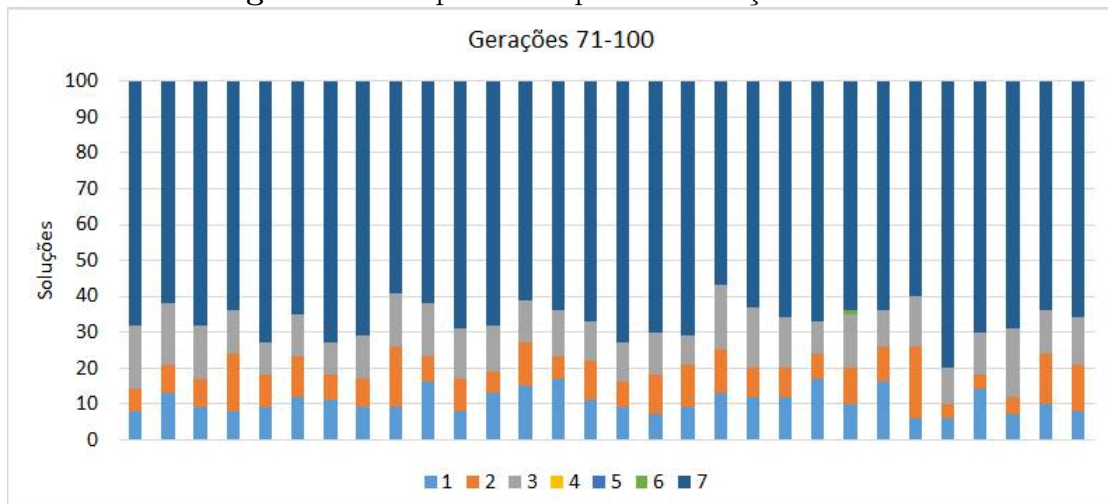
A combinação dos três padrões de projeto (*Mediator e Strategy e Bridge*) não foi considerada pois não houve ocorrência dela em nenhuma solução das 10.000 analisadas.

As Figuras 6.4, 6.5 e 6.6 apresentam a taxa de aplicação de padrões de projeto em cada uma das 100 gerações, sendo que cada coluna representa uma geração. A primeira coluna da Figura 6.4 indica que a população inicial não apresenta padrões de projeto aplicados em nenhum indivíduo, contudo, já na segunda geração, a taxa de aplicação se aproxima dos 40%. Ainda, pode-se observar que a aplicação de padrões está mais presente em 40% das soluções nas 35 primeiras gerações. No entanto, com a evolução das gerações, a taxa de aplicação de padrões está diminuindo de forma constante como pode ser visto na Figura 6.5 até o estágio em que aproximadamente 2/3 das soluções não têm padrão, como pode ser visto na Figura 6.6.

**Figura 6.4:** Exp 01 - Etapa 02 - Gerações 1 - 35

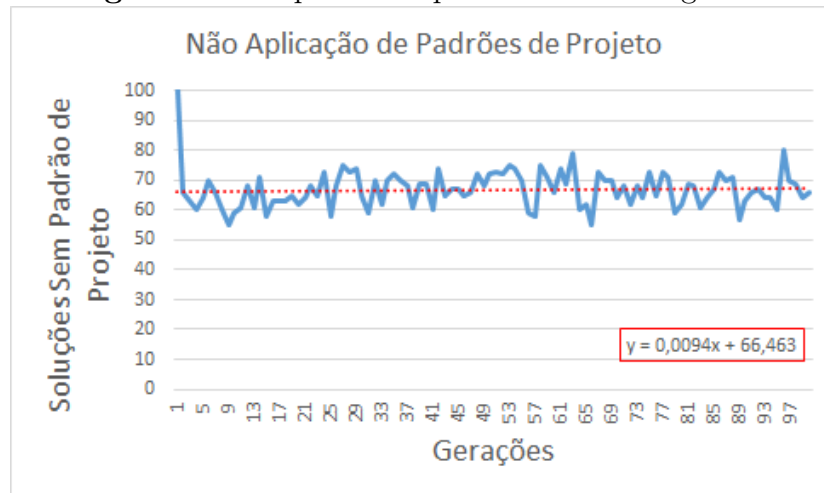


A Figura 6.7 apresenta a taxa de não aplicação de padrões durante o processo de evolução das gerações. A reta de regressão formada ( $y = 0,0094x + 66,463$ ) é positiva com uma leve inclinação. A porcentagem de aplicação de padrões de projeto diminui cerca de uma solução a cada 100 gerações, como pode ser visto na Figura 6.7.

**Figura 6.5:** Exp 01 - Etapa 02 - Gerações 36 - 70**Figura 6.6:** Exp 01 - Etapa 02 - Gerações 71 - 100

Embora pelo menos 30% das soluções tenham padrões aplicados, algumas soluções não sobreviveram ao longo das gerações ou foram dominadas por outras soluções e, portanto, não fazem parte das  $PF_{knowns}$ .

Um estudo qualitativo (Exp 03 - Etapa 03), apresentado na Subseção 6.1.2, foi desenvolvido a fim de compreender os motivos que levam ao descarte de soluções que contém padrão de projeto ao longo das gerações e, por que as soluções que contém algum padrão foram dominadas.

**Figura 6.7:** Exp 01 - Etapa 02 - Reta de regressão.**Exp 01 - Etapa 03**

O Exp 01 - Etapa 03 foca nas soluções de PLA encontradas pelos algoritmos Bestof2, Bestof12, NoChoice e UntilBest analisando pontos onde os padrões foram aplicados, bem como possíveis inconsistências na aplicação de padrões.

Inicialmente, foram analisadas as soluções com as EDs mais baixas de cada algoritmo. A Tabela 6.9 apresenta os valores de ED para cada solução analisada e a existência de padrão de projeto aplicado. Considerando as soluções analisadas, apenas a solução obtida pela versão do NoChoice para a PLA MOM teve o padrão *Strategy* aplicado. Todas as outras soluções não contém padrões.

Para a PLA AGM a solução com menor ED foi encontrada pelo AG. Entre as soluções geradas utilizando a PLA MOM, a solução com melhor ED foi encontrada pelo algoritmo UntilBest, esta solução mesmo sendo encontrada por um AM não apresentou padrões de projeto aplicado. Com exceção da solução VAR\_All.MobileMedia-8126871813, encontrada pelo algoritmo NoChoice, todas as outras soluções da Tabela 6.9 não apresentaram a aplicação de padrões de projeto.

Como somente uma solução de melhor ED apresentou aplicação de padrões, foi investigada a taxa de aplicação de padrões em todas as execuções dos AMs no Exp 01 - Etapa 01. A Tabela 6.10 mostra, na coluna 2, o número total de soluções obtidas para cada PLA com a execução de todos os AM no Exp 01 - Etapa 01. A terceira coluna da tabela mostra o número de soluções que continham pelo menos um padrão aplicado e a quarta coluna mostra a porcentagem de aplicação de padrões. Observa-se uma incidência de aplicação de padrões em cerca de 50% nas soluções obtidas pelas versões de AM nas PLAs BANK e BET e, porcentagens inferiores a 6% nos outras duas PLAs.

**Tabela 6.9:** Exp 01 - Etapa 03 - PLAs selecionadas - Parte 1

PLA	Soluções			
	Algoritmo	Identificação	Melhor ED	Padrões
AGM	Genético	VAR_All_agm-8913157441	<b>7,9660</b>	-
	Bestof2	VAR_All_agm-8777278214	51,0000	Não Aplicado
	Bestof12	VAR_All_agm-7121979371	55,0000	Não Aplicado
	NoChoice	VAR_All_agm-1771313264	49,0000	Não Aplicado
	UntilBest	VAR_All_agm-1627643334	53,0179	Não Aplicado
MOM	Genético	VAR_All_MobileMedia-1235244526	15,6205	-
	Bestof2	VAR_All_MobileMedia-6841372227	13,9861	Não Aplicado
	Bestof12	Var_All_MobileMedia-6964911226	10,7676	Não Aplicado
	NoChoice	VAR_All_MobileMedia-8126871813	14,7502	<b>Strategy</b>
	UntilBest	VAR_All_MobileMedia-3298114161	<b>9,9770</b>	Não Aplicado

**Tabela 6.10:** Exp 01 - Etapa 03 - Número de Soluções.

PLA	Total de Soluções	Total de Soluções com Padrão	Percentual de Aplicação (%)
AGM	298	6	<b>2,013423</b>
BANK	140	71	<b>50,714286</b>
MOM	225	13	<b>5,777778</b>
BET	202	96	<b>47,524752</b>

Deste ponto em diante, foi realizada uma análise dos projetos de PLAs com objetivo de identificar características estruturais como: padrões de projeto aplicados, anomalias e possíveis regiões para aplicação de padrões que não tiveram a efetiva aplicação.

Identificou-se que a PLA AGM original possui 3 escopos propícios à aplicação de padrão de projeto, sendo eles: um para o padrão *Bridge* e dois para o padrão *Strategy*. A família de algoritmo composta pelas classes *IPlayBowling*, *IPlayPong* e *IPlayBrickles* realizam uma característica em comum (*Play*) permitindo a aplicação do padrão de projeto *Bridge*, como pode ser visto na Figura 6.8. Contudo, como o operador seleciona de forma aleatória o conjunto de classes e interfaces que terão seu escopo de aplicação analisado, não há garantias que o algoritmo encontrará especificamente o conjunto favorável a aplicação de padrão de projeto.

A PLA MOM original possui três escopos de aplicação de padrões sendo eles: um para o padrão *Mediator* e dois para o padrão *Strategy*. A Figura 6.9 apresenta a família de algoritmos formada pelas classes *Music*, *Video* e *Photo* ligadas a uma classe comum *Media* permitindo a aplicação do padrão de projeto *Strategy*.

Figura 6.8: Escopo propício à aplicação do padrão *Bridge*.

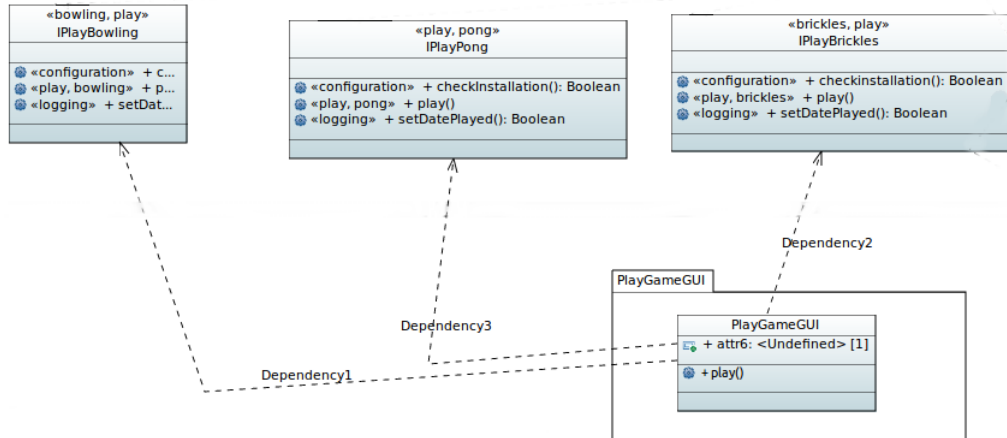
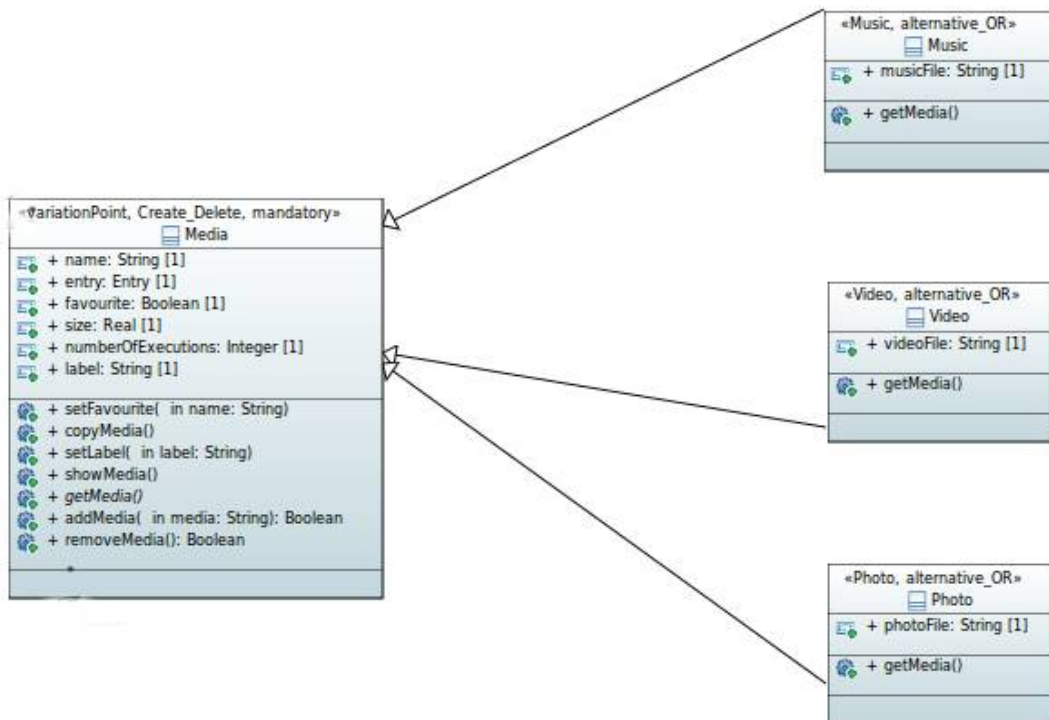


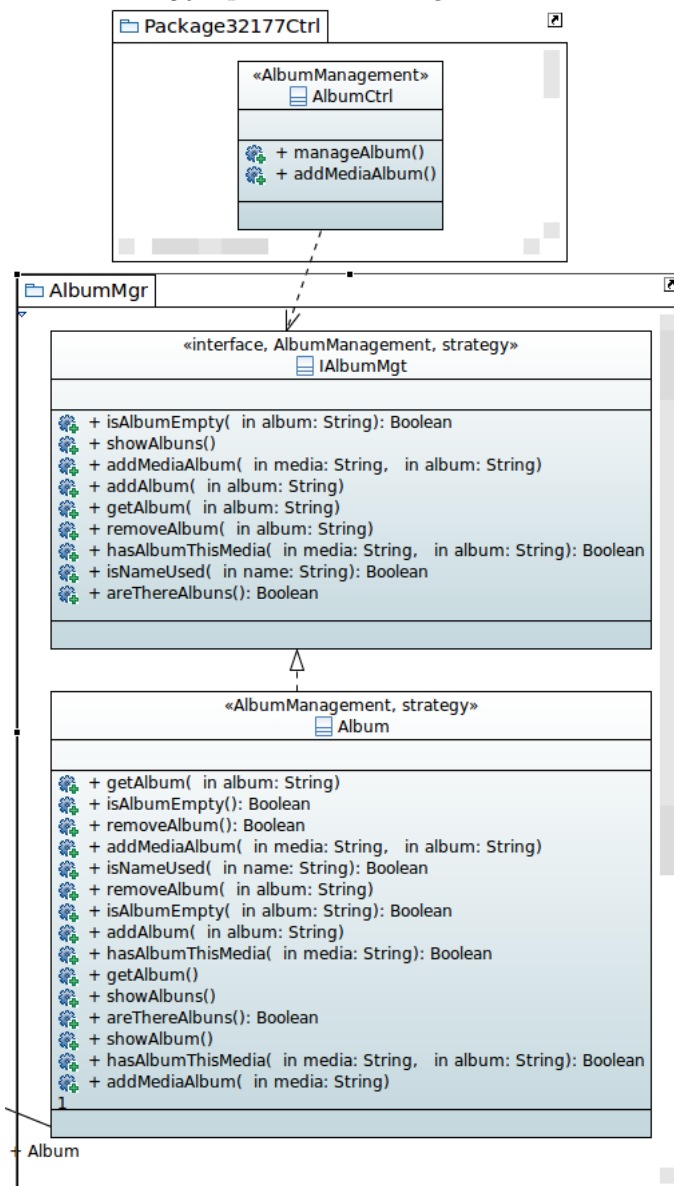
Figura 6.9: Escopo propício à aplicação do padrão *Strategy*.



O algoritmo NoChoice retornou uma solução para a PLA MOM cujo projeto contém o padrão *Strategy*, como pode ser visto na Figura 6.10. Analisando essa solução (Figura 6.10) detectou-se uma inconsistência na aplicação da padrão. O padrão *Strategy* aparece somente em duas classes. A interface *IAlbumMgt* representa a interface *Strategy* e a classe *Album* é um membro da família de algoritmos que implementa a interface *Strategy*. Além disso, a classe *AlbumCtrl* tem um relacionamento de dependência com a interface *Strategy*.

Embora a aplicação do padrão apresente todos os artefatos arquiteturais necessários para o padrão *Strategy*, a família de algoritmos contém um único membro. No entanto, é necessário pelos menos dois membros para caracterizar uma família. Como o operador de busca local só aplica o padrão *Strategy* caso haja uma família de algoritmos, é possível concluir que a aplicação do padrão está corrompida.

**Figura 6.10:** Padrão *Strategy* aplicado na solução da MOM obtida pelo NoChoice.

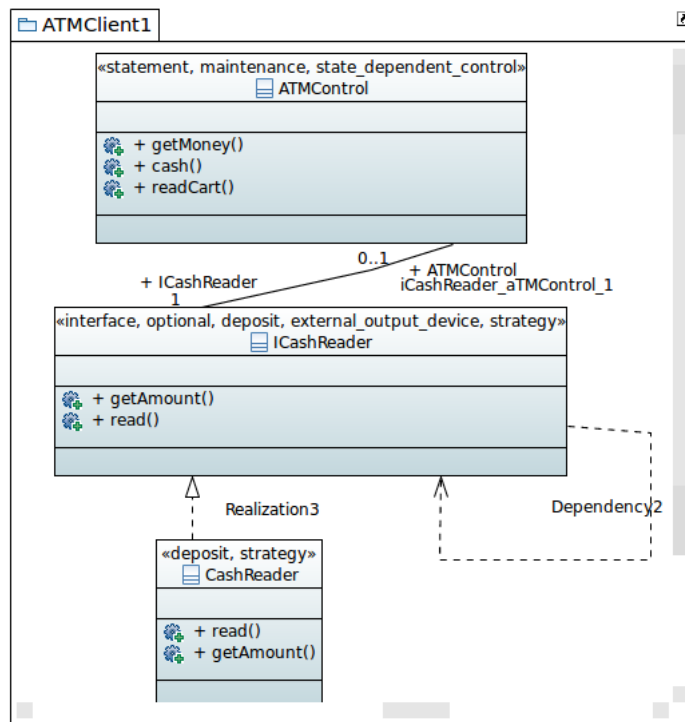


Como as  $PF_{knows}$  das PLAs BANK e BET não continham padrões e, em razão dessas PLAs apresentarem as maiores taxas de aplicação de padrões, as soluções analisadas

foram selecionadas aleatoriamente. As soluções da BANK continham apenas o padrão de *Strategy*, e as soluções da BET apresentaram os padrões *Strategy* e *Mediator*.

A Figura 6.11 apresenta parte de uma solução da BANK obtida pelo algoritmo Bestof2 que contém dois elementos arquiteturais nos quais o padrão *Strategy* está aplicado. A interface *ICashReader* representa a interface *Strategy* e a classe *CashReader* representa a família de algoritmos que implementa a interface *Strategy*. No entanto, a família de algoritmos não tem pelo menos dois membros, a classe *ATMControl* não tem relação de dependência com a interface e o auto-relacionamento de dependência na interface *ICashReader* representa inconsistências para aplicação deste padrão.

**Figura 6.11:** Padrão *Strategy* aplicado na solução da BANK obtida pelo Bestof2.



## Respondendo as Questões de Pesquisa

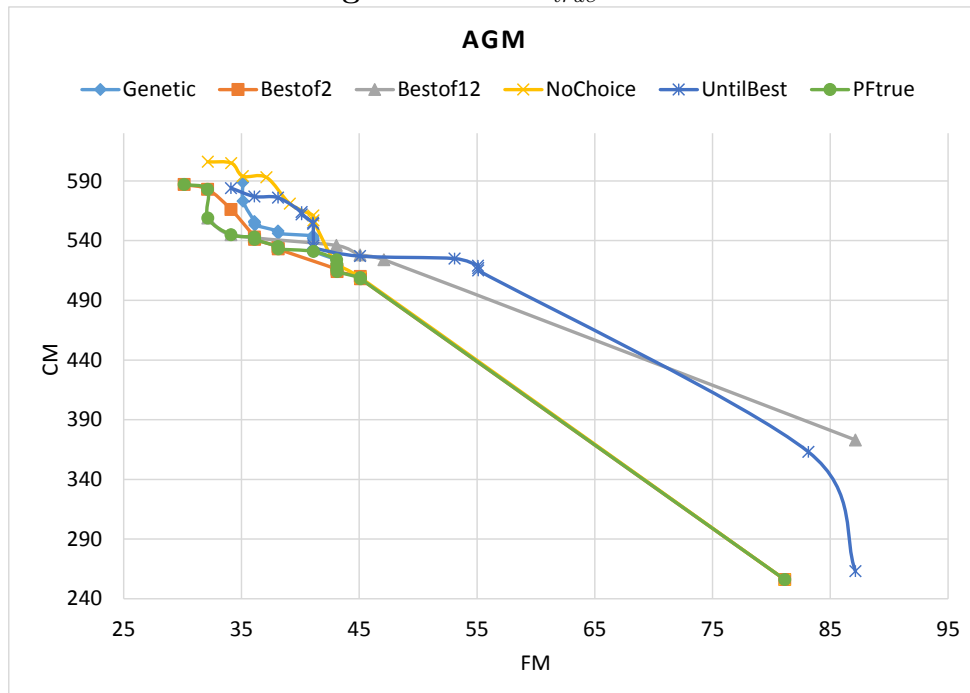
Respondendo a primeira questão de pesquisa: **QP1:** *O algoritmo memético encontra soluções de melhor qualidade do que as soluções obtidas pelo algoritmo genético no contexto de otimização de projeto de PLA usando a MOA4PLA?*

Não houve diminuição significativa nos *fitness* das soluções da abordagem memética, no entanto, há uma ligeira melhoria. No Exp 01 - Etapa 01 as PLAs BANK e BET apresentaram diferença estatística para todos os indicadores de qualidade, entretanto, para a PLA MOM não houve diferença estatística em nenhum indicador. A PLA AGM

obteve diferença estatística somente no indicador de ED. Os resultados de Exp 01 - Etapa 01 mostram que a abordagem memética alcançou soluções de *fitness* ligeiramente melhores do que a abordagem genética, mas no contexto deste estudo a abordagem memética não é estatisticamente melhor do que genética em todos os casos. Um ponto observado na abordagem memética é que ela pode encontrar um maior número de soluções em  $PF_{true}$ , o que sugere que ele explora o espaço de soluções mais extensivamente do que o AG.

Isso pode ser visto nas Figuras 6.12 e 6.13 que mostram o  $PF_{true}$  e  $PF_{known}$  de cada algoritmo para AGM e BANK. É possível ver que as  $PF_{known}$ s são semelhantes, mas a  $PF$  alcançada pelo algoritmo Bestof2 está mais próxima da  $PF_{true}$ , confirmando os resultados dos indicadores de qualidade.

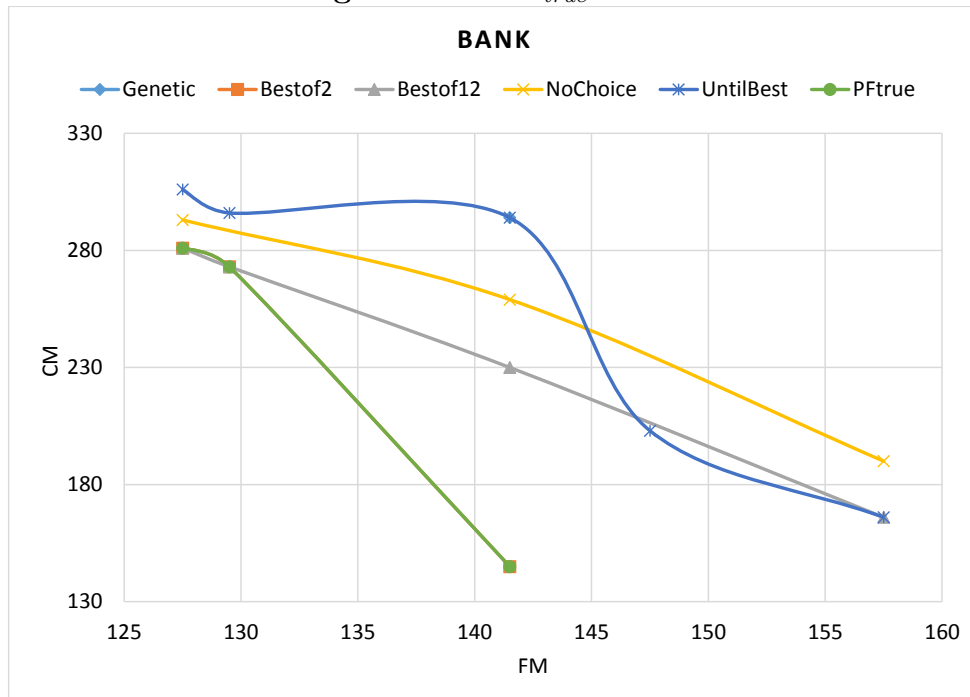
**Figura 6.12:**  $PF_{true}$  AGM.



Em todas as soluções analisadas no Exp 01 - Etapa 03 foi possível identificar que os padrões de projeto aplicados estão incompletos. Os testes de validação da implementação do operador no contexto da busca local (Exp 01 - Etapa 02) não detectaram tal *bug* provavelmente porque os testes foram executados com um menor número de gerações (10000). Nossa hipótese é que as inconsistências são causadas por modificações feitas pelos operadores fundamentais (da busca global) que corrompem o padrão de projeto ao longo das gerações, isto, porque eles não foram preparados para não alterar regiões que contém padrão de projeto.



Figura 6.13:  $PF_{true}$  BANK.



Com os resultados apresentados no Exp 01 - Etapa 02 entende-se que durante as gerações existem indivíduos que têm algum escopo de aplicação de padrão, em cerca de 30%. No entanto, os operadores de busca global causam mudanças na aplicação de padrões de projeto, criando danos à estrutura do padrão (produzindo um padrão de projeto incompleto). Isso enfraquece os valores de *fitness* das funções objetivo levando a eliminação da solução durante o processo de evolução.

Existem indícios de que o projeto de PLA fornecido como entrada pode influenciar no desempenho da busca local, em razão de que, quanto maior número de regiões adequadas à aplicação de padrões de projeto, maior a chance da busca local otimizar a PLA.

Com base nessa evidência, a hipótese de que o algoritmo memético é capaz de encontrar soluções de melhor qualidade do que o algoritmo genético no contexto do design de PLA baseado em pesquisa pode ser mantida. Para investigar esta hipótese, foi necessário refinar os operadores de busca global e repetir os experimentos.

Respondendo a segunda questão de pesquisa **QP2**: *Qual abordagem de busca local é mais eficaz no referido contexto?*

Para responder a esta questão, foi realizada uma análise dos resultados encontrados nos indicadores de qualidade de Exp 01 - Etapa 01. Esta análise contou quantas vezes cada versão de algoritmo teve o melhor desempenho nos cinco indicadores de qualidade. A versão Bestof2 apresentou melhores resultados (com 12 melhores performances), a versão

Bestof12 (com 6 melhores performances) veio em segundo lugar e os outros algoritmos têm o melhor melhor no máximo 1 vez. Assim, a busca local usando a abordagem First Improvement, que mantém na próxima geração a melhor solução entre as soluções alcançadas pela busca global e local foi mais efetiva no contexto da otimização de projeto PLA.

Após responder as duas questões de pesquisa e levantar algumas evidências, para dar continuidade a pesquisa, o próximo passo realizado foi a implementação de restrições nos operadores fundamentais para inibir a alteração de classes e interfaces com aplicação de padrões de projeto e, a repetição do Exp 01 - Etapa 01 para uma nova análise. A próxima seção inicia a discussão do Exp 02.

## 6.2 Estudo Experimental - Exp 02

Para o desenvolvimento deste estudo experimental foram desenvolvidas as restrições nos operadores de mutação fundamentais (busca global) para impedi-los de modificar regiões com aplicação de padrões de projeto.

Dentro do projeto OPLA-Core existe uma classe denominada *PLAFeatureMutation*, ela se encontra no seguinte caminho *jmetal.operators.mutation.PLAFeatureMutation*. As restrições foram desenvolvidas nessa classe. De forma simplificada a restrição verificava se a classe ou interface selecionada para mutação continha um padrão de projeto aplicado, em caso positivo, o processo de mutação era encerrado. Essa restrição foi imposta em todos os seis operadores de mutação fundamentais (*Move Method*, *Move Attribute*, *Add Class*, *Move Operation*, *Add Manager Package* e *Feature-Driven-Operator*). Desse modo, os operadores de mutação fundamentais só poderiam modificar regiões do projeto de PLA em que não existisse algum padrão de projeto aplicado.

Como foram realizadas essas modificações no OPLA-Core, foi necessário repetir o experimento Exp 01 - Etapa 01 <sup>2</sup>.

### 6.2.1 Projeto do Estudo

A questão de pesquisa QP1 foi novamente utilizada: **QP1:** *O algoritmo memético encontra soluções de melhor qualidade do que as soluções obtidas pelo algoritmo genético no contexto de otimização de projeto de PLA usando a MOA4PLA?* e, uma nova questão

<sup>2</sup>O pacote experimental que engloba os projetos de PLA, os resultados dos indicadores de qualidade, dados estatísticos e as soluções obtidas estão disponíveis em <https://github.com/Oplamemetic/experimentalpackage2/>

foi levantada **QP3**: *Qual o par de funções objetivo é mais apropriado para avaliar as mudanças realizadas pela busca local nas soluções de projeto de PLA?*

Para responder as duas perguntas o Exp 02 foi dividido em três etapas: (i) na primeira etapa foi realizado um experimento semelhante ao experimento Exp 01 - Etapa 01, utilizando as funções objetivo CM e FM; (ii) na segunda etapa foi realizado um experimento semelhante ao experimento Exp 02 - Etapa 01, utilizando as funções objetivo COE e ACLASS; (iii) por fim, na terceira etapa foi realizada uma análise qualitativa de algumas soluções encontradas nos experimentos Exp 02 - Etapa 01 e Etapa 02.

### **Exp 02 - Etapa 01**

A primeira etapa do Exp 02 (Exp 02 - Etapa 01) teve o mesmo objetivo do Exp 01 - Etapa 01, ou seja, testar a desempenho das abordagens memética e genética com a seguinte hipótese: a abordagem memética tem melhor desempenho do que a abordagem genética com relação a ER, HV, ED e IGD.

Dessa forma, este experimento utilizou as mesmas configurações e forma de análise do Exp 01 - Etapa 01 com as seguintes diferenças: foram realizadas 30 execuções do AM e foi utilizado somente a versão Bestof2.

Logo o Exp 02 - Etapa 01 utilizou a ferramenta OPLA-Tool com as seguintes configurações: 30 execuções do algoritmo NSGA-II; todos os operadores de mutação fundamentais com probabilidade de mutação de 0.9; sem a utilização do Crossover; tamanho da população igual a 100 e o número de avaliações de *fitness* igual a 30000. O número de avaliações de *fitness* foi utilizado como critério de parada. As funções objetivo utilizadas foram CM e FM (Seção 2.3.1). O experimento foi repetido para o AG e AM na versão Bestof2. A variável independente é um projeto de PLA e as variáveis dependentes são os indicadores de qualidade ER, HV, ED, GD e IGD.

Cada algoritmo (Genético e Bestof2) foi executado 30 rodadas para cada variável independente. Assim, a união de 30 amostras (de soluções resultantes) compõe um conjunto de amostras. Como quatro PLAs (AGM, BANK, BET e MOM) foram utilizadas como variáveis independentes, existe um conjunto de amostras para cada PLA, resultando em 4 conjuntos de amostrais por algoritmo. Nesses conjuntos amostrais que são aplicados os indicadores de qualidade.

## Exp 02 - Etapa 02

A segunda parte do experimento Exp 02 (Exp 02 - Etapa 02) foi executada com mesmas as configurações do Exp 02 - Etapa 01, com a diferença de que as funções objetivo utilizadas foram: COE e ACLASS (Subseção 2.3.1).

Segundo Guizzo (2014) o operador *Design Pattern Mutation Operator* influencia de forma mais acentuada as métricas de coesão relacional e acoplamento. Diante disso, foram selecionadas as funções objetivo COE e ACLASS (refinadas de CM).

O objetivo desse experimento foi averiguar se as funções objetivo COE e ACLASS favorecem a busca local.

## Exp 02 - Etapa 03

A terceira etapa do Exp 02 (Exp 02 - Etapa 03) teve como objetivo, avaliar qualitativamente as soluções encontradas pelo AM Bestof2 nas duas primeiras etapas do Exp 02. As soluções consideradas para este estudo foram as que apresentaram algum padrão aplicado ao projeto de PLA ou as soluções com menor ED em relação as outras soluções.

## 6.2.2 Resultados

### Exp 02 - Etapa 01

Nos 4 conjuntos amostrais encontrados pelo Exp 04 foram aplicados o teste de normalidade de *Shapiro Wilk* além dos indicadores de qualidade: ER, HV, ED e IGD. Não foi possível aplicar o teste de normalidade para a amostra do AG utilizando a PLA BANK pois a mesma solução foi encontrada em todas as 30 rodadas. Todos os conjuntos amostrais apresentaram comportamento não normal com exceção AG com a PLA AGM como pode ser visto na Tabela 6.11. Por esse motivo e, por se comparar somente dois algoritmos, no Exp 04 foi utilizado o teste de *Wilcoxon* (Seção 5.1.2).

A Tabela 6.12 apresenta a quantidade de soluções que compõem o  $PF_{true}$  do Exp 02 - Etapa 01 bem como a cardinalidade do  $PF_{know}$  obtido pelos algoritmos, além, do ER. A segunda coluna da tabela apresenta a cardinalidade de  $PF_{true}$ . A terceira e quinta colunas apresentam a cardinalidade de  $PF_{known}$ . As colunas denominadas ER indicam a proporção de soluções de  $PF_{known}$  que não são membros de  $PF_{true}$ . Os menores valores de ER estão destacados em negrito. O algoritmo Bestof2 encontrou um maior número de soluções que estão presentes no  $PF_{true}$  do que o AG, para as PLAs AGM, BANK e BET.

Considerando BANK, o AG encontrou soluções com os mesmos valores de *fitness* em cada rodada independentemente. Tal solução tem *fitness* (141,5; 294,0) no formato (CM,

Tabela 6.11: Exp 02 - Etapa 01 - Teste de Normalidade.

Algoritmo	Variável Independente	Variáveis Dependentes			
		CM		FM	
		<i>Shapiro Wilk</i>			
		<i>p-value</i>	Condição	<i>p-value</i>	Condição
<b>Bestof2</b>	AGM	<2,2e-16	Não Normal	1,088e-11	Não Normal
	BANK	7,791e-09	Não Normal	2,373e-12	Não Normal
	MOM	4,138e-08	Não Normal	4,58e-13	Não Normal
	BET	<2,2e-16	Não Normal	<2,2e-16	Não Normal
<b>Genético</b>	AGM	<2,2e-16	Não Normal	0,3087	Normal
	BANK	—	Não Aplicado	—	Não Aplicado
	MOM	7,419e-06	Não Normal	3,879e-14	Não Normal
	BET	3,045e-12	Não Normal	1,685e-09	Não Normal

Tabela 6.12: Exp 02 - Etapa 01 - Resultados do ER.

PLA	#PF <sub>true</sub>	Bestof2		Genético	
		#PF <sub>known</sub>	ER	#PF <sub>known</sub>	ER
AGM	11	10	<b>0,37</b>	12	0,55
BANK	3	3	<b>0</b>	1	1
MOM	5	7	0,8	4	<b>0</b>
BET	7	4	<b>0</b>	11	0,57

FM). No entanto, esta não é a solução ótima porque o algoritmo Bestof2 encontrou uma solução com melhor *fitness*: (129,5; 284,0).

A Tabela 6.13 apresenta os valores médios de HV. As maiores médias estão destacadas em negrito. O algoritmo Bestof2 superou o AG para as PLAs AGM, BANK e BET, enquanto o AG obteve a melhor média para MOM. Além disso, pode-se afirmar com 95% de confiança que o algoritmo Bestof2 é significativamente melhor do que o AG para a PLA BANK (quarta coluna da Tabela 6.13).

Tabela 6.13: Exp 02 - Etapa 01 - Resultados do HV.

PLA	Bestof2	Genético	<i>p-value</i>
AGM	<b>0,2138182</b>	0,213566	0,8883
BANK	<b>0,077089</b>	0,005433	<b>6,522e-13</b>
MOM	0,359850	<b>0,424334</b>	0,1872
BET	<b>0,291085</b>	0,264803	0,2643

Os médias dos valores do indicador IGD são apresentados na Tabela 6.14. Os valores mais baixos de IGD estão destacados. O algoritmo Bestof2 obteve os melhores resultados

para IGD. No entanto, há diferença significativa apenas para PLA BANK como pode ser vista na última coluna da Tabela 6.14 com  $p\text{-value} < 0,05$ .

**Tabela 6.14:** Exp 02 - Etapa 01 - Resultados do IGD.

PLA	Bestof2	Genético	$p\text{-value}$
AGM	<b>27,43842</b>	27,95924	0,5894
BANK	<b>41,01332</b>	48,42061	<b>6,522e-13</b>
MOM	<b>6,631277</b>	9,20852	0,5229
BET	<b>83,39019</b>	96,91239	0,1494

A ED de cada solução do conjunto amostral é calculada tendo em conta a solução ideal. No Exp 04 o *fitness* das soluções ideais, bem como, o *fitness* do projeto original de cada PLA, são apresentados na Tabela 6.15.

**Tabela 6.15:** Exp 02 - Etapa 01 - *Fitness* Original e *Fitness* Ideal.

PLA	Exp 04	
	<i>Fitness</i> Original	<i>Fitness</i> Ideal
AGM	(84,14; 758,0)	(34,11; 263,0)
BANK	(157,0; 326,0)	(127,5; 166,0)
MOM	(72,14; 1122,0)	(45,1; 801,0)
BET	(461,02; 1486,0)	(275,01; 616,0)

A segunda e quinta colunas da Tabela 6.16 apresentam a menor ED encontrado por cada algoritmo. A menor ED de cada PLA está destacada em negrito. A terceira e a sexta colunas contêm os *fitness* da solução de menor ED. A porcentagem de melhoria no valor de cada função objetivo, em relação ao *fitness* original, também é apresentada. O AM encontrou valores mais baixos de ED para AGM, BANK e BET.

**Tabela 6.16:** Exp 02 - Etapa 01 - Soluções com menor ED.

PLA	Bestof2			Genético		
	ED	(CM; FM)	Porcentagem	ED	(CM; FM)	Porcentagem
AGM	<b>55,03</b>	(89,14; 263,0)	(-5,94%; 65,3%)	250,71	(53,09; 513,0)	(36,9%; 32,32%)
BANK	<b>30,0</b>	(157,5; 166,0)	(-0,31%; 49,07%)	128,76	(141,5; 294,0)	(9,87%; 9,81%)
MOM	16,98	(62,08; 801,0)	(13,94%; 28,6%)	<b>4,98</b>	(49,08; 804,0)	(31,96%; 28,34%)
BET	<b>186,0</b>	(461,02; 616,0)	(0%; 58,54%)	596,21	(313,02; 1211,0)	(32,10%; 18,5%)

Levando em consideração a adequação da solução com a menor ED (Tabela 6.16) contra o *fitness* original de todos os projetos de PLA, observa-se que o algoritmo Bestof2 otimizou na PLA MOM os valores de CM em 13% e o AG otimizou CM em cerca de 30%. Para AGM e BANK, o valor de FM foi melhorado pelo algoritmo Bestof2 mas

comprometeu CM. Além disso, o Bestof2 obteve maiores percentagens de melhoria em FM para todas as PLAs se comparado ao AG.

Tendo em conta os resultados dos indicadores de qualidade, podemos afirmar que o algoritmo Bestof2 foi o melhor para as PLAs BANK e BET. Por outro lado, o AG obteve os melhores resultados para a PLA MOM.

### Exp 02 - Etapa 02

Nos 4 conjuntos amostrais encontrados pelo Exp 02 - Etapa 02 foram aplicados o teste de normalidade de *Shapiro Wilk* além dos indicadores de qualidade: ER, HV, ED e IGD. Todos os conjuntos amostrais apresentaram comportamento não normal, com  $p\text{-value} < 0,05$ , como pode ser visto na Tabela 6.11. Por esse motivo, e por se comparar somente dois algoritmos, no Exp 05 foi utilizado o teste estatístico de *Wilcoxon* (Seção 5.1.2).

**Tabela 6.17:** Exp 02 - Etapa 02 - Teste de Normalidade.

Algoritmo	Variável Independente	Variáveis Dependentes			
		COE		ACLASS	
		<i>Shapiro Wilk</i>			
		$p\text{-value}$	Condição	$p\text{-value}$	Condição
<b>Bestof2</b>	<b>AGM</b>	2,849e-07	Não Normal	8,266e-09	Não Normal
	<b>BANK</b>	1,578e-07	Não Normal	1,565e-05	Não Normal
	<b>MOM</b>	8,673e-10	Não Normal	2,632e-09	Não Normal
	<b>BET</b>	<2,2e-16	Não Normal	<2,2e-16	Não Normal
<b>Genético</b>	<b>AGM</b>	5,096e-09	Não Normal	7,807e-08	Não Normal
	<b>BANK</b>	4,261e-05	Não Normal	9,944e-07	Não Normal
	<b>MOM</b>	9,605e-10	Não Normal	6,537e-10	Não Normal
	<b>BET</b>	<2,2e-16	Não Normal	<2,2e-16	Não Normal

A cardinalidade da  $PF_{true}$  e  $PF_{known}$  dos algoritmos Bestof2 e Genético são apresentadas na Tabela 6.18. A quarta e sexta colunas mostram o valor ER onde os menores valores estão realçados em negrito. O algoritmo Bestof2 encontrou maior número de soluções que estão presentes na  $PF_{true}$  para as PLAs AGM e BET. Para BANK existe apenas uma solução em  $PF_{true}$ , que foi encontrada por ambos os algoritmos. O AG obteve o melhor desempenho para MOM em relação a ER.

A Tabela 6.19 contém os valores médios de HV para o Exp 02 - Etapa 02. Para cada PLA, a maior média é destacada em negrito. O algoritmo Bestof2 superou o AG para as PLAs BANK e BET. Por outro lado, o AG obteve melhores médias para AGM e MOM. De acordo com o teste estatístico de *Wilcoxon*, o Bestof2 é melhor que o AG para BANK, como está mostrado em negrito na última coluna da Tabela 6.19.

**Tabela 6.18:** Exp 02 - Etapa 02 - Resultados do ER.

PLA	#PF <sub>true</sub>	Bestof2		Genético	
		#PF <sub>known</sub>	ER	#PF <sub>known</sub>	ER
AGM	7	5	<b>0,2</b>	5	<b>0,2</b>
BANK	1	1	<b>0</b>	1	<b>0</b>
MOM	1	2	0,5	1	<b>0</b>
BET	19	20	<b>0,15</b>	18	0,5

**Tabela 6.19:** Exp 02 - Etapa 02 - Resultados do HV.

PLA	Bestof2	Genético	<i>p-value</i>
AGM	0,2154935	<b>0,2179287</b>	0,7403
BANK	<b>0,463322</b>	0,236488	<b>0,001047</b>
MOM	0,339211	<b>0,366989</b>	0,4757
BET	<b>0,517609</b>	0,515095	0,153

A Tabela 6.20 apresenta as médias de IGD para o Exp 02 - Etapa 02. Os valores mais baixos de IGD são destacados em negrito. O algoritmo Bestof2 obteve os melhores resultados para as PLAs AGM, BANK e BET, com diferença estatística atestada para AGM e BANK. Para este indicador de qualidade o AG foi ligeiramente melhor do que o Bestof2 para MOM.

**Tabela 6.20:** Exp 02 - Etapa 02 - Resultados do IGD.

PLA	Bestof2	Genético	<i>p-value</i>
AGM	<b>3,568129</b>	3,950564	<b>2,428e-11</b>
BANK	<b>7,152134</b>	17,4355	<b>0,0006751</b>
MOM	2,226487	<b>2,126487</b>	0,4757
BET	<b>1,646808</b>	1,686444	0,4114

Como no Exp 01 e nesse experimento o valor de ED de cada solução do conjunto amostral é calculado tendo em conta a solução ideal. No Exp 02 - Etapa 02 o *fitness* das soluções ideais, bem como o *fitness* do projeto original da PLA são apresentados na Tabela 6.21.

**Tabela 6.21:** Exp 02 - Etapa 02 - *Fitness* Original e *Fitness* Ideal.

PLA	Exp 02 - Etapa 02	
	<i>Fitness</i> Original	<i>Fitness</i> Ideal
AGM	(26,0; 30,0)	(9,0; 1,0)
BANK	(16,0; 133,0)	(91,0; 13,0)
MOM	(28,0; 14,0)	(8,0; 28,0)
BET	(100,0; 122,0)	(113,0; 75,0)



Os menores valores de EDs das respectivas soluções encontradas pelos algoritmos Bestof2 e Genético estão apresentados na Tabela 6.22. Para cada PLA, o menor ED está destacado em negrito. A porcentagem de melhoria no valor de cada função objetivo em relação ao *fitness* original são apresentadas nas colunas quatro e sete. O algoritmo Bestof2 encontrou valores mais baixos de ED para as PLAs AGM e BET. Já o AG alcançou soluções com o menor ED para as PLAs BANK e MOM.

**Tabela 6.22:** Exp 02 - Etapa 02 - Soluções com menor ED.

PLA	Bestof2			Genético		
	ED (COE; ACLASS)	Porcentagem	ED (COE; ACLASS)	ED (COE; ACLASS)	Porcentagem	
<b>AGM</b>	<b>21,54</b>	(29,0; 9,0)	(-11,53%; 70,0%)	27,29	(13,0; 28,0)	(50,0%; 6,66%)
<b>BANK</b>	62,12	(29,0; 9,0)	(-81,25%; 93,23%)	<b>59,22</b>	(33,0; 25,0)	(-106,25%; 81,20%)
<b>MOM</b>	1,0	(8,0; 29,0)	(71,42%; -107,14%)	<b>0</b>	(8,0; 28,0)	(71,42%; -100,0%)
<b>BET</b>	<b>22,47</b>	(121,0; 96,0)	(-21,0%; 21,31%)	23,41	(121,0; 97,0)	(-21,0%; 20,49%)

No que diz respeito ao percentual de melhoria do *fitness* original, é possível notar que em quase todas as PLAs um objetivo comprometeu o outro, comportamento esse já esperando, visto que as funções objetivo são concorrentes. Independentemente de ser positiva ou negativa as porcentagens de alteração nos valores de *fitness* são maiores no Exp 02 - Etapa 2 do que na Etapa 01. Este fato, traz evidências de que as funções objetivo COE e ACLASS são mais sensíveis às mudanças realizadas pelo AM do que a função objetivo CM.

### Exp 02 - Etapa 03

Uma análise qualitativa das soluções obtidas nas Etapas 01 e 02 do Exp 02 foi realizada, para verificar como os padrões de projeto foram aplicados nos projetos de PLA durante o processo de busca e, compreender as diferenças apontadas pelos indicadores de qualidade.

A análise permitiu identificar que os padrões de projeto foram corretamente aplicados nas soluções. Para fins de ilustração, a Figura 6.14 apresenta um excerto de uma solução retornada pelo algoritmo Bestof2 para BET. A figura descreve a aplicação do padrão *Mediator* envolvendo sete elementos arquiteturais (classes e interfaces). A classe *EventOfInterest* representa a classe *Mediator* e os outros elementos arquiteturais do pacote *CartaoMgr* formam a família de algoritmos, onde todos os elementos estão associadas com a característica *AcBasico*.

Outro exemplo que caracteriza a correta aplicação dos padrões é apresentada na Figura 6.15, a qual contém elementos arquiteturais com a aplicação do *Strategy* em uma solução obtida para a MOM. A família de algoritmos é composta por *Album* e *IAlbumMgt*. A classe *AlbumCtrl* depende da interface *IAlbumMgt*, caracterizando o padrão.

Figura 6.14: Aplicação do padrão *Mediator* na solução BET do Exp 02 - Etapa 01.

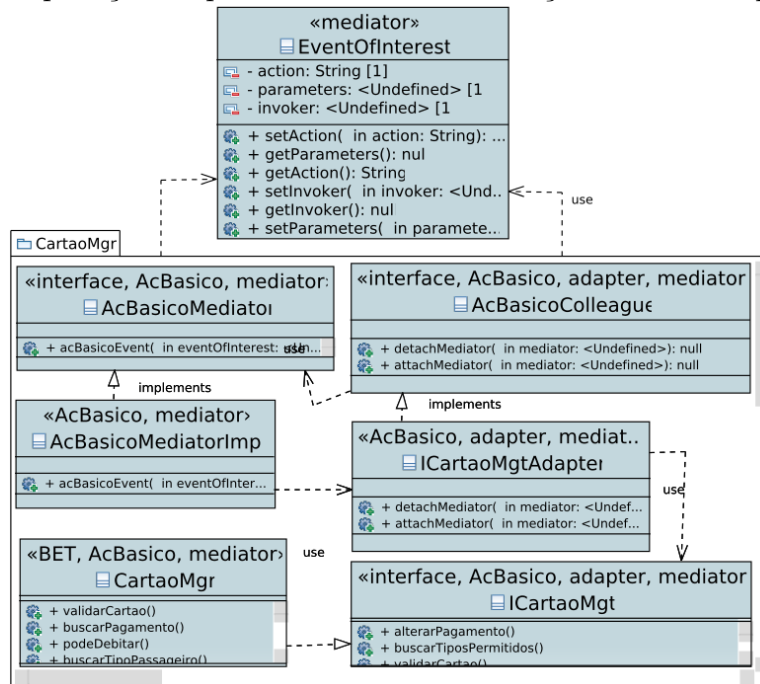
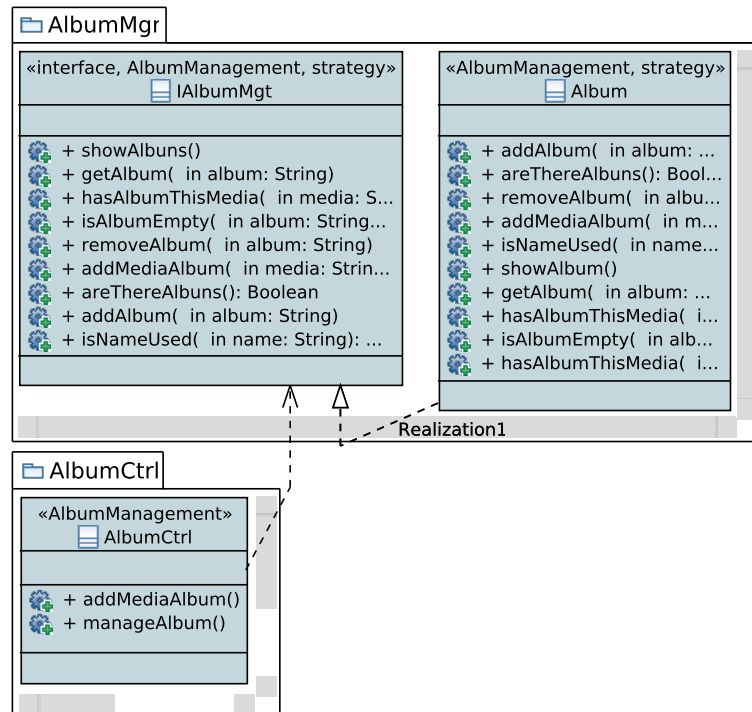


Figura 6.15: Aplicação do padrão *Strategy* na solução MOM do Exp 02 - Etapa 01.



Durante a análise das soluções obtidas para responder por que o algoritmo genético tem melhor desempenho para a PLA MOM, percebe-se que a modularização de características

tem impactado os resultados. As PLAs AGM e MOM têm menor modularização de características do que BANK e BET. Por exemplo, como pode ser visto na Figura 6.15, a característica *AlbumManagement* está espalhada em dois pacotes: *AlbumCtrl* e *AlbumMgr*. Por outro lado, na PLA BANK cada característica está associada a apenas um pacote. Na PLA BET as características também estão modularizados, por exemplo, a característica *AcBasico* está modularizada no pacote *CartaoMgt* (Figura 6.14).

A Tabela 6.23 contém a taxa de aplicação de padrões obtida das soluções pertencentes ao conjunto de amostras, que contém todas as soluções obtidas nas 30  $PF_{approx}$  de cada PLA. As colunas identificadas como  $\#S$  mostram o número total de soluções, as colunas identificadas como  $\#A$  e  $\#NA$  mostram o número de soluções que contêm algum padrão de projeto e que não contêm padrões de projeto, respectivamente. Finalmente, as colunas identificadas como *TaxaAplic.* mostram a taxa de aplicação de padrões.

O operador de busca local do algoritmo memético é favorecido pelos projetos de PLA com melhor modularização de característica (Guizzo *et al.*, 2014), como acontece com a PLA BANK e BET. A taxa de aplicação do padrão apresentada na Tabela 6.23 corrobora esta afirmação.

**Tabela 6.23:** Exp 02 - Etapa 03 - Taxa de aplicação de padrões de projeto no Conjunto de Amostras do Exp 02 - Etapa 01 e 02.

PLA	Exp 04				Exp 05			
	#S	#A	#NA	Taxa Aplic.	#S	#A	#NA	Taxa Aplic.
<b>AGM</b>	97	0	97	<b>0</b>	184	3	181	<b>0,02</b>
<b>BANK</b>	51	18	33	<b>0,35</b>	32	32	0	<b>1</b>
<b>MOM</b>	95	6	89	<b>0,06</b>	63	0	63	<b>0</b>
<b>BET</b>	144	82	62	<b>0,56</b>	473	114	359	<b>0,24</b>

De 24 a 100% das soluções encontradas para BANK e BET contêm algum padrão de projeto, enquanto que para AGM e MOM a taxa é inferior a 10%. Assim, os projetos com melhor modularização de características (BANK e BET) favorecem a aplicação do padrão, provavelmente porque a modularização beneficia os escopos adequados necessários ao operador de busca local. Também observou-se que nas PLAs originais que as características de AGM e MOM são mais dispersas e entrelaçadas do que as características da BANK e BET. Isto significa que o projeto de PLA fornecido como entrada para a otimização impacta na taxa de aplicação de padrões.

## Respondendo as Questões de Pesquisa

Respondendo a primeira questão de pesquisa, **QP1:** *O algoritmo memético encontra soluções de melhor qualidade do que as soluções obtidas pelo algoritmo genético no contexto de otimização de projeto de PLA usando a MOA4PLA?*

Duas evidências foram encontradas: (i) o desempenho do AM varia de acordo com o projeto de PLA fornecido como entrada. Assim, uma PLA alta modularização de características e baixo entrelaçamento de características, beneficiam o desempenho do AM; (ii) o indicador de ED apontou que o AM encontrou soluções com melhor *trade-off* do que o AG. Contudo, de acordo com os indicadores HV e IGD, a melhora é significativa somente para a PLA BANK.

Respondendo a segunda questão de pesquisa **QP3:** *Qual o par de funções objetivo é mais apropriado para avaliar as mudanças realizadas pela busca local nas soluções de projeto de PLA?*

As funções objetivo utilizadas no Exp 02 - Etapa 02 estão relacionadas aos princípios de projeto afetados pelas mudanças realizadas pelo operador de busca local: coesão (COE) e acoplamento (ACCLASS). As funções usadas no Exp 02 - Etapa 01 são mais gerais. A porcentagem de melhoria apontou que as funções de Exp 02 - Etapa 01 (COE e ACLASS) são mais sensíveis às mudanças causadas pelo AM. No entanto, esse par não inclui a avaliação do grau de modularização de características, o que é medido pela função FM usada em Exp 02 - Etapa 01. À medida que a modularização de características impacta no desempenho do AM e, MOA4PLA têm operadores de busca para melhorar a modularização de características, inferimos que a otimização de COE, ACLASS e FM levariam à resultados ainda melhores. Esta investigação será conduzida em experimentos futuros.

## 6.3 Estudo Qualitativo - Exp 03

### 6.3.1 Projeto do Estudo

Por fim, foi conduzido um estudo qualitativo (Exp 03) que contou com o auxílio de arquitetos de software. Neste texto, os arquitetos são denominados avaliadores. Este experimento utilizou algumas soluções encontradas pelo algoritmo Bestof2, nas duas primeiras etapas do Exp 02.

O objetivo do Exp 03 foi avaliar algumas soluções encontradas pela OPLA-Tool com o AM para responder a seguinte questão de pesquisa: **QP4:** *As soluções encontradas pela OPLA-Tool com o AM são consideradas boas, do ponto de vista de arquitetos de software?*

Foram selecionados os projetos das PLAs AGM, BANK e MOM, uma vez que, seus portes permitiam uma avaliação visual como um todo em uma página, diferentemente da BET. Do Exp 02 - Etapa 01 os projetos utilizados foram os que apresentaram menor valor de CM, menor valor de FM, menor valor de ED e os que continham aplicação de padrões. Da mesma forma, os projetos utilizados do Exp 02 - Etapa 02 foram os que apresentaram menor valor de COE, menor valor de ACLASS, menor valor de ED e os que continham aplicação de padrões.

A amostra foi composta por 18 avaliadores do âmbito profissional e acadêmico (mestrandos, mestres, doutorandos e doutores) divididos em dois grupos, um avaliando as soluções encontradas pela otimização das funções CM e FM e, o outro, avaliando as soluções encontradas pela otimização das funções COE e ACLASS. A distribuição dos participantes foi feita de forma aleatória resultando em seis participantes para cada LPS, como pode ser visto na Tabela 6.24.

**Tabela 6.24:** Exp 03 - Distribuição dos avaliadores.

Avaliadores	PLA	Funções Objetivo	Avaliadores	PLA	Funções Objetivo
n_01	AGM	CM, FM	n_10	AGM	COE, ACLASS
n_02	AGM	CM, FM	n_11	AGM	COE, ACLASS
n_03	AGM	CM, FM	n_12	AGM	COE, ACLASS
n_04	BANK	CM, FM	n_13	BANK	COE, ACLASS
n_05	BANK	CM, FM	n_14	BANK	COE, ACLASS
n_06	BANK	CM, FM	n_15	BANK	COE, ACLASS
n_07	MOM	CM, FM	n_16	MOM	COE, ACLASS
n_08	MOM	CM, FM	n_17	MOM	COE, ACLASS
n_09	MOM	CM, FM	n_18	MOM	COE, ACLASS

Para a execução do estudo foi necessário realizar um treinamento com os participantes. O treinamento teve como objetivo instruir sobre os conceitos de LPS, SMarty (uma abordagem para gerenciamento de variabilidades em projetos de LPS) (OliveiraJr *et al.*, 2010) além das funções objetivo utilizadas. O início do treinamento utilizou um vídeo desenvolvido no trabalho de (Conti, 2016). Os documentos utilizados se resumem em: Termo de adesão ao Estudo qualitativo, Questionário de caracterização de Participante, Conceitos de LPS e SMarty, Descrição da LPS (utilizada), Conjunto de Características da LPS (utilizada) e, por fim, o Questionário Principal. Os documentos utilizados são apresentados no Apêndice B.

Todas as documentações oferecidas tem o objetivo de auxiliar o avaliador a responder as perguntas apresentadas no Questionário Principal. Os participantes que avaliaram os projetos otimizados pelas funções objetivo CM e FM responderam o Questionário 1 com as seguintes perguntas:

- 1 Qual a sua opinião com relação à coesão relacional desse projeto de PLA?
- 2 Qual a sua opinião com relação ao acoplamento desse projeto de PLA?
- 3 Qual a sua opinião com relação ao tamanho das interfaces desse projeto de PLA?
- 4 Qual a sua opinião com relação à modularização de características desse projeto de PLA?
- 5 Qual a melhor alternativa de projeto sob o ponto de vista de modularização de características?
- 6 Quais propriedades você julga importantes para se avaliar em um projeto de PLA?

Os participantes que avaliaram os projetos otimizados pelas funções objetivo COE e ACLASS responderam o Questionário 2 com as seguintes perguntas:

- 1 Qual a sua opinião com relação à coesão relacional desse projeto de PLA?
- 2 Qual a sua opinião com relação ao acoplamento entre classes desse projeto de PLA?
- 3 Qual a melhor alternativa de projeto sob o ponto de vista de coesão relacional?
- 4 Qual a melhor alternativa de projeto sob o ponto de vista de acoplamento de classes?
- 5 Qual a melhor alternativa de projeto sob o ponto de vista de modularização de características?
- 6 Quais propriedades você julga importantes para se avaliar em um projeto de PLA?

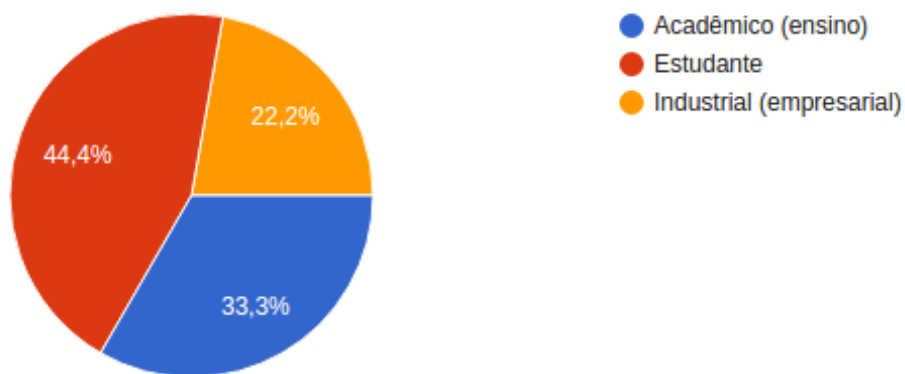
Não foi possível realizar o estudo em uma única etapa devido a disponibilidade dos participantes, por esse motivo, foi utilizada a ferramenta GoogleForms possibilitando a realização do estudo de forma individual e à distância. Todas as respostas foram inseridas em formulários preenchidos pelos próprios participantes. As respostas foram coletadas e tratadas para serem processadas pela ferramenta Iramuteq (Seção 5.2.2).

### 6.3.2 Resultados

Participaram deste experimento 18 pessoas com diferentes características. O experimento foi executado quantas vezes se fez necessário devido a diferença de localidade e a disponibilidade dos participantes <sup>3</sup>.

Os 18 participantes foram divididos em três áreas de atuação: acadêmico (professor), estudante e industrial (empresarial). Como pode ser visto na Figura 6.16 a maioria dos participantes foram estudantes, contudo, 22,2% estão atuando na indústria de software. Cerca de 55% dos participantes são mestrandos e mestres, 28% são doutorandos e doutores e, por fim, 11,1% são graduados.

**Figura 6.16:** Exp 03 - Área de atuação dos participantes.



Todos os participantes tinham experiência com notação UML nos diagramas de casos de uso e de classes se dividindo em 38,9% experiência média, 33,3% experiência avançada e 27,8% experiência básica. Um único participante do setor industrial não relatou experiência com LPS, contudo, 33,3% dos participantes apresentaram experiência avançada, 22,2% experiência moderada e básica e, 16,7% apresentaram experiência superficial.

Com as respostas dos 18 participantes foram construídos três *corpus* textuais para análise (Apêndice C). O primeiro foi composto pelas respostas das questões 1 a 5 do grupo 1 (CM, FM), o segundo *corpus* foi composto pelas respostas das questões 1 a 5 do grupo 2 (COE, ACLASS) e, por fim, o terceiro *corpus* textual foi composto pelas respostas da questão 6 dos grupos 1 e 2. Os grupos podem ser identificados olhando a Tabela 6.24

<sup>3</sup>O pacote experimental que engloba os projetos de PLA, as soluções utilizadas na avaliação e os *corpus* textuais estão disponíveis em <https://github.com/Oplamemetic/experimentalpackage3/>

onde os avaliadores n\_01 a n\_09 compõem o grupo 1 e os avaliadores n\_10 a n\_18 compõem o grupo 2.

### Análise do Corpus 1

Na análise do corpus 1 houve a ocorrência de 1645 palavras sendo 516 formas distintas e, 58 segmentos de texto. Em torno de 60% das palavras apareceram mais de uma vez e foram equiparadas por meio de classificações hierárquicas descendentes de segmentos de texto. Foram identificados 58 segmentos de texto. Um maior grau de semelhança no vocabulário foi encontrado em três classes/categorias. Uma classe/categoria é composta por um agrupamento de palavras que apresentaram um maior grau de semelhança no contexto do *corpus*.

A Tabela 6.25 apresenta o dendograma e as classes/categorias que surgiram com a divisão de conteúdo. Na tabela são apresentadas somente as palavras que apresentaram significância no *corpus*. O Corpus 1 está apresentado no Apêndice C.1.

**Tabela 6.25:** Exp 03 - Dendograma Corpus 1.

<b>Classe 2</b> <b>19,1%</b> <b>Reusabilidade</b>	<b>Classe 1</b> <b>59,6%</b> <b>Propriedades</b> <b>Arquiteturais</b>	<b>Classe 3</b> <b>21,3%</b> <b>Seleção</b> <b>do melhor</b> <b>projeto</b>
maior	projeto	melhor
operação	alto	identificar
parecer	coesão	estereótipo
representar	baixo	modularização
reúso	difusão	plas
precisar	dar	opcional
interface	dependência	conseguir
	tornar	
	apenas	
	além	
	forte	
	pla	
	interface	
	muito	

A Classe 1 apresenta uma incidência de 59,6%. Essa Classe está relacionada às propriedades de projetos arquiteturais de LPS com ênfase nos princípios básicos de



projeto. As palavras que se destacaram foram: alto, coesão, baixo, difusão, projeto, dependência, pla e interface.

De forma geral o termo alto está relacionado a difusão, uma vez que os avaliadores em sua maioria avaliaram as PLAs como difusas e, o termo baixo está relacionado a coesão baseada em características.

*(...) os pacotes `displaypromptsclient1` e `atmclient1` são os que possuem menos modularização de características por causa do elevado número de **difusão** de características e a **coesão** baseada em características, de forma geral, os demais pacotes possuem **alta** modularidade (n\_02).*

Dentre as respostas relacionadas a modularização de características, todos os avaliadores consideraram que as PLAs continham características difusas, com exceção dos avaliadores n\_02, n\_05 e n\_09. Os avaliadores n\_02 e n\_05 avaliaram que as PLAs continham características modulares e, o n\_09 disse que as características não eram difusas ou modulares.

*(...) Considero nem alta nem baixa, pois não são todas as características da PLA contêm alta difusão, interação e coesão (n\_09).*

O termo projeto está relacionado às PLAs. Já o termo dependência se refere as dependências existentes entre as classes e interfaces que influenciam diretamente nas métricas básicas (acoplamento) do projeto. Segundos os avaliadores os projetos de PLA analisados apresentaram baixa coesão e alto acoplamento.

*(...) O **projeto** possui funcionalidades que estão devidamente isoladas em pequenos módulos com poucas responsabilidades. Não são todos, mas algumas funcionalidades estão devidamente isoladas em módulos (n\_05).*

*(...) o acoplamento de toda a **pla** poderia ser melhorado levando em consideração principalmente o pacote `atmclient1` que possui 13 classes com **dependências** heranças e realizações confusas com relação a suas reais responsabilidades isso é consequência de um alto acoplamento suas **interfaces** também estão muito simples e necessitam ser repensadas (n\_01).*

*(...) dado o número de relacionamentos internos de cada pacote e o número de pacotes, entendo que esta pla tem **baixa coesão**, cinco pacotes não possuem relacionamento entre Classes e interfaces ou os relacionamentos são com interfaces de outros pacotes (n\_02).*

*(...) De forma geral, não existe forte **coesão** entre as classes, interfaces ou pacotes. A PLA possui 7 pacotes que poderiam ser simplificados (unidos) com relação as suas interfaces e classes para tornar o projeto mais coeso. (n\_01).*

*(...) O projeto tem um grande volume de ligações tendo como um dos extremos somente um pacote, o que pode indicar que o mesmo pacote possui muitas responsabilidades,*

viola pelo menos um dos princípios do SOLID e indica um **alto acoplamento** e, por consequência, uma **baixa coesão** (n\_05).

(...) A PLA possui um **acoplamento elevado**, principalmente devido a uma forte dependência da classe *MediaMgr* (n\_08).

Dentre as resposta à esse tema, um avaliador, levantou algumas preocupações com a apresentação da arquitetura em relação a nomenclatura de representação utilizada no projeto. (...) *De um modo geral, algumas coisas que não gostei na arquitetura. Vários atributos tem **nomes genéricos**, attr5 por exemplo. O mesmo acontece com o nome dos pacotes. Isso é ruim e não faz sentido quando falamos de Orientação a Objeto. O **nome** das classes, operações, métodos e pacotes devem representar o conceito / função / característica que eles mapeiam ou representam.* (n\_04). (...) *Encontrei algumas coisas que me parece **inconformidades**. Não entendi muito bem porque uma parte da arquitetura é **desconexa** da outra.* (n\_04).

Com 19,15% de incidência a Classe 2 está relacionada a reusabilidade dos projetos analisados além de indicar causas positivas e negativas que interferem diretamente na reusabilidade. As palavras mais expressivas foram: reuso, maior, interface, operação.

O termo reuso ligado a maior é utilizado para expressar o que se espera com um projeto coeso e pouco acoplado, contudo, os termos interface e operação estão em destaque pois representam os elementos que devem ser administrados para se alcançar um projeto reusável.

(...) *considero baixo (o número de interfaces) pois a maioria das **interfaces** contêm entre 1 à 4 **operações** e a **maior** imanagemedia teve 16 operações não **prejudicando** a manutenção e o **reuso** da pla* (n\_09).

(...) *as **interfaces** são pequenas o que pode influenciar positivamente no **reuso** das mesmas e na facilidade de substituições de suas implementações* (n\_05).

Um avaliador questiona a construção e utilização das interfaces, que estão interferindo diretamente na reusabilidade do projeto. (...) *O acoplamento de toda a PLA poderia ser melhorado levando em consideração, principalmente, o pacote *ATMClient1* (PLA BANK) que possui 13 classes com dependências, heranças e realizações confusas com relação a suas reais responsabilidades. Isso é consequência de um alto acoplamento. Suas interfaces também estão muito simples e necessitam ser repensadas* (n\_01).

Por fim a Classe 3 apresenta incidência de 21,38%. Essa classe está relacionada a escolha do melhor projeto, no ponto de vista dos avaliadores, além de dificuldades para realizar a escolha. As palavras mais representativas foram: melhor, identificar, plas, conseguir.

Os termos melhor, identificar, conseguir e plas estão relacionados à escolha da melhor alternativa de projeto de PLA do ponto de vista do avaliador.

(...) *Acredito que a VAR\_All\_banking-8489715735, no entanto não está muito claro para mim (n\_03).*

(...) *MobileMedia. (...) Achei que as responsabilidades ficaram **melhor** divididas em seus pacotes. Fica mais fácil **identificar** onde as responsabilidades estão (n\_07).*

(...) *VAR\_All\_agm-4953622311 eu ficaria entre ela e a AGM. Me incomodou ter um pacote na AGM com muitas classes. Entretanto, as outras arquiteturas criaram pacotes com nomes estranhos, e que não pareciam fazer muito sentido (n\_04).*

(...) *dentre as alternativas analisadas a **melhor** alternativa de projeto é a banking pdf tal escolha foi devido ao fato dessa alternativa definir esteriótipos e notações para as interfaces e classes do projeto opcional e mandatory dentre outros. (n\_01).*

(...) *mobilemedia achei que as responsabilidades ficaram **melhor** divididas em seus pacotes fica mais fácil **identificar** onde as responsabilidades estão (n\_07).*

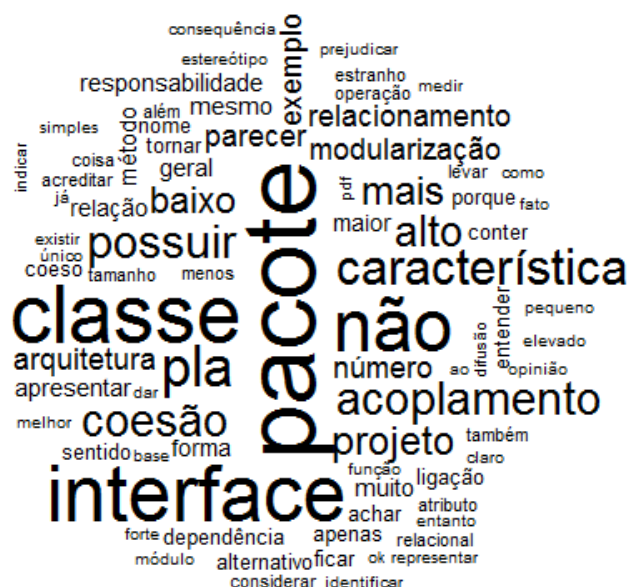
A Figura 6.17 apresenta a análise de similitude produzida pela ferramenta Iramuteq. Neste grafo identificam-se os vértices principais com as palavras: não, interface, classe, pacote e pla. Os vértices *interface*, *classe* e *pacote* estão relacionados aos principais elementos arquiteturais identificados nos projetos de PLA, o vértice *pla* se ramifica para os vértices *possuir*, *coesão*, *baixo*, *projeto* e *alto* retratando a análise das propriedades arquiteturais do projeto fazendo referência a Classe 1 (Tabela 6.25). De forma implícita a Classe 2 (Tabela 6.25) pode ser identificada no vértice *interface* que se ramifica em *pla*, *possuir*, *responsabilidade* e *coesão* caracterizando a preocupação do avaliador em garantir a reusabilidade em um projeto de PLA durante sua análise. Por fim, o vértice *não* se ramifica em *identificar*, *estereótipo* e *modularização* retratando a escolha do melhor projeto referente a Classe 3 (Tabela 6.25).

A Nuvem de Palavras se baseia na frequência das palavras utilizadas no corpus, a Figura 6.18 apresenta a Nuvem de Palavras do Corpus 1. Essa nuvem reforça as conclusões alcançadas com a análise de similitude, nela é possível identificar que os avaliadores discutiram sobre pacotes, classes e interfaces, sobre os níveis de coesão e acoplamento, além da modularização de características. Identificou-se que as palavras mais utilizadas, com suas respectivas frequências, foram: pacote (41), classe (33), interface (31), não (29), pla (21), característica (18), acoplamento (17), coesão (16), alto (15), mais(14), projeto (14) e baixo (12).

A divisão das três classes representou claramente a otimização trazida pelas funções CM e FM, métricas convencionais e métricas dirigidas a características respectivamente.



Figura 6.18: Exp 03 - Nuvem de Palavras Corpus 1.



classes (nome específico da classe avaliada), interfaces e pacotes, apresentando seus nomes, bem como, suas interpretações. As palavras mais representativas da Classe 1 foram: acreditar, perceber, melhorado, elevado, projeto, pacote.

(...) Isto leva a crer que alguns **pacotes** já estavam **coesos** (ex **pacote BankingServiceServer1**) e outros poderiam (e **acredito** que foram) ser **melhorados** (ex. **pacote ATMClient1**). Com relação ao **pacote CardServiceServer2**, notei que a classe **CardAccount** simplesmente sumiu. Apesar de que, com isso, a **coesão** relacional tenha **melhorado** nesse **pacote**, achei estranho não ver os métodos ou atributos dessa classe na classe que ela depende (**DebitCard**) (n\_12).

(...) o mesmo vale para implementação da interface do **pacote gamemgr acredito** que essas classes faltantes impactariam fortemente no **acoplamento** (n\_14).

(...) com relação ao **acoplamento** desse **projeto** de **pla**, **acredito** que ele apresente um grau **elevado** de **acoplamento** uma vez que os relacionamentos entre classes são abundantes (n\_16).

(...) Pude **perceber** que esse **projeto** há um número considerável de relacionamento interno no **pacote BankingServiceServer1**. O **pacote ATMClient1** possui um número médio de relacionamentos internos. Já o **pacote** e **DisplayPromptsClient1** não possui nenhum relacionamento interno. Em relação as características, os **pacotes ATMClient1** e **DisplayPromptsClient** são pouco **coesos**, já os **pacotes BankingServiceServer1** e **CardServiceServer2** são altamente **coesos**. Portanto, de maneira

Tabela 6.26: Exp 03 - Dendograma Corpus 2.

Classe 4 22,8% Dificuldades na Avaliação dos Projetos	Classe 3 36,8% Seleção do Melhor Projeto	Classe 2 24,6% Coesão Relacional e Acoplamento	Classe 1 15,8% Avaliação Geral do Projeto
parecer	alternativo	observar	acreditar
faltar	banking	muito	perceber
também	alternativo	interno	médio
play	melhor	pacote	vez
gamectrl	coesão	externo	melhorado
ficar	menor	elemento	maneira
destacar	possuir	relacionamento	elevado
agm	vista	único	relação
	ponto	repetido	nível
	aparentar	realização	já
	opção	possível	projeto
	nomear	implementar	geral
	mobilemedia	generalização	pacote
	citado	exceção	valer
	pla	analisar	totalmente
		além	

geral pode perceber que há um nível de coesão relacional médio para baixo nesse **projeto** (n\_10).

A Classe 2, com incidência de 24,56%, aborda as discussões sobre os relacionamentos internos e externos das classes, interfaces e pacotes das PLAs avaliadas. Nesta classe os avaliadores focaram na interpretação das métricas de coesão relacional e acoplamento analisando de forma mais intensificada as propriedades trazidas nos projetos de PLAs. As palavras mais representativas foram: externo, interno, observar, relacionamento, nível.

*Entendo que esse projeto de PLA apresenta as seguintes características: Pacotes que apresentam uma única classe e/ou interface, que são utilizadas/implementadas por elementos **externos**. Isso pode ser observado nos pacotes Package13C, Package4E e Package86; Pacotes cujas classes não se relacionam, como é o caso do pacote DisplayPromptsClient1, mas que apresentam muitos **relacionamentos externos**; Pacotes com uma única interface, que é implementada por um elemento **externo** ao pacote. Isso acontece por exemplo, com o pacote Package41; Pacotes que apresentam um alto*

número de **relacionamentos internos**, como é o caso do pacote *ATMClient1*; Dadas as características apresentadas acima, entendo que a maioria dos pacotes apresentam um número maior de **relacionamentos externos**, em comparação com o número de **relacionamentos internos**. Uma exceção é feita ao pacote *ATMClient1*, que apresenta uma situação oposta. Por conta disso, entendo que a coesão relacional desse projeto de PLA é baixa (n.11).

É evidente nesse projeto o alto **nível** de acoplamento, principalmente entre os elementos dos pacotes *BankingServiceServer1*, *CardServiceServer2* e *DisplayPromptsClient1*. No pacote *BankingServiceServer1* o **nível** de elementos que tem dependências de outras classes chega a ser de 100%, ou seja, todas as classes desse pacote fornecem interfaces para outros pacotes isolados. No pacote *ATMClient1* é baixo o número de dependências. Portanto, de uma maneira geral pode perceber que há um **nível** alto de acoplamento entre as classes desse projeto (n.10).

Analisando o número médio de **relacionamentos internos**, é possível notar que houveram melhorias no sentido de diminuição desse número em alguns pacotes. Porém, em outros, o número permaneceu o mesmo. Isto leva a crer que alguns pacotes já estavam coesos (ex pacote *BankingServiceServer1*) e outros poderiam (e acredito que foram) ser melhorados (ex. pacote *ATMClient1*). Com relação ao pacote *CardServiceServer2*, notei que a classe *CardAccount* simplesmente sumiu. Apesar de que, com isso, a coesão relacional tenha melhorado nesse pacote, achei estranho não ver os métodos ou atributos dessa classe na classe que ela depende (*DebitCard*) (n.12).

A Classe 3 teve a maior incidência no Corpus 2. Com 36,8%, essa classe se relaciona com a escolha da “melhor” alternativa de projeto escolhida pelo avaliador. O Questionário 2 (Seção 6.3.1) apresenta 3 perguntas relacionadas a escolha de uma PLA do ponto de vista dos avaliadores. Dessa forma, esperava-se uma incidência maior de termos relacionados a uma “escolha” ou “seleção”. As palavras mais representativas foram: alternativo, melhor, pla, coesão, menor, possuir, vista, ponto, apresentar e opção. Os termos alternativo, melhor, pla, vista, ponto, apresentar e opção, estão relacionados ao momento em que o avaliador escolhe a melhor PLA.

(...)já na **pla** *var\_28\_banking* algumas classes estão isoladas ou sem nenhum tipo de relacionamento no meu **ponto de vista** diante das **plas** expostas a **melhor alternativa** do projeto sob o **ponto de vista de coesão** relacional é a **pla** *banking* (n.10).

(...) *var\_28\_mobilemedia* considerando a distribuição das classes e interfaces dentro dos pacotes e as respectivas características de cada um deles a **alternativa** citada no início da resposta se apresenta como a **melhor alternativa**, sob o aspecto da modularização de características (n.16).

(...)Entendo que a modularidade da **alternativa** nomeada *banking.pdf* é a mais adequada, pois entendo que foi a **alternativa** que apresentou a maior **coesão** e o menor nível de acoplamento, considerando aqui que coesão e acoplamento são bons indicadores de modularidade (n\_11).

Os termos menor, possuir e coesão se associam com a avaliação da métrica de coesão relacional.

(...)a mm apresenta boa **coesão** relacional pois o relacionamento de suas classes e interfaces são claros e fáceis de entender (n\_18).

(...) Analisando o número médio de relacionamentos internos, é possível notar que houveram **melhorias** no sentido de diminuição desse número em alguns pacotes. Porém, em outros, o número permaneceu o mesmo. Isto leva a crer que alguns pacotes já estavam coesos (ex pacote *BankingServiceServer1*) e outros poderiam (e acredito que foram) ser **melhorados** (ex. pacote *ATMClient1*) (n\_12).

(...)O pacote *GameCtrl* achei pouco coeso (n\_14).

(...)Com relação à **coesão** relacional, esse projeto de PLA é pouco ou fracamente coeso (n\_16).

A quarta classe teve uma incidência de 22,81%. A Classe 4 expressa a insegurança encontrada pelo avaliadores em escolher uma melhor alternativa de projeto de PLA. Isso fica evidente nas palavras mais representativas dessa classe: parecer, ficar. A interpretação das responsabilidades de cada elemento arquitetural em relação as métricas avaliadas causaram uma certa dúvida ao avaliador em escolher um projeto de PLA como a “melhor” alternativa.

O projeto do arquivo *VAR\_21\_agm-8156195255.pdf* é o que **parece** menos acoplado, e também não **parece** ter classe faltando (o que poderia indicar relacionamentos faltando que piorariam o acoplamento caso fossem completados) (n\_14).

O arquivo *agm.pdf* **parece** ter uma melhor coesão, pois está entrelaçado com a mesma característica (n\_15).

*VAR\_28\_MobileMedia-6161996518*. Considerando a distribuição das classes e interfaces dentro dos pacotes e as respectivas características de cada um deles, ... Numa análise visual, essa alternativa me **pareceu** ser a mais modularizada (ou a mais próxima da modularizada) (n\_16).

Um avaliador ao analisar uma PLA, julga estar faltando algo no projeto para que ele consiga tomar uma decisão com clareza de interpretação. (...) apesar de ter classes que **parecem** estar faltando, o que deixou algumas classes isoladas, o projeto do arquivo *VAR\_All\_agm-4444523912.pdf* é o que **parece** mais coeso (n\_14). Esse mesmo avaliador questionou a estrutura das PLAs, indicando que poderiam estar faltando partes



importantes para a compreensão total do projeto, uma vez que, há regiões desconexas na maioria das PLAs. (...) *Para esse quesito devo destacar que o diagrama parece estar incompleto, (...) Acredito que essas classes faltantes impactariam fortemente no acoplamento, e sem elas alguns pacotes foram totalmente isolados. Avaliando só as classes apresentadas o acoplamento é baixo, mas o projeto parece quebrado* (n\_14).

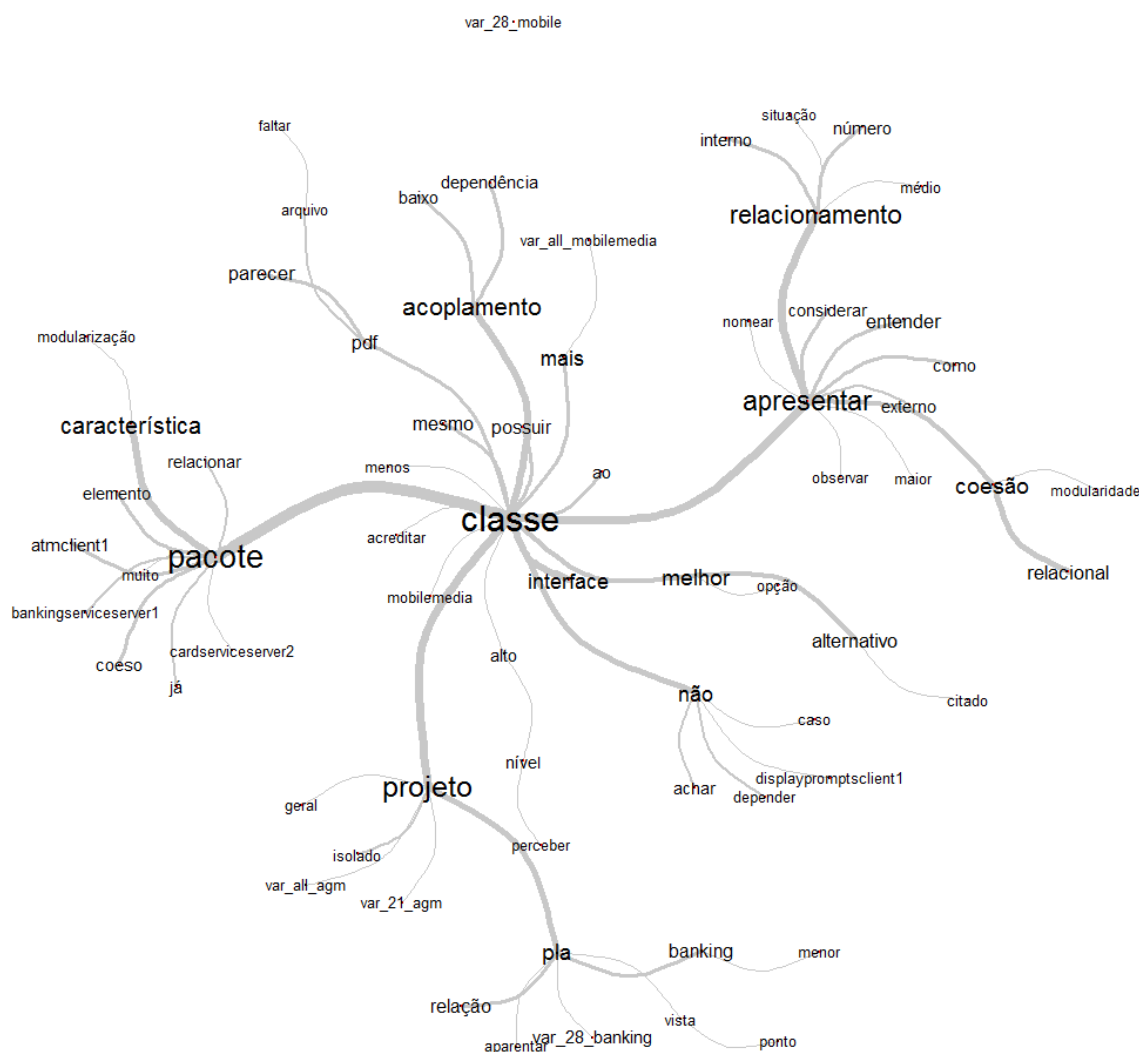
A Figura 6.19 apresenta a análise de similitude produzida pela ferramenta Iramuteq. Neste grafo os vértices principais são as palavras: **classe, pacote, projeto, apresentar, pla, acoplamento e relacionamento**. O Corpus 2 apresentou inúmeras análises de projeto de PLA, uma vez que, foram feitas 3 perguntas relacionadas a escolha da melhor alternativa, além, da pergunta relacionada a coesão relacional e ao acoplamento. Essa característica de análise pode ser identificada no grafo de similitude pois apresenta vértices como **pacote** que tem como ligações os termos **característica, relacionar, muito e coeso**, expressando uma interpretação sobre as propriedades que foram encontradas nos pacotes de algum projeto de PLA. Ainda, identifica-se uma avaliação mais aprimorada dos projetos de PLA retratando a avaliação feita pelas funções objetivo COE e ACLASS que são funções refinadas da função CM. A análise e interpretação das propriedades das PLAs foram consideradas de maneira mais rica em termos de coesão relacional e acoplamento. Tais características podem ser encontradas nos trechos a seguir:

*Analizando o número médio de relacionamentos internos, é possível notar que houveram melhorias no sentido de diminuição desse número em alguns pacotes. Porém, em outros, o número permaneceu o mesmo. Isto leva a crer que alguns pacotes já estavam coesos (ex pacote BankingServiceServer1) e outros poderiam (e acredito que foram) ser melhorados (ex. pacote ATMClient1). Com relação ao pacote CardServiceServer2, notei que a classe CardAccount simplesmente sumiu. Apesar de que, com isso, a coesão relacional tenha melhorado nesse pacote, achei estranho não ver os métodos ou atributos dessa classe na classe que ela depende (DebitCard)* (n\_12).

*Pude perceber que esse projeto há um número considerável de relacionamentos internos no pacote BankingServiceServer1. O pacote ATMClient1 possui um número médio de relacionamentos internos. Já o pacote e DisplayPromptsClient1 não possui nenhum relacionamento interno. Em relação as características, os pacotes ATMClient1 e DisplayPromptsClient são pouco coesos, já os pacotes BankingServiceServer1 e CardServiceServer2 são altamente coesos. Portanto, de maneira geral pude perceber que há um nível de coesão relacional médio para baixo nesse projeto* (n\_10).

*Entendo que esse projeto de PLA apresenta as seguintes características: Pacotes que apresentam uma única classe e/ou interface, que são utilizadas/implementadas por elementos externos ... Isso acontece por exemplo, com o pacote Package41; Pacotes*

Figura 6.19: Exp 03 - Similitude Corpus 2.



que apresentam um alto número de relacionamentos internos, como é o caso do pacote *ATMClient1*; Dadas as características apresentadas acima, entendo que a maioria dos pacotes apresentam um número maior de relacionamentos externos, em comparação com o número de relacionamentos internos ... Por conta disso, entendo que a coesão relacional desse projeto de PLA é baixa (n.11).

Analisando o número médio de relacionamentos internos, é possível notar que houveram melhorias no sentido de diminuição desse número em alguns pacotes. Porém, em outros, o número permaneceu o mesmo. Isto leva a crer que alguns pacotes já estavam coesos (ex pacote *BankingServiceServer1*) e outros poderiam (e acredito que foram) ser melhorados (ex. pacote *ATMClient1*) (n.12).

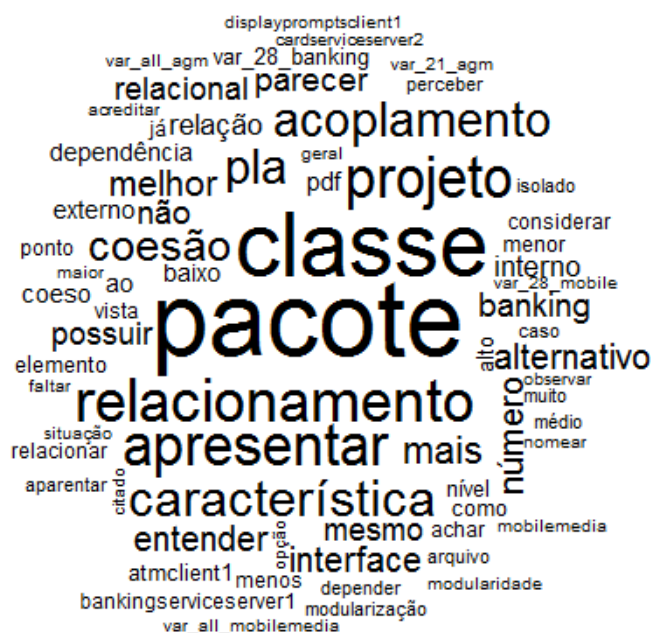
*Há muito acoplamento entre as classes e interfaces. Muitas características sendo repetida nessas classes (n\_15).*

*(...) Considerando que a coesão relacional é diretamente relacionada a reusabilidade do produto, de acordo com a função CM; a opção citada anteriormente se apresenta como a melhor alternativa de projeto (...) Considerando que o baixo acoplamento de classes é um fator que corrobora para a reusabilidade e manutenibilidade do código desenvolvido, a opção citada anteriormente se apresenta como a melhor alternativa de projeto (n\_16).*

*Seria a MobileMedia, pois a mesma possui menos classes dependentes umas das outras. Chegando a ter pouco mais da metade de dependências em relação a VAR\_All\_MobileMedia (...) pois a mesma possui mais módulos em relação as demais e se subentende que sendo assim a PLA está mais dividida e mais legível com uma melhor manutenção e melhor desempenho (n\_18).*

A Nuvem de Palavras se baseia na frequência das palavras utilizadas no corpus, a Figura 6.18 apresenta a Nuvem de palavras do Corpus 2. Identificou-se que as palavras mais utilizadas, com suas respectivas frequências, foram: pacote (46), classe (38), relacionamento (25), apresentar (23), projeto (23), característica (20), acoplamento (19), pla (18), coesão (16), mais (14), melhor (13), alternativo (12), entender (12), interface (12) e número (12). Esses termos corroboram a análise realizada no grafo de similitude.

**Figura 6.20:** Exp 03 - Nuvem de Palavras Corpus 2.



### Análise do Corpus 3

O Corpus 3 foi composto somente pelas respostas da sexta questão dos dois questionários utilizados: *Quais propriedades você julga importantes para se avaliar em um projeto de PLA?*. O objetivo dessa questão foi listar quais propriedades para avaliação de PLAs são importantes do ponto de vista dos avaliadores. O estudo desse corpus envolveu somente a Análise de Similitude e Nuvem de Palavras. O Corpus 3 está apresentado no Apêndice C.3.

O grafo de similitude, apresentado na Figura 6.21, destaca os termos **coesão** e **acoplamento** ramificando para as palavras **característica**, **pacote**, **classe**, **modularização**, **projeto**, **pla** e **acreditar**. Contudo todas as respostas se concentram na utilização da coesão como uma propriedade significativa para avaliação de PLAs.

*Julgo mais essenciais a **coesão** e o **acoplamento**. Sempre procuro analisar e definir as responsabilidades para os pacotes, interfaces, classes e **características** que serão modeladas e são realmente fundamentais para um bom projeto (n\_1).*

***Coesão**, **acoplamento**, **reusabilidade** em termos de elementos abstratos, **tamanho** e **estabilidade** (n\_2).*

***Acoplamento**, **coesão**, **modularização**. Entretanto é importante observar o nome dos métodos, atributos, classes pacotes. Isso orienta o programador a encontrar e navegar na arquitetura (n\_4)*

*Primeiramente o **acoplamento**, por segundo a **coesão** e então as demais propriedades. (n\_8).*

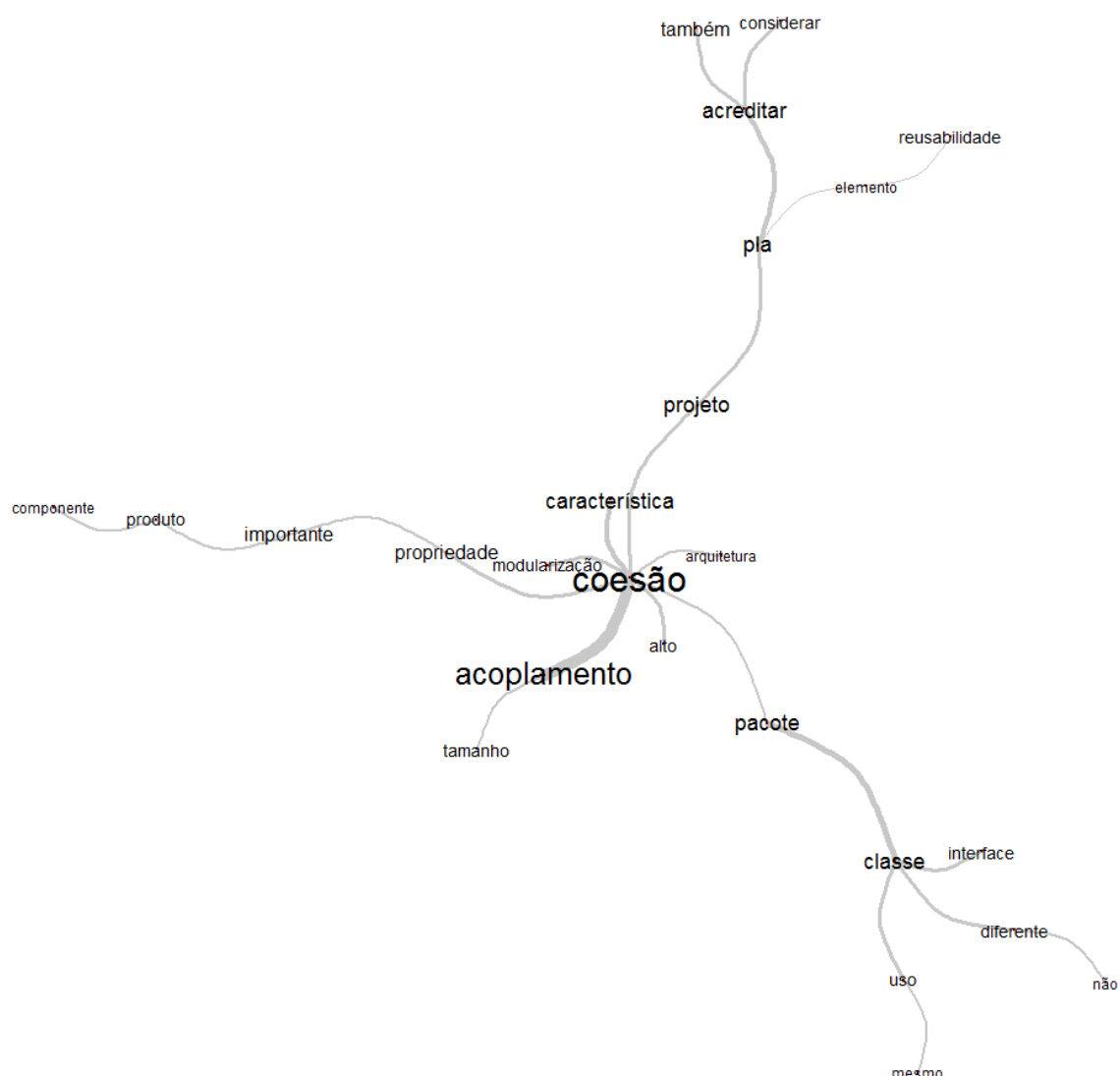
*Acredito que tendo uma alta **coesão** é melhor, pois permite centralizar as propriedades. (n\_15)*

*Uma arquitetura madura e sustentável, apresenta **alta coesão** e **baixo acoplamento**, pois cada um deles tem um propósito específico que visa melhorar o projeto da PLA (n\_18).*

A Nuvem de Palavras se baseia na frequência das palavras utilizadas no corpus, a Figura 6.22 apresenta a Nuvem de palavras do Corpus 3. Identificou-se que as palavras mais utilizadas, com suas respectivas frequências, foram: coesão (11), acoplamento (9), característica (7), classe (7), pacote (7), acreditar (5), pla (5), projeto (5), propriedade (4) e importante (4).

Com a análise do Corpus 3 foi possível listar quais propriedades para avaliação de PLAs são importantes do ponto de vista dos avaliadores, essas propriedades foram identificadas direta e indiretamente. As propriedades foram: coesão e acoplamento de classes, modularização de características, reusabilidade, extensibilidade e elegância.

**Figura 6.21:** Exp 03 - Similitude Corpus 3.

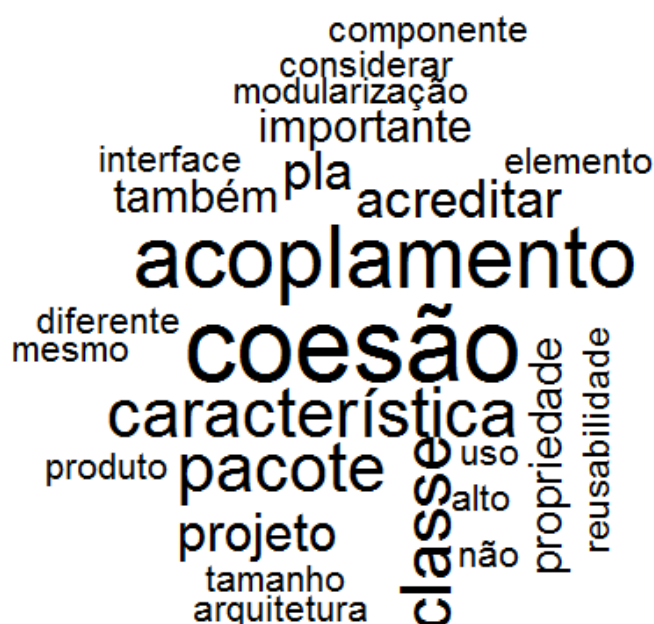


Os termos coesão e acoplamento de classes, modularização de características estavam explícitos nas perguntas do questionário. Já os termos reusabilidade, extensibilidade e elegância vieram das seguintes respostas:

(...) *Sempre procuro analisar e definir as responsabilidades para os pacotes, interfaces, classes e características que serão modeladas e são realmente fundamentais para um bom projeto. Deste modo, é possível tornar o projeto mais consistente, simples, reutilizável* (n\_01).

(...) *acredito que a extensibilidade e a manutenibilidade sejam também importantes* (n\_09).

**Figura 6.22:** Exp 03 - Nuvem de Palavras Corpus 3.



(...) *Não criar pacotes muito grandes que abriguem características diferentes mas também não criar muitos pacotes que partilham do mesmo módulo (n\_14).*

(...) *acredito que seja importante a definição de componentes bem delimitados; a documentação dos requisitos atendidos, sejam parcial ou totalmente, por cada componente; a formalização dos relacionamentos entre os componentes, independentemente da notação adotada; (n\_16).*

A próxima subseção apresenta as respostas às questões de pesquisa levantadas no Exp 03.

### **Contabilização das soluções escolhidas**

Foi realizada uma análise com objetivo de identificar se os avaliadores estão escolhendo os melhores projetos de PLA de acordo com os valores indicados pelo estudo quantitativo.

No Questionário 1 havia somente uma pergunta relacionada a escolha da melhor alternativa de projeto no quesito de modularização de características (*Qual a melhor alternativa de projeto sob o ponto de vista de modularização de características?*). Para realizar essa análise, foi construída a Tabela 6.27 com a seguinte estrutura: a coluna 1 o projeto de PLA analisado, a coluna 2 indica qual solução tem melhor modularização de características (menor FM), a coluna 3 apresenta o número dos avaliadores, a coluna 4 indica qual solução foi escolhida pelo avaliador correspondente e, a coluna 5 informa

se a solução escolhida pelo avaliador tem algum padrão de projeto aplicado e, em caso positivo, qual o padrão. As Tabelas 6.28 e 6.29 exibem as mesmas colunas da Tabela 6.27 com diferença na coluna 2.

Na Tabela 6.27, um terço dos avaliadores escolheram a solução com menor FM. Dois deles indicaram que o melhor projeto ainda seria o original mas, como visto na subseção anterior, houveram alguns questionamentos quanto a nomenclatura e disposição dos artefatos que compõem o projeto, que pode ter sido a causa dessa escolha.

**Tabela 6.27:** Exp 03 - Contabilização de acertos da questão 5 - Corpus 1.

PLA	Solução de melhor FM e (CM; FM)	Avaliador	Solução escolhida (CM; FM)	Aplicação de Padrão de Projeto
AGM	VAR_All_agm-4953622311 / (53,1; 510,0)	n_04	VAR_All_agm-4953622311 / (53,1; 510,0)	Sem padrão
		n_05	VAR_1_agm-1137411359 / (49,1; 528,0)	Strategy
		n_06	VAR_1_agm-1137411359 / (49,1; 528,0)	Strategy
BANK	VAR_All_banking-8489715735 / (141,5; 230,0)	n_01	BANK	Sem padrão
		n_02	VAR_All_banking-7294316121 / (127,5; 281,0)	Strategy
		n_03	VAR_All_banking-8489715735 / (141,5; 230,0)	Sem padrão
MOM	VAR_All_MobileMedia-7172446371 / (72,14; 407,0)	n_07	MOM	Sem padrão
		n_08	VAR_All_MobileMedia-7172446371 / (72,14; 407,0)	Sem padrão
		n_09	VAR_All_MobileMedia-3748334128 / (50,09; 862,0)	Sem padrão

(...) *Tal escolha foi devido ao fato dessa alternativa definir estereótipos e notações para as interfaces e classes do projeto (optional, mandatory, dentre outros). Acredito que isso torna as características mais consistentes e modulares, propiciando a forte coesão e o baixo acoplamento. Além disso, a quantidade de dependências entre interfaces e classes*

*nesta alternativa de projeto foi minimizada, tornando as características mais consistentes e modulares (n\_1).*

*(...) MobileMedia. Achei que as responsabilidades ficaram melhor divididas em seus pacotes. Fica mais fácil identificar onde as responsabilidades estão (n\_7).*

*VAR\_All\_agm-4953622311 eu ficaria entre ela e a AGM. Me incomodou ter um pacote na AGM com muitas classes. Entretanto, as outras arquiteturas criaram pacotes com nomes estranhos, e que não pareciam fazer muito sentido (n\_4).*

Na coluna 5 da Tabela 6.27 é possível visualizar que um terço dos avaliadores escolheram soluções com padrão de projeto aplicado. Um fato que pode ter favorecido essas escolhas é que a busca local tem um melhor desempenho em PLAs modulares. Além disso, o fato de ter um padrão de projeto aplicado implica em uma solução mais amigável ao olhos de um especialista.

A Tabela 6.28 apresenta as escolhas realizadas pelos avaliadores na questão 3 do Corpus 2 (*Qual a melhor alternativa de projeto sob o ponto de vista de coesão relacional?*). A coluna 2 apresenta as soluções com melhor coesão relacional (menor COE).

Nenhum dos avaliadores escolheu a solução de melhor coesão relacional. Cerca de 50% dos avaliadores escolheram os projetos originais. Contudo, foram feitas justificativas com base nas propriedades arquiteturais de cada PLA.

*A PLA banking e a PLA VAR\_28\_banking-7114272521 apresentam um alto nível de coesão relacional, porém na PLA banking pude perceber que a coesão relacional é maior entre as classes, visto que todas as classes possuem algum tipo de relacionamento, já na PLA VAR\_28\_banking-7114272521 algumas classes estão isoladas ou sem nenhum tipo de relacionamento (...) (n\_10).*

*Entendo que a alternativa nomeada banking.pdf apresenta a melhor situação de coesão relacional, pois a maioria dos relacionamentos entre classes e/ou interfaces se concentra no pacote (...) (n\_11).*

*A melhor opção é a MobileMedia, pois a mesma possui menos classes e as mesmas possuem menos responsabilidades (n\_18).*

Com exceção de uma solução (VAR\_All\_agm-4444523912) todas as outras soluções escolhidas pelo avaliadores apresentaram padrões de projeto, retratando novamente a preferência dos avaliadores, por projetos com alguma solução de projeto implantada.

A Tabela 6.28 apresenta as escolhas realizadas pelos avaliadores na questão 4 do Corpus 2 (*Qual a melhor alternativa de projeto sob o ponto de vista de acoplamento de classes?*). A coluna 2 apresenta as soluções com melhor acoplamento de classes (menor ACLASS).



**Tabela 6.28:** Exp 03 - Contabilização de acertos da questão 3 - Corpus 2.

PLA	Solução de melhor COE e (COE; ACLASS)	Avaliador	Melhor solução escolhida (COE; ACLASS)	Aplicação de Padrão de Projeto
AGM	VAR_All_agm-1192958118 / (25,0; 26,0)	n_13	VAR_21_agm-8156195255 / (28,0; 17,0)	Strategy
		n_14	VAR_All_agm-4444523912 / (29,0; 9,0)	Sem padrão
		n_15	AGM	Sem padrão
BANK	VAR_All_banking-4119842252 / (13,0; 91,0)	n_10	BANK	Sem padrão
		n_11	BANK	Sem padrão
		n_12	VAR_28_banking-7114272521 / (14,0; 91,0)	Strategy
MOM	VAR_All_MobileMedia-6111288751 / (28,0; 14,0)	n_16	MOM	Sem padrão
		n_17	VAR_28_MobileMedia-6161996518 / (28,0; 14,0)	Strategy
		n_18	MOM	Sem padrão

Um terço das escolhas dos avaliadores corresponderam aos projetos de melhor acoplamento de classes, outro um terço escolheu as PLAs originais e, o restante dos avaliadores escolheram projetos com aplicação de padrões de projeto, exceto o avaliador n\_16.

*Entendo que a alternativa nomeada banking.pdf apresenta o grau mais baixo de acoplamento de classes, entre as alternativas. Tal entendimento se justifica pelo fato de que alguns tipos de relacionamentos, tais como o de dependência, são reduzidos em comparação com as outras alternativas (n\_11).*

*VAR\_All\_MobileMedia-6111288751. Considerando que o baixo acoplamento de classes é um fator que corrobora para a reusabilidade e manutenibilidade do código desenvolvido, a opção citada anteriormente se apresenta como a melhor alternativa de projeto (n\_16).*

A questão 5 do corpus 2 (*Qual a melhor alternativa de projeto sob o ponto de vista de modularização de características?*) não foi contabilizada pois a função FM não foi utilizada e por esse motivo não houve como realizar a comparação entre a solução de melhor FM com as escolhas dos avaliadores. Apesar disso, foi contabilizado o número de soluções selecionadas pelos avaliadores, assim, somente três avaliadores escolheram os projetos originais sendo dois na PLA BANK e um na PLA AGM, dessa forma 2/3

**Tabela 6.29:** Exp 03 - Contabilização de acertos da questão 4 - Corpus 2.

PLA	Solução de melhor ACLASS e (COE; ACLASS)	Avaliador	Melhor solução escolhida (COE; ACLASS)	Aplicação de Padrão de Projeto
AGM	VAR_All_agm-4444523912 / (29,0; 9,0)	n_13	VAR_All_agm-4444523912 / (29,0; 9,0)	Sem padrão
		n_14	VAR_21_agm-8156195255 / (28,0; 17,0)	Strategy
		n_15	VAR_All_agm-4444523912 / (29,0; 9,0)	Sem padrão
BANK	VAR_28_banking-7114272521 / (14,0; 91,0)	n_10	BANK	Sem padrão
		n_11	BANK	Sem padrão
		n_12	VAR_28_banking-7114272521 / (14,0; 91,0)	Strategy
MOM	VAR_All_MobileMedia-8589111691 / (29,0; 10,0)	n_16	VAR_All_MobileMedia-6111288751 / (28,0; 14,0)	Sem padrão
		n_17	VAR_28_MobileMedia-6161996518 / (28,0; 14,0)	Strategy
		n_18	MOM	Sem padrão

dos avaliadores escolheram soluções otimizadas e, dentre elas, cerca de 67% das soluções continham o padrão de projeto *Strategy* aplicado.

### Respondendo a Questão de Pesquisa

Esta seção responde a questão de pesquisa **QP4:** *As soluções encontradas pela OPLA-Tool com o AM são consideradas boas, do ponto de vista de arquitetos de software?*

Analisando as três últimas tabelas é possível identificar que 37% dos avaliadores escolheram os projetos de PLA originais. Os outros 63% escolheram soluções otimizadas pela abordagem memética e, por esse motivo, é possível dizer que as soluções encontradas pela OPLA-Tool no contexto da MOA4PLA são boas no ponto de vista de arquitetos de software.

## 6.4 Considerações Finais

Com o Exp 01 foi possível identificar que o código dos operadores de mutação fundamentais necessitava de modificações, além de, mostrar a efetividade da busca local na aplicação de padrões de projeto alcançando soluções com *fitness* menores.

O Exp 02 foi possível identificar qual par de funções objetivo melhor retratava as modificações trazidas pela busca local e, apontar que o AM encontrou melhores soluções do que o AG.

Apesar do esforço dispendido para minimizar o viés durante a realização do estudo qualitativo alguns pontos de viés não puderam ser corrigidos. O principal ponto foi o fato de que o estudo foi executado virtualmente e, na maioria dos casos, à distância. O experimento à distância inibiu os avaliadores a sanarem pequenas dúvidas e, a avaliação dos projetos de PLA virtualmente prejudicou a visão total do projeto dificultando a análise.

Houveram críticas positivas e negativas quanto a reusabilidade das PLAs o que permite realizar melhorias na geração dos projetos arquiteturais, um exemplo disso é deixar mais amigável o nome de classes e pacotes gerados pela ferramenta, contudo, em um ambiente real o arquiteto pode renomear os elementos arquiteturais de acordo ao domínio da LPS.

Dentre as respostas um avaliador indicou sua opinião sobre o estudo de otimizações em projetos de PLA. *Acredito que o conteúdo deste estudo é relevante para se avaliar projetos de PLA, considerando os níveis de coesão alto e de acoplamento baixo, e também a avaliação de modularização das características.* (n\_10).

O próximo capítulo apresenta as conclusões alcançadas com o desenvolvimento deste trabalho, bem como, os trabalhos futuros identificados.

---

## Conclusão

---

A identificação da lacuna de pesquisa, por meio de um MS, com relação a não utilização de AMs na otimização de projeto de LPS, motivou a proposta deste trabalho. Os resultados promissores de AM para resolver outros problemas da Engenharia de Software reforçaram a motivação.

Neste contexto, este trabalho propôs o desenvolvimento de um AM, no contexto da MOA4PLA. Essa proposta investigou a utilização da abordagem memética para otimização de projeto de PLA. A hipótese principal foi: a aplicação de algoritmos meméticos no processo de otimização da MOA4PLA é capaz de obter projetos de PLA melhores do que os projetos obtidos na versão original da MOA4PLA.

A avaliação da abordagem memética se deu por meio de um estudo exploratório, um estudo experimental e, por fim, um estudo qualitativo.

Durante o estudo exploratório foi identificado que os operadores de mutação fundamentais da busca global estariam corrompendo os padrões aplicados pela busca local. Por esse motivo, restrições foram implementadas nos operadores de mutação fundamentais de modo que eles não pudessem alterar regiões do projeto de PLA onde haviam padrões aplicados.

O estudo experimental identificou que a abordagem memética encontra soluções com melhor balanceamento entre os objetivos do que o AG. Além disso, foi identificado que a busca local alcança melhores resultados quando otimiza PLAs que contém características bem modularizadas. E, por fim, observou-se que as funções objetivo COE e ACLASS são mais sensíveis para detectar as melhorias trazidas pela abordagem memética.

A análise quantitativa baseada em indicadores de qualidade indicou que não há diferença estatística entre os algoritmos para os projetos de PLA envolvidos. Contudo, a

hipótese principal pode ser mantida em razão de que o AM identificou soluções melhores, em termos de *fitness*, do que as alcançadas pelo AG.

O estudo qualitativo foi desenvolvido com auxílio de arquitetos de software (avaliadores) e identificou possíveis melhorias na apresentação gráfica dos projetos de PLAs. Alguns itens foram questionados: nomenclatura dos artefatos que compõe os projetos (classe, pacotes e métodos) devem ser mais amigáveis; a apresentação dos projetos deve ser de forma impressa, ao invés de digital, em razão do tamanho dos projetos. Contudo, todas as melhorias podem ser resolvidas.

Na análise qualitativa identificou-se que, de forma geral, tanto o algoritmo memético quanto o genético otimizam os projetos de PLA de acordo com as funções objetivo. No entanto, no momento da escolha das melhores alternativas de projeto os avaliadores mostraram preferir soluções que contêm padrões de projeto.

A abordagem memética pode ser utilizada para otimização de projetos de PLA na MOA4PLA, uma vez que, foram identificados resultados favoráveis em sua aplicação. Entretanto, precisa ser lapidada conforme os achados do estudo qualitativo, apresentados no capítulo anterior.

As principais contribuições alcançadas com este trabalho foram: encontrar resultados favoráveis da utilização de algoritmos meméticos na abordagem da MOA4PLA e, com isso, beneficiar a atividade de projeto de PLA a qual é uma tarefa difícil. Identificar que a ferramenta OPLA-Tool está gerando soluções viáveis sob o ponto de vista de arquitetos de software. Além, de expandir o estado da arte de SBSE inserindo a abordagem memética como uma solução favorável para resolver o problema de otimização multiobjetivo de projeto de PLA.

## 7.1 Dificuldades Encontradas

Durante o desenvolvimento deste trabalho foram identificadas dificuldades. Esta subseção lista as limitações e dificuldades, além da forma como foram mitigadas.

1. A principal limitação foi o número reduzido de projetos de LPS disponíveis para realizar estudos experimentais. Apesar disso, foram utilizadas LPS de diferentes domínios e tamanhos, além de que, foram repetidos os experimentos 30 vezes para garantir sua validade estatística.
2. Em relação ao código da ferramenta OPLA-Tool, a dificuldade foi na expansão do código. Devido ao número de desenvolvedores envolvidos na implementação dos módulos da ferramenta, inúmeros métodos de desenvolvimento foram utilizados e

isso inseriu complexidade na momento de expansão do código. Essa limitação foi superada estudando-se o código de forma integral.

3. Como a OPLA-Tool está construída sobre o *framework* JMetal, se faz necessária uma constante atualização das bibliotecas utilizadas no OPLA-Encoding e OPLA-Decoding. Neste trabalho, foram realizadas pequenas alterações nestes módulos para desenvolver os experimentos com os quatro projetos de PLA (AGM, BANK, MOM e BET).
4. Em relação ao estudo qualitativo, a limitação foi encontrar um número razoável de avaliadores para o estudo.
5. Ainda com relação ao estudo qualitativo, não foi possível reunir todos os avaliadores em um ambiente controlado para realizar o experimento. Muito embora houvesse um acompanhamento remoto, alguns avaliadores ficaram inibidos em sanar pequenas dúvidas o que atribuiu viés às respostas. Essa limitação pode ser minimizada com a repetição do estudo qualitativo.

Novas ideias foram surgindo no desenvolvimento deste trabalho e são apresentadas na próxima seção.

## 7.2 Trabalhos Futuros

Nessa seção são apresentados possíveis trabalhos futuros que podem ser desenvolvidos com base neste trabalho.

Considerar o número de avaliações de *fitness* da busca local juntamente com a global implicaria na diminuição do custo em processamento mantendo a convergência do algoritmo, uma vez que, a busca local aumenta o espaço de busca.

Atualmente o operador de mutação fundamental a ser aplicado é escolhido por meio de um processo de roleta, em que, todos os operadores têm a mesma probabilidade de serem escolhidos. Da mesma forma, a busca local define qual padrão de projeto aplicar por meio do processo de roleta. Poderia ser desenvolvido um método de avaliação que definisse quais operadores de mutação fundamentais alcançam melhores resultados, para que o processo de roleta operasse somente com tais operadores. De forma geral, a ideia seria, identificar quais dos seis operadores são mais favoráveis à região analisada e restringir a roleta somente à esses operadores, aumentando a chance de um operador mais proveitoso ser selecionado. Essa ideia pode ser estendida ao operador local.

Como houve implementações de paralelismo em duas versões da busca local (Bestof12 e UntilBest) resultando no melhor aproveitamento de recursos de processamento,

seria possível estender esse paralelismo para o processo integral da busca global o que acarretaria em uma diminuição considerável no tempo de otimização dos projetos.

Seria interessante realizar novos estudos envolvendo outras funções objetivo, principalmente com a função FM, que prioriza a modularização dos projetos de PLA em termos de característica. Essa modularização pode favorecer a busca local em termo de otimização, mesmo em projetos não modularizados. Um exemplo seria executar testes envolvendo três funções objetivo: COE, ACLASS e FM.

Por fim, poderia ser implementado um operador de mutação responsável por realizar algo semelhante a um *shake* no processo de evolução impedindo que o algoritmo memético fique preso em ótimos locais.

## REFERÊNCIAS

---

ARCURI, A.; BRIAND, L. A practical guide for using statistical tests to assess randomized algorithms in software engineering. In: *2011 33rd International Conference on Software Engineering (ICSE)*, IEEE, 2011, p. 1–10.

BAI, X.; LI, Y.; YANG, K. Hybrid genetic algorithm and its application in structural optimization design. In: *Information Management and Engineering (ICIME), 2010 The 2nd IEEE International Conference on*, 2010, p. 415–417.

CHAWLA, P.; CHANA, I.; RANA, A. A novel strategy for automatic test data generation using soft computing technique. *Frontiers of Computer Science*, v. 9, n. 3, p. 346–363, 2015.

CHOMA, J.; COLANZI, T. E.; AMARAL, A. M. Application of memetic algorithms in the search-based product line architecture design: An exploratory study. *International Conference on Enterprise Information System (ICEIS)*, 2017.

CLEMENTS, P.; NORTHROP, L. A framework for software product line practice, version 4.2. 2007.

Disponível em <http://www.sei.cmu.edu/productlines/tools/framework/>

COELLO, C. C.; LAMONT, G. B.; VAN VELDHUIZEN, D. A. *Evolutionary algorithms for solving multi-objective problems*. Springer Science & Business Media, 2007.

COLANZI, T. E. *Uma abordagem de otimização multiobjetivo para projeto arquitetural de linha de produto de software*. Tese de Doutorado, Universidade Federal do Paraná, 2014.

Disponível em <http://dspace.c3sl.ufpr.br/dspace/bitstream/handle/1884/36061/R%20-%20T%20-%20THELMA%20ELITA%20COLANZI.pdf?sequence=1>

COLANZI, T. E.; VERGILIO, S. R. A comparative analysis of two multi-objective evolutionary algorithms in product line architecture design optimization. In: *IEEE 26th*



*International Conference on Tools with Artificial Intelligence (ICTAI)*, IEEE, 2014, p. 681–688.

COLANZI, T. E.; VERGILIO, S. R.; GIMENES, I.; OIZUMI, W. N. A search-based approach for software product line design. In: *Proceedings of the 18th International Software Product Line Conference-Volume 1*, ACM, 2014, p. 237–241.

CONTI, G. Avaliação de projeto baseado em busca para arquitetura de linha de produto de software. Trabalho de Conclusão de Curso (Bacharel em Informática), 2016.

CONTIERI JR, A. C.; CORREIA, G. G.; COLANZI, T. E.; GIMENES, I. M.; OLIVEIRA JR, E. A.; FERRARI, S.; MASIERO, P. C.; GARCIA, A. F. Extending uml components to develop software product-line architectures: lessons learned. In: *European Conference on Software Architecture*, Springer, 2011, p. 130–138.

COPLIEN, J. O. *Multi-paradigm design for c++*. Addison-Wesley, 1999.

DEB, K.; PRATAP, A.; AGARWAL, S.; MEYARIVAN, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, v. 6, n. 2, p. 182–197, 2002.

DELGADO, Y. V.; COLANZI, T. E.; VERGILIO, S. R.; DOS SANTOS, M. C. B. An enhanced evaluation model for search-based product line architecture design. *Congresso Iberoamericano em Engenharia de Software - CibSE*, 2017.

DONEGAN, P. M.; MASIERO, P. C. Design issues in a component-based software product line. In: *Simpósio Brasileiro de Componentes, Arquiteturas e Reúso de Software (SBCARS)*, 2007, p. 3–16.

ETXEBERRIA, L.; SAGARDUI, G. Product-line architecture: New issues for evaluation. In: *Software Product Lines*, Springer, p. 174–185, 2005.

FÉDERLE, É. L.; DO NASCIMENTO FERREIRA, T.; COLANZI, T. E.; VERGILIO, S. R. Opla-tool: a support tool for search-based product line architecture design. In: *Proceedings of the 19th International Conference on Software Product Line*, ACM, 2015, p. 370–373.

FRASER, G.; ARCURI, A.; MCMINN, P. A memetic algorithm for whole test suite generation. *Journal of Systems and Software*, v. 103, p. 311–327, 2015.

GAMMA, E.; HELM, R.; JOHNSON, R.; VLISSIDES, J. Padrões de projeto-soluções reutilizáveis de software orientado a objetos, 2004, ed. 2000.

GARZON-RODRIGUEZ, L. P.; DIOSA, H. A.; ROJAS-GALEANO, S. Deconstructing gas into visual software components. In: *Proceedings of the Companion Publication of the 2015 on Genetic and Evolutionary Computation Conference*, GECCO Companion '15, New York, NY, USA: ACM, 2015, p. 1125–1132 (*GECCO Companion '15*, ).

GOLDBERG, D. E.; *et al.* *Genetic algorithms in search optimization and machine learning*, v. 412. Addison-wesley Reading Menlo Park, 1989.

GOMAA, H. *Software modeling and design: Uml, use cases, patterns, and software architectures*. Cambridge University Press, 2011.

GUIZZO, G. *Uso de padrões em projeto arquitetural baseado em busca de linha de produto de software*. Dissertação de Mestrado, Universidade Federal do Paraná, 2014. Disponível em <http://dspace.c3sl.ufpr.br/dspace/handle/1884/37081>

GUIZZO, G.; COLANZI, T. E.; VERGILIO, S. R. A pattern-driven mutation operator for search-based product line architecture design. In: *International Symposium on Search Based Software Engineering (SSBSE)*, p. 77–91, 2014.

HARMAN, M.; JIA, Y.; KRINKE, J.; LANGDON, W. B.; PETKE, J.; ZHANG, Y. Search based software engineering for software product line engineering: A survey and directions for future work. In: *Proceedings of the 18th International Software Product Line Conference - Volume 1*, SPLC '14, New York, NY, USA: ACM, 2014, p. 5–18 (*SPLC '14*, ).

HARMAN, M.; MANSOURI, A. Search based software engineering: Introduction to the special issue of the iee transactions on software engineering. *IEEE transactions on Software Engineering*, v. 36, n. 6, p. 737, 2010.

HARMAN, M.; MANSOURI, S. A.; ZHANG, Y. Search-based software engineering: Trends, techniques and applications. *ACM Computing Surveys (CSUR)*, v. 45, n. 1, p. 11, 2012.

HARMAN, M.; MCMINN, P. A theoretical and empirical study of search-based testing: Local, global, and hybrid search. *IEEE Transactions on Software Engineering*, v. 36, n. 2, p. 226–247, 2010.

- ISHIBUCHI, H.; YOSHIDA, T.; MURATA, T. Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling. *IEEE Transactions on Evolutionary Computation*, v. 7, n. 2, p. 204–223, 2003.
- JEYA MALA, D.; SABARI NATHAN, K.; BALAMURUGAN, S. Critical components testing using hybrid genetic algorithm. *SIGSOFT Softw. Eng. Notes*, v. 38, n. 5, p. 1–13, 2013.
- KANG, K. C.; COHEN, S. G.; HESS, J. A.; NOVAK, W. E.; PETERSON, A. S. *Feature-oriented domain analysis (FODA) feasibility study*. Relatório Técnico, DTIC Document, 1990.
- KNOWLES, J. D.; CORNE, D. W. Approximating the nondominated front using the pareto archived evolution strategy. *Evolutionary Computation*, v. 8, n. 2, p. 149–172, 2000.
- LIASKOS, K.; ROPER, M. Hybridizing evolutionary testing with artificial immune systems and local search. In: *IEEE International Conference on Software Testing Verification and Validation Workshop, 2008. ICSTW'08.*, IEEE, 2008, p. 211–220.
- LINDEN, F. J. V. D.; ROMMES, E. *Software product lines in action - the best industrial practice in product line engineering*. Springer, 2007.
- LINDEN, F. J. V. D.; SCHMID, K.; ROMMES, E. *Software product lines in action: the best industrial practice in product line engineering*. Springer Science & Business Media, 2007.
- MARIANI, T.; VERGILIO, S. R.; COLANZI, T. E. Optimizing aspect-oriented product line architectures with search-based algorithms. In: *International Symposium on Search Based Software Engineering*, Springer, 2015, p. 173–187.
- MOLINA, D.; LOZANO, M.; GARCÍA-MARTÍNEZ, C.; HERRERA, F. Memetic algorithms for continuous optimisation based on local search chains. *Evolutionary Computation*, v. 18, n. 1, p. 27–63, 2010.
- MOSCATO, P.; COTTA, C. A gentle introduction to memetic algorithms. In: *Handbook of metaheuristics*, Springer, p. 105–144, 2003.
- Ó CINNÉIDE, M.; NIXON, P. Automated software evolution towards design patterns. In: *Proceedings of the 4th international workshop on Principles of software evolution*, ACM, 2001, p. 162–165.

- OCHOA, G.; VEREL, S.; TOMASSINI, M. First-improvement vs. best-improvement local optima networks of nk landscapes. In: *International Conference on Parallel Problem Solving from Nature*, Springer, 2010, p. 104–113.
- OLIVEIRAJR, E.; GIMENES, I. M.; MALDONADO, J. C.; MASIERO, P. C.; BARROCA, L. Systematic evaluation of software product line architectures. *Journal of Universal Computer Science*, v. 19, p. 25–52, 2013a.
- OLIVEIRAJR, E. A.; GIMENES, I. M.; MALDONADO, J. C. Systematic management of variability in uml-based software product lines. *Journal of Universal Computer Science*, v. 16, n. 17, p. 2374–2393, 2010.
- OLIVEIRAJR, E. A.; GIMENES, I. M.; MALDONADO, J. C.; MASIERO, P. C.; BARROCA, L. Systematic evaluation of software product line architectures. *Journal of Universal Computer Science*, v. 19, n. 1, p. 25–52, 2013b.
- OLIVEIRAJR, E. A. D.; MALDONADO, J. C.; DE SOUZA GIMENES, I. M. Uma revisão sistemática sobre avaliação de linha de produto de software. *Relatórios Técnicos do ICMC/USP*, , n. 310, ISSN - 0103-2569, 2007.
- PARETO, V.; POLITIQUE, M. D.; PRESS, A. Paris. 1927.
- POHL, K.; BOCKLE, G.; LINDEN, F. J. v. D. *Software product line engineering: foundations, principles and techniques*. Springer Science & Business Media, 2005.
- R, P. The r project for statistical computing. 2017.  
Disponível em <https://www.r-project.org/>
- RATINAUD, P. Iramuteq: Interface de r pour les analyses multidimensionnelles de textes et de questionnaires [computer software]. *Recuperado de <http://www.iramuteq.org>*, 2017.
- REINERT, M. Alceste une méthodologie d'analyse des données textuelles et une application: Aurelia de gerard de nerval. *Bulletin de méthodologie sociologique*, v. 26, n. 1, p. 24–54, 1990.
- REZENDE, S. O. *Sistemas inteligentes: fundamentos e aplicações*. Editora Manole Ltda. p. 225-248, 2003.
- ROYSTON, J. An extension of shapiro and wilk's w test for normality to large samples. *Applied Statistics*, p. 115–124, 1982.

- RUSSELL, S. J.; NORVIG, P. *Artificial intelligence: A modern approach*. 2 ed. Pearson Education, 2003.
- SANTOS, M. C. B.; MOSCHETTA, W.; COLANZI, T. E.; OLIVEIRAJR, E. A. Otimizando o projeto de arquitetura de linha de produto de software com muitos objetivos: Um estudo exploratório. In: *Workshop de Engenharia de Software Baseada em Busca (WESB)*, 2015.
- SCHMID, K.; VERLAGE, M. The economic impact of product line adoption and evolution. *IEEE Software*, v. 19, n. 4, p. 50, 2002.
- SEI Software Engineering Institute. 2016.  
Disponível em <http://www.sei.cmu.edu/productlines/>
- SMITH, J.; SIMONS, C. L. A comparison of two memetic algorithms for software class modelling. In: *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation (GECCO)*, New York, NY, USA: ACM, 2013, p. 1485–1492.
- VAN VELDHUIZEN, D. A. *Multiobjective evolutionary algorithms: classifications, analyses, and new innovations*. Relatório Técnico, DTIC Document, 1999.
- VAN VELDHUIZEN, D. A.; LAMONT, G. B. *Multiobjective evolutionary algorithm research: A history and analysis*. Relatório Técnico, Citeseer, 1998.
- YOO, S.; HARMAN, M. Pareto efficient multi-objective test case selection. In: *Proceedings of the 2007 International Symposium on Software Testing and Analysis*, ACM, 2007, p. 140–150.
- ZELENY, M.; COCHRANE, J. L. *Multiple criteria decision making*. University of South Carolina Press, 1973.
- ZITZLER, E.; LAUMANN, M.; THIELE, L.; *et al.* Spea2: Improving the strength pareto evolutionary algorithm. In: *Eurogen*, 2001, p. 95–100.
- ZITZLER, E.; THIELE, L.; LAUMANN, M.; FONSECA, C. M.; DA FONSECA, V. G. Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Transactions on Evolutionary Computation*, v. 7, n. 2, p. 117–132, 2003.

---

# Protocolo de Mapeamento Sistemático

---

## A.1 Planejamento

### Questões principais:

Quais são os problemas relacionados ao projeto de software solucionados com a utilização de algoritmos meméticos?

Como os algoritmos meméticos estão sendo utilizados no contexto de linhas de produto de software?

**População:** Projeto de software e projeto de linha de produto de software.

**Intervenção:** Aplicação de algoritmos meméticos na otimização de projeto de linha de produto de software e/ou projeto de software.

**Controle:** Como até o presente momento não existe controle, esse estudo secundário pretende mapear o que existe em um contexto mais amplo visando elucidar os possíveis usos dos algoritmos meméticos em projeto de software ou projeto de linha de produto de software.

**Resultados:** Espera-se encontrar trabalhos relacionados a aplicação de algoritmos meméticos no contexto de projeto de linha de produto de software e/ou projeto de software, informando como esses algoritmos estão sendo utilizados e quais problemas estão resolvendo.

Os resultados encontrados neste mapeamento servirão para:

- Verificar como os algoritmos meméticos são utilizados no contexto de projeto de PLA e/ou de projeto de software;

- Identificar os tipos de benefícios trazidos para possibilitar a comparação com outras estratégias de otimização já existentes em projeto de LPS;
- Extrair os potenciais mecanismos para utilização de algoritmos meméticos em projeto de LPS;
- Embasamento teórico para a dissertação;
- Obter uma visão de alto nível do estado da arte do tema defendido; e,
- Permitir o enriquecimento de conceitos que serão utilizados durante a execução do projeto de dissertação, bem como no trabalho de dissertação.

Além disso, arquitetos e projetistas serão beneficiados com este mapeamento, permitindo uma visão mais abrangente do uso de algoritmos meméticos para otimizar problemas relacionados a projeto de linha de produto de software e/ou projeto de software.

**Crítérios de seleção para Definição de Fontes:** As bibliotecas digitais usadas nesse MS foram selecionadas pela relevância que possuem na área de pesquisa em Ciência da Computação, bem como, pelo número de estudos relacionados que apresentam.

**Estudo nas fontes de pesquisa:** A pesquisa foi executada por meio de buscas utilizando as ferramentas de pesquisa oferecidas em cada biblioteca, além da seleção e classificação dos estudos retornados em cada base de dados. Dessa forma, foram executadas as seguintes etapas:

**Etapa 01:** Busca das palavras chaves e sinônimos nas seções: título, *abstract* e palavras-chaves por meio das ferramentas de filtragem oferecidas pelas bibliotecas digitais selecionadas.

**Etapa 02:** Leitura do título e do *abstract* para identificar a relevância dos estudos encontrados no contexto da pergunta do MS, utilizando os critérios de seleção listados em “Crítérios de Seleção”.

**Etapa 03:** Nos trabalhos selecionados na Etapa 02, será realizada a leitura da introdução e conclusão para sanar possíveis dúvidas quanto contribuição dos trabalhos para o mapeamento.

**Etapa 04:** Estudos categorizados como relevantes serão lidos na íntegra, buscando-se informações relacionadas à pesquisa. Um estudo é categorizado relevante uma vez que ele contenha pelo menos um critério de inclusão e nenhum critério de exclusão.

**Palavras-chave e Sinônimos:**

- *Software Design*
- *Memetic algorithm = (Local search + Genetic algorithm)*
- *Software Product Line = Software Family = Product Line Architecture*

#### **Critérios de Seleção:**

- 1 - **Inclusão:** Trabalhos publicados no período de 2005 a 2015.
- 2 - **Inclusão:** Utilização de algoritmos meméticos em linha de produto de software.
- 3 - **Inclusão:** Utilização de algoritmos meméticos em projeto de software.
- 4 - **Inclusão:** Utilização de algoritmo genético e busca local em linha de produto de software.
- 5 - **Inclusão:** Utilização de algoritmo genético e busca local em projeto de software.
- 6 - **Exclusão:** Trabalhos não disponibilizados na íntegra.
- 7 - **Exclusão:** Trabalhos não relacionados a linha de produto de software ou projeto de software.
- 8 - **Exclusão:** Trabalhos que não estejam disponibilizados em inglês ou português.

## **A.2 Execução**

De acordo com os critérios estabelecidos na seção “Critérios de seleção para Definição de Fontes”, foram selecionadas as seguintes bibliotecas digitais para busca dos trabalhos:

- ACM
- IEEE
- Scopus
- Springer

As *strings* de busca utilizadas nas bibliotecas de busca são apresentadas na sequência. O motivo encontrado para a utilização de duas strings de busca é devido as características interpretação de caracteres utilizadas em cada uma das bibliotecas. A IEEE não necessita da inserção de aspas para identificação de uma *substrings*, dessa forma a primeira *string* foi utilizada nas bibliotecas ACM, Scopus e Springer e a segunda na IEEE.



- (*“memetic algorithm” OR (“Local search” AND “Genetic algorithm”)*) AND (*“software design” OR “software product line” OR “software family” OR “product line architecture”*)
- (*memetic algorithm OR (Local search AND Genetic algorithm)*) AND (*software design OR software product line OR software family OR product line architecture*)

**Tipos de Estudos:** O desenvolvimento de leitura e análise dos trabalhos selecionados como relevantes pela pesquisa encontrou: artigos científicos, monografias, dissertações, teses e relatórios técnicos.

**Seleção Inicial:** Para documentar os estudos encontrados nas buscas realizadas nas bibliotecas digitais, inicialmente foram guardados os seguintes dados: autores, título, palavras-chave, *abstract*, ano e local de publicação, e posteriormente, para os trabalhos selecionados, a obra completa foi armazenada. Os trabalhos retornados nas buscas foram avaliados por duas pessoas. Primeiramente as decisões de inclusão e exclusão serão tomadas e justificadas pelo autor do mapeamento e, em segundo momento tais decisões foram validadas pelo supervisor da pesquisa, professor Dr. Igor Steinmacher.

**Avaliação da Qualidade dos Estudos:** A qualidade dos estudos selecionados foram medida pelos critérios de inclusão e exclusão. Os trabalhos selecionados devem conter no mínimo um critério de inclusão e nenhum critério de exclusão.

**Extração de dados:** Inicialmente somente os dados de identificação dos estudos retornados pelas bibliotecas digitais, foram armazenados, como: autores, título, palavras-chave, *abstract*, ano e local de publicação. Em seguida foi executada a seleção dos estudos justificando a razão pela qual cada um dos estudos foi categorizado por determinado critério de inclusão ou exclusão. Em um terceiro momento, para os trabalhos selecionados, foi feito resgate dos estudos na íntegra. Os problemas encontrados e as soluções propostas com a utilização dos algoritmos meméticos, bem como métricas, funções objetivos e operadores de vizinhança foram anotados. Os dados iniciais servirão para realização de análises quantitativas, já os trabalhos lidos na íntegra serão utilizados na análise qualitativa da mapeamento sistemático.

**Resultados Esperados e Sumarização:** Ao fim do mapeamento sistemático se espera realizar tanto uma classificação quantitativa, apresentando as perspectivas relacionadas a quantidade de estudos encontrados, períodos de maior publicação, autores relacionados a esse foco de estudo, referências mais utilizadas, entre outras classificações, e também, realizar uma classificação qualitativa dos resultados encontrados proporcionando

ou não a resposta positiva as perguntas do mapeamento, bem como, trazendo conceitos que auxiliarão o trabalho de dissertação.

---

<i>B</i>
----------

## Documentos utilizados no Exp 03

---

### B.1 Documentos

Na sequência são apresentados os documentos utilizados na aplicação do Exp 07. Ao todo foram disponibilizados seis documentos além do vídeo instrucional. O vídeo está disponível no link: <https://www.youtube.com/watch?v=pwnx4oINqb8>.

Os documentos 1, 4 e 6 variam para cada LPS e o documento 5 varia para cada par de funções objetivo.

## Termo de Adesão a Estudo Experimental

### "Avaliação experimental do uso de algoritmos meméticos na otimização de projeto de arquitetura de linha de produto de software"

Declaro estar ciente de participar da execução de um estudo experimental, denominado Avaliação experimental do uso de algoritmos meméticos na otimização de projeto de arquitetura de linha de produto de software, a ser coordenado por **João Choma Neto (DIN-UEM)** sob a orientação da **Prof. Dra. Thelma Elita Colanzi Lopes (DIN-UEM)**. Neste estudo utilizarei a especificação da linha de produto de software (LPS) Arcade Game Maker e a abordagem de gerenciamento de variabilidades SMarty para avaliação de projeto baseado em busca para arquitetura de linha de produto de software. Deverei ainda preencher um questionário sucinto declarando minha formação, minha experiência com a notação UML e com a abordagem de LPS. Declaro estar ciente de que os resultados coletados a meu respeito serão confidenciais e de que não receberei nenhum tipo de ônus pela participação, com exceção ao aprendizado de novas técnicas/tecnologias contribuindo para a minha formação profissional.

Acesse o link:

<https://goo.gl/forms/3GokACST0ajY1hQh2>

## Termo de Adesão a Estudo Experimental

*"Avaliação experimental do uso de algoritmos meméticos na otimização de projeto de arquitetura de linha de produto de software"*

Declaro estar ciente de participar da execução de um estudo experimental, denominado *Avaliação experimental do uso de algoritmos meméticos na otimização de projeto de arquitetura de linha de produto de software*, a ser coordenado por **João Choma Neto (DIN-UEM)** sob a orientação da **Prof. Dra. Thelma Elita Colanzi Lopes (DIN-UEM)**. Neste estudo utilizarei a especificação da linha de produto de software (LPS) Banking System e a abordagem de gerenciamento de variabilidades SMarty para avaliação de projeto baseado em busca para arquitetura de linha de produto de software. Deverei ainda preencher um questionário sucinto declarando minha formação, minha experiência com a notação UML e com a abordagem de LPS. Declaro estar ciente de que os resultados coletados a meu respeito serão confidenciais e de que não receberei nenhum tipo de ônus pela participação, com exceção ao aprendizado de novas técnicas/tecnologias contribuindo para a minha formação profissional.

Acesse o link:

<https://goo.gl/forms/2Jbsmkw5Fip6NLSm2>

## Termo de Adesão a Estudo Experimental

*"Avaliação experimental do uso de algoritmos meméticos na otimização de projeto de arquitetura de linha de produto de software "*

Declaro estar ciente de participar da execução de um estudo experimental, denominado Avaliação experimental do uso de algoritmos meméticos na otimização de projeto de arquitetura de linha de produto de software, a ser coordenado por **João Choma Neto (DIN-UEM)** sob a orientação da **Prof. Dra. Thelma Elita Colanzi Lopes (DIN-UEM)**. Neste estudo utilizarei a especificação da linha de produto de software (LPS) Mobile Media e a abordagem de gerenciamento de variabilidades SMarty para avaliação de projeto baseado em busca para arquitetura de linha de produto de software. Deverei ainda preencher um questionário sucinto declarando minha formação, minha experiência com a notação UML e com a abordagem de LPS. Declaro estar ciente de que os resultados coletados a meu respeito serão confidenciais e de que não receberei nenhum tipo de ônus pela participação, com exceção ao aprendizado de novas técnicas/tecnologias contribuindo para a minha formação profissional.

Acesse o link:

<https://goo.gl/forms/OU6L0fYytkqkhOY52>

## Questionário de Caracterização de Participante em Estudo Experimental

*“Avaliação experimental do uso de algoritmos meméticos na otimização de projeto de arquitetura de linha de produto de software”*

ID do Participante

Nas perguntas a seguir, quando duas ou mais alternativas forem válidas, marque a alternativa que mais se aplica ao seu caso.

### 1. Qual o seu nível de formação?

- |                                     |                                   |
|-------------------------------------|-----------------------------------|
| <input type="checkbox"/> Graduando  | <input type="checkbox"/> Graduado |
| <input type="checkbox"/> Mestrando  | <input type="checkbox"/> Mestre   |
| <input type="checkbox"/> Doutorando | <input type="checkbox"/> Doutor   |

### 2. Em qual setor atua?

- |   |                                    |   |
|---|------------------------------------|---|
| <input type="checkbox"/> Acadêmico (ensino) | <input type="checkbox"/> Estudante | <input type="checkbox"/> Industrial (empresarial) |
|---|------------------------------------|---|

### 3. Qual o nome da empresa/universidade em que atua?

.....

### 4. Quanto tempo possui de experiência na área em que atua ?

..... meses ou ..... anos

### 5. Qual a sua experiência com a notação UML com relação aos diagramas de casos de uso e de classes?

- Eu **nunca** modeliei um software usando a UML.
- Minha experiência com a notação UML é básica.**  
Eu modelo software somente no nível dos elementos mais comuns da UML como classes e herança.
- Minha experiência com a notação UML é moderada.**  
Eu modelo software no nível dos elementos da opção anterior, além de: polimorfismo, associação (uni e bi-direcionais), dependência, agregação e composição em classes.
- Minha experiência com a notação UML é avançada.**  
Eu modelo software que exige a utilização de todos os elementos de diagramas de classes, além de outros diagramas da UML como, por exemplo, diagramas de comunicação, sequência, e componentes.

**6. Qual a sua experiência com relação à abordagem de Linha de Produto de Software (LPS) e Gerenciamento de Variabilidade?**

[ ] Eu **nunca** ouvi falar a respeito de LPS.

[ ] **Já lí**, de forma superficial, algo a respeito de LPS.

[ ] **Minha experiência com LPS é básica.**

Eu conheço os seguintes conceitos da abordagem: ciclo de desenvolvimento de LPS e suas atividades (engenharia de domínio e engenharia de aplicação). Porém, **não tenho experiência com gerenciamento de variabilidades.**

[ ] **Minha experiência com LPS é moderada.**

Eu conheço os conceitos da opção anterior, e com relação ao gerenciamento de variabilidades, eu sei o conceito de pontos de variação, variantes e os seus relacionamentos, além dos conceitos de resolução de variabilidades e tempos de resolução (*design time*, *link time*, *runtime*, entre outros).

[ ] **Minha experiência com LPS é avançada.**

Eu conheço os conceitos da opção anterior, além de alguns processos existentes de desenvolvimento de LPS (FODA, PLP, PLUS, PuLSE, entre outros). Com relação ao gerenciamento de variabilidades, eu sei os conceitos da opção anterior, além de: modelos de resolução; abordagens existentes para o gerenciamento de variabilidades, e representação de variabilidades (usando a UML, modelos de características, entre outras).



## I. Linha de Produto de Software

Uma linha de produto de software (LPS) corresponde a um conjunto de sistemas de *software* que compartilham características (*features*) comuns e gerenciáveis que satisfazem a necessidade de um segmento particular ou de uma missão. Este conjunto de sistemas é denominado também, família de produtos. Os membros da família são produtos específicos desenvolvidos de maneira sistemática a partir da instanciação de uma infraestrutura comum de uma LPS, chamada núcleo de artefatos.

O núcleo de artefatos é formado por um conjunto de características comuns (similaridades) e características variáveis (variabilidades). As variabilidades podem estar associadas a diferentes níveis de abstração, como a descrição da arquitetura, o código fonte, etc., e auxiliam na geração de produtos específicos distintos em um mesmo domínio e, desta forma, diminuem o custo e o tempo de desenvolvimento, reduzem riscos e perdas, além de reduzirem o *time to market* e justificarem o retorno de investimento (ROI).

Uma das formas de se implementar características de LPS é tratando cada uma delas como um interesse presente na família. Desta forma, pode-se focar na melhor maneira para desenvolver e modularizar o referido interesse/característica. Vale ressaltar que além da presença de características, outros interesses podem estar presentes. É comum a presença de interesses não funcionais, tais como persistência, *logging*, tratamento de exceções, dentre outros. Os interesses podem ser mapeados em artefatos da LPS usando estereótipos UML, como pode ser visto na Figura 1 em que a interface `ISaveScore` está associada ao interesse `save` que é uma característica da LPS e algumas de suas operações também estão associadas a esta característica. Além disso, a operação `savelnRanking` está associada ao interesse `ranking` que é uma característica e ao interesse não funcional `logging`.



Figura 1 – Exemplo de Interface com Interesses associados usando estereótipos UML

O gerenciamento de variabilidades é uma atividade muito importante no gerenciamento de uma LPS. Em síntese variabilidade é a forma como os membros de uma família de produtos podem se diferenciar entre si, ou seja, é o que permite distinguir os diversos produtos de uma LPS.

A variabilidade é descrita por pontos de variação e variantes:

- **Ponto de variação:** Um local específico de um artefato em que uma decisão de projeto ainda não foi tomada, ou seja, foi adiada;
- **Variante:** Corresponde a uma alternativa de projeto para resolver uma determinada variabilidade.
- **Restrições entre variantes:** define os relacionamentos entre duas ou mais variantes para que seja possível resolver um ponto de variação ou uma variabilidade.

A aplicação destes conceitos é apresentada na Figura 2.

<sup>1</sup> Adaptação e complementação do material gentilmente cedido por Anderson Marcolino e Edson Oliveira Junior (DIN/UEM).

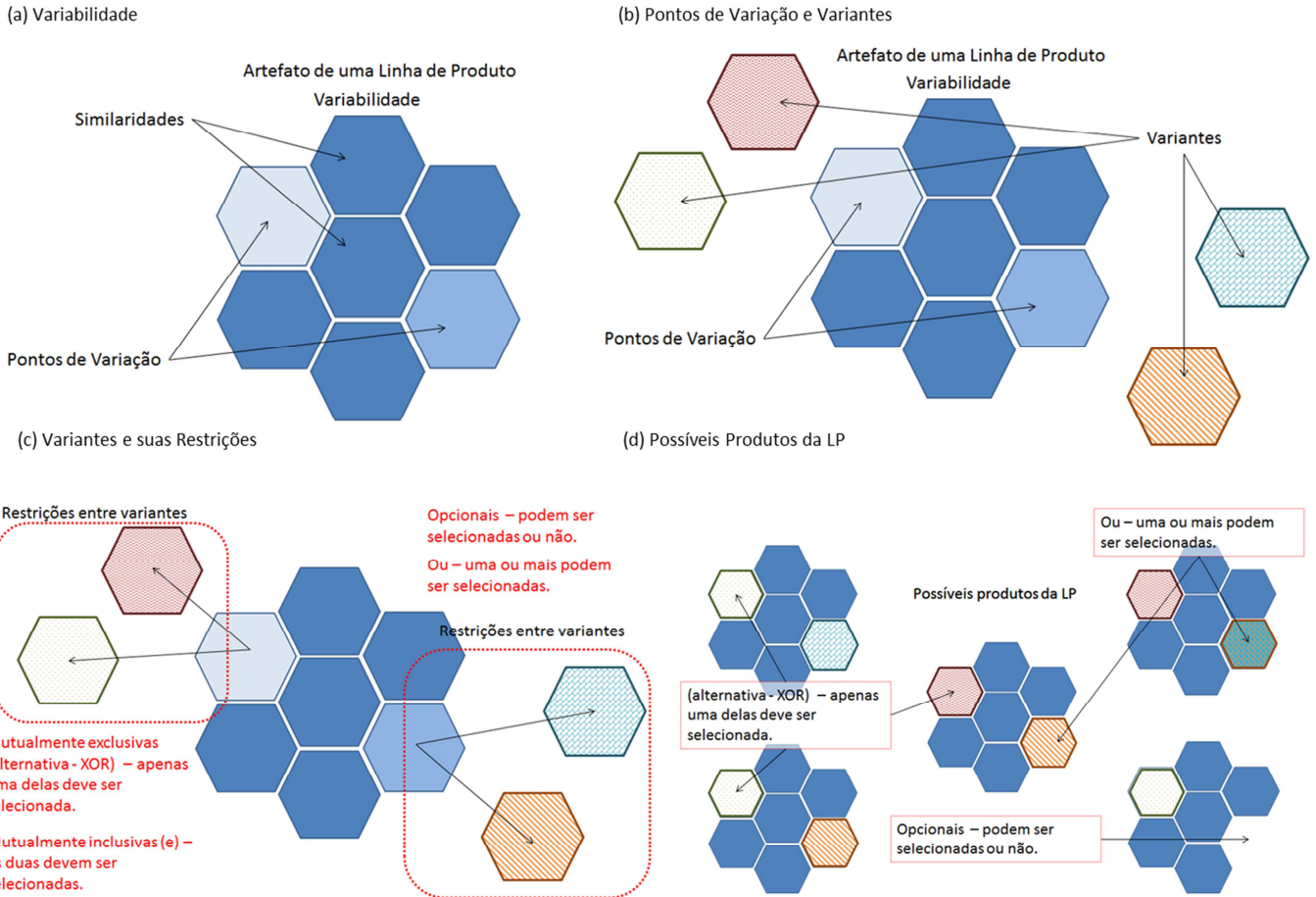


Figura 2 – Exemplo dos Conceitos de Variabilidade, Pontos de Variação, Variantes e Restrições entre Variantes.

## II. Abordagem SMarty para Gerenciamento de Variabilidades

A maioria das abordagens desenvolvidas para auxiliar no gerenciamento de variabilidades envolvem diversos conceitos e modelos de representação. A abordagem SMarty apresentada a seguir possui como base a UML – *Unified Modeling Language*. Esta abordagem utiliza diversos conceitos da UML como o uso de estereótipos e diagramas. Entre os itens que a SMarty contempla, temos os que seguem na Tabela I.

Tabela I – Visão Geral SMarty

Item	Identificação
<b>Baseada em UML</b>	SMarty utiliza os modelos UML, meta atributos, etc., como forma de representação da LPS e de suas variabilidades.
<b>Perfil</b>	SMarty apresenta um perfil específico que é formado por estereótipos e meta atributos, geralmente derivados de uma linguagem de modelagem, como a UML.
<b>Processo</b>	O processo contempla a sistematização da utilização de um perfil para o gerenciamento de variabilidades, guiando o usuário no uso das definições do perfil.
<b>Estereótipos</b>	Estereótipos, como os da UML, são um padrão de mecanismo de extensão e são usados para distinguir diferentes tipos de elementos modelados. Em LPS são ferramentas úteis para identificar variabilidade, seus pontos de variação, variantes e outros itens necessários ao seu gerenciamento, tal como interesses associados aos elementos arquiteturais. SMarty usa estereótipos específicos padrões para representar variabilidades em todos os modelos UML.
<b>Diretrizes</b>	São os passos sistematizados, definidos no processo, que permitem a aplicação facilitada do perfil da abordagem a que corresponde.

## II.1 Estereótipos e Diretrizes

Nesta seção são apresentados os estereótipos para aplicação em diagrama de classes, existentes no perfil da abordagem SMarty por meio da Tabela II, em seguida são apresentados exemplos do uso destes, seguidos pelas diretrizes para cada tipo de modelo.

Tabela II – Estereótipos da Abordagem SMarty para Classes.

Estereótipos Abordagem SMarty (aplicáveis também aos demais modelos da abordagem)		
Estereótipo	Utilização	Exemplo
<<variationPoint>>	Representa o local em que ocorre uma variabilidade. Um ponto de variação está sempre associado a uma ou mais variantes.	Figura 3
<<mandatory>>	A variante estará obrigatoriamente presente na configuração de qualquer produto da linha de produto.	Figura 3
<<optional>>	A variante pode ou não estar presente na configuração de um produto da linha de produto. Variantes opcionais também podem ou não estar associadas a um ponto de variação.	Figura 5
<<alternative_OR>>	Estão sempre associadas aos pontos de variação. Pelo menos uma das variantes deverá ser escolhida para resolver o ponto de variação, ou seja, para estar presente na configuração de um produto da linha de produto.	Figura 3
<<alternative_XOR>>	Estão sempre associadas aos pontos de variação. Somente uma das variantes deverá ser escolhida para resolver o ponto de variação.	-
<<variability>>	Indica uma variabilidade existente em um modelo UML.	Figuras 3 e 4
<<requires>>	Indica um relacionamento de dependência (em UML) entre variantes no qual a variante dependente (origem da dependência) só existirá em uma configuração se a variante relacionada (destino da dependência) existir.	Figura 4
<<mutex>>	Indica um relacionamento de dependência (em UML) entre variantes no qual a variante dependente (origem da dependência) só existirá em uma configuração se a variante relacionada (destino da dependência) obrigatoriamente não existir. São conhecidas como variantes mutuamente exclusivas.	Figura 4

### II.1.1 Exemplos

Na Figura 3 observamos a aplicação da abordagem SMarty, e seus elementos. Passamos a analisar cada um deles, bem como as diretrizes presentes no processo da SMarty, que auxiliam sua utilização em outras LPSs:

A classe **AlgoritmoOrdenacao** identifica uma classe obrigatória (<<mandatory>>) e representa também um ponto de variação (<<variationPoint>>), com três variantes. Estas variantes estão descritas no elemento comentário, relacionado a classe, por meio do *Tagged Value* (**variants**). As três variantes desta classe são **OrdenacaoPorTroca**, **OrdenacaoPorSelecao** e **OrdenacaoPorInsercao**. Todas estas são estereotipadas como <<alternative\_OR>>, o que indica o tipo de restrição para tais variantes, neste caso, significa que ao menos uma ou todas elas podem solucionar o ponto de variação.

**OrdenacaoPorTroca**, **OrdenacaoPorSelecao** e **OrdenacaoPorInsercao**, além de variantes, são, por sua vez, pontos de variação (<<variationPoint>>), e assim, cada uma delas apresenta um comentário, que descreve as suas variantes (**variants**), bem como o nome da mesma (**name**). Neste caso, todas as variantes são marcadas como <<alternative\_OR>> e, como anteriormente, uma delas, ao menos, deve ser selecionada ou todas.

A classe **ProgramaPrincipalOrdenacao**, representa uma classe obrigatória, portanto é marcada como <<mandatory>>, e estará presente em todos os produtos desta LPS.

A classe **ElementoOrdenacao**, também é obrigatória (<<mandatory>>) e representa um ponto de variação (<<variationPoint>>). Ela está ligada a um comentário com o estereótipo <<variability>>, que identifica os dados da variabilidade, que é nomeada, por exemplo, de "sorting element" e possui duas classes variantes (**variants**): **ElementoNumerico** e **ElementoString**, marcadas como variantes alternativas <<alternative\_OR>>, onde, ambas podem ser selecionadas, ou ao menos uma.

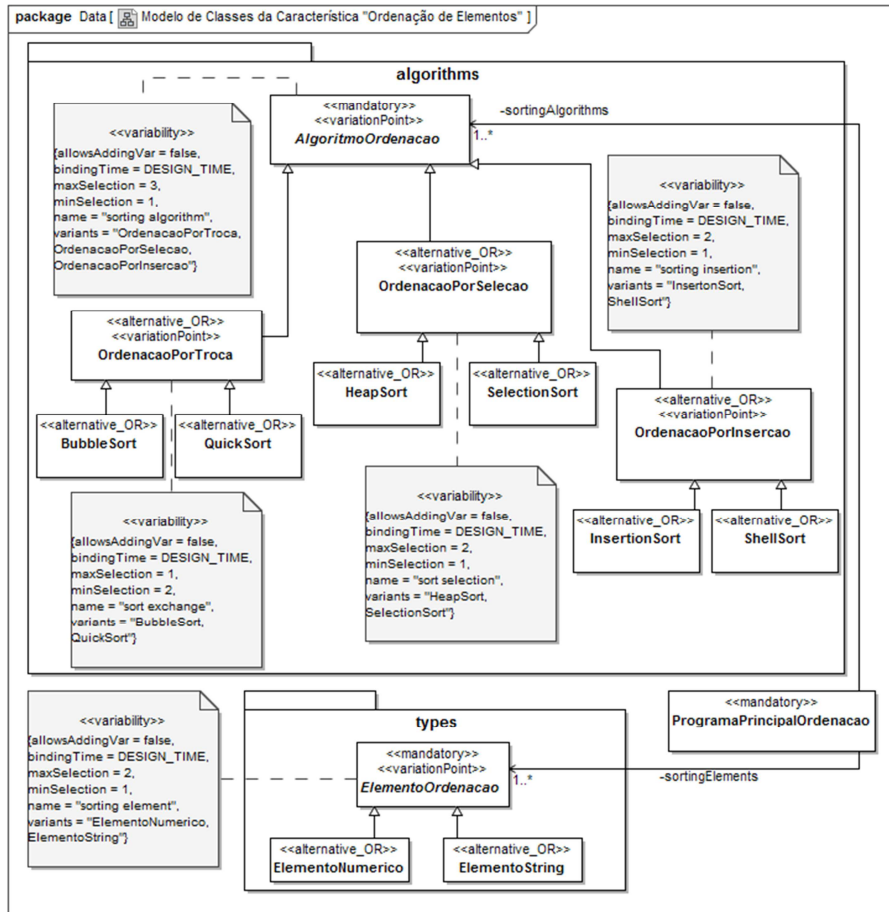


Figura 3 – Exemplo de Modelo de Variabilidade em Diagrama de Classes com a Abordagem SMarty.

Desta forma, as variabilidades são identificadas por meio do comentário UML, estereotipada com `<<variability>>`. Nestes comentários estão contidos os meta atributos que seguem:

- **Name**: nome da variabilidade;
- **minSelection**: a quantidade mínima de variantes a serem selecionadas;
- **maxSelection**: a quantidade máxima de variantes a serem selecionadas;
- **bindingTime**: em qual momento a variabilidade será resolvida;
- **allowsAddingVar**: se permite incluir novas variantes para resolver o ponto de variação; e
- **variants**: quais as variantes para resolver o ponto de variação (casos de uso ligados ao ponto de variação).

Estas notas são inseridas em todas as variabilidades.

**Diretrizes para Identificação e Representação de Variabilidade** - Para a criação e classificação de variabilidades, as seguintes diretrizes presentes no processo da abordagem, sugerem ser seguidas:

**D.3.1** Variabilidades com variantes opcionais possuem multiplicidade ***minSelection = 0*** e ***maxSelection = 1***;

**D.3.2** Variabilidades com variantes exclusivas possuem multiplicidade ***minSelection = maxSelection = 1***;

**D.3.3** Variabilidades com variantes inclusivas possuem multiplicidade ***minSelection = 1*** e ***maxSelection = size(variants)*** em que ***size(x)*** é uma função que retorna a quantidade de elementos da coleção ***x***;

**D.3.4** O valor ***bindingTime*** deve ser definido escolhendo-se um dos valores da classe de enumeração ***BindingTime***, que são: DESIGN\_TIME, LINK\_TIME, COMPILE\_TIME, RUNTIME;

**D.3.5** O valor ***booleano*** do atributo ***allowsAddingVar*** deve ser analisado de acordo com a possibilidade de manter o ponto de variação aberto (***true***) ou fechado (***false***); e

**D.3.6** O valor da coleção ***variants*** é o conjunto formado pelas instâncias das variantes associadas ao ponto de variação ou variabilidade.

**Diretrizes para Diagrama de Classes** - As diretrizes especificadas para auxiliar na identificação das variabilidades em diagramas de classes são expressas abaixo:

**D.2.2** Em modelos de classes, pontos de variação e suas variantes são identificadas nos seguintes relacionamentos:

- a) **generalização**, os classificadores mais gerais são os pontos de variação, enquanto os mais específicos são as variantes;
- b) **realização de interface**, os "suppliers" (especificações) são os pontos de variação e as implementações (clientes) são as variantes;
- c) **agregação**, as instâncias tipadas com losangos não preenchidos são os pontos de variação e as instâncias associadas são as variantes; e
- d) **composição**, as instâncias tipadas com losangos preenchidos são os pontos de variação e as instâncias associadas são as variantes.

**D.2.4** Elementos de modelos de classes, relacionados à associações nas quais os seus atributos ***aggregationKind*** possuem valor ***none***, ou seja, não representam nem agregação nem composição, sugerem variantes obrigatórias ou opcionais. Na Figura 3, a classe **ProgramaPrincipalOrdenacao**, é um exemplo de variante obrigatória.

Elementos de modelos de casos de uso relacionados aos mecanismos de extensão e de pontos de extensão sugerem pontos de variação com variantes associadas, as quais podem ser inclusivas ou exclusivas;

**D.2.3** Em modelos de caso de uso relacionadas com a associação de inclusão (<<include>>) ou associados a atores sugerem variantes obrigatórias ou opcionais;

**D.2.5** Variantes que, ao serem selecionadas para fazer parte de um produto, exigem a presença de outra(s) determinada(s) variante(s) devem ter seus relacionamentos de dependência marcados com o estereótipo <<requires>>;

**D.2.6** Variantes mutuamente exclusivas para um determinado produto devem ter seus relacionamentos de dependência marcados com o estereótipo <<mutex>>.

**D.2.7** Componentes formados por classes com variabilidades são marcados com o estereótipo <<variable>>.

Na Figura 4, que representa fragmento de um diagrama de classes, nele notas o uso dos estereótipos para identificar a dependência entre classes, bem como a seleção mutualmente exclusiva.

A classe **ElementoNumerico** requer a presença da classe **QuickSort**, para que possa ser incluída no produto, já a classe **ElementoString** restringe que a classe **InsertionSort** não seja inserida no

produto, para que possa ser selecionada, ou seja, se selecionada a classe **ElementoString** como variante para o ponto de variação **ElementoOrdenacao**, a classe **InsertionSort**, que é ponto de variação de outra classe, não pode ser selecionada. No caso da classe **ElementoNumerico**, ser selecionada, a classe **QuickSort**, deverá ser selecionada como variante para o ponto de variação a qual pertence.

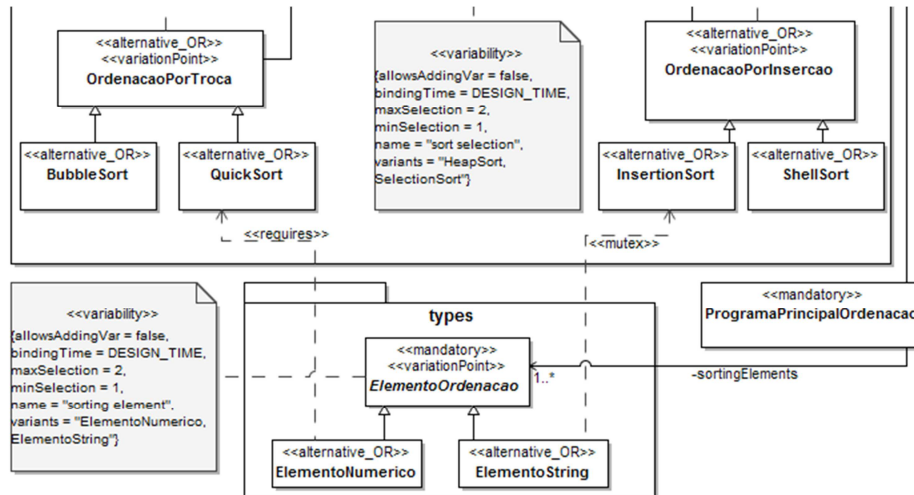


Figura 4 – Exemplo de Identificação de Variabilidade em Classes – *requires* e *mutex*.

A Figura 5 representa a aplicação do estereótipo `<<optional>>`. A classe **SaveGame** é opcional, e assim é marcada como uma variabilidade, pelo comentário da UML – quando opcional a classe pode ou não estar inserida no produto da LPS.

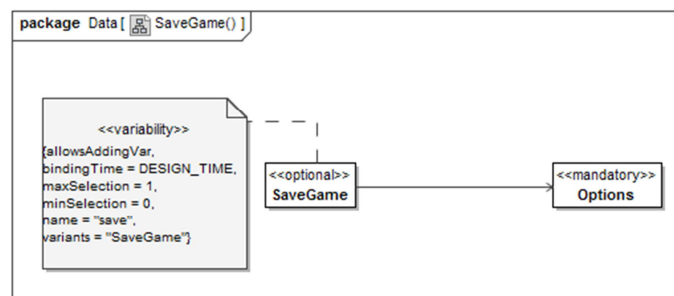


Figura 5 – Exemplo de Identificação de Variabilidade em Classes – *optional*.

# Arcade Game Maker:

## Descrição Geral da Linha de Produto

---

### I. Identificação

A linha de produto (LPS) Arcade Game Maker (AGM) produz uma série de jogos arcade, ou seja, produtos com um ou mais jogos (SEI, 2013). Cada jogo é jogado por um único jogador que controla, parcialmente, os objetos que se movem. O objetivo é marcar pontos acertando obstáculos estáticos. Os jogos vão desde aqueles com obstáculos baixos até obstáculos altos e estão disponíveis para uma variedade de diferentes plataformas.

### II. Similaridades e Variabilidades

Nesta seção são apresentadas as similaridades da LPS, ou seja, os aspectos comuns a todos os produtos desta LPS, bem como as variabilidades, que representam os aspectos que diferem de um produto em relação ao outro.

#### II.1 Similaridades

- Todos os jogos possuem elementos;
  - todo jogo possui um conjunto de *Sprites* (seção II.3);
  - todo jogo possui um conjunto de *Rules* (seção II.3);
- Todos os jogos envolvem movimentação;
- Os elementos dos jogos com os quais o jogador interage estão inseridos em um retângulo com um tamanho específico, bem como possui pontos para o funcionamento do jogo; e,
  - Todos os jogos são acessados via menu principal e possuem menus específicos que dão acesso as mesas destes.

#### II.2 Variabilidades

- Os elementos dos jogos podem ou não ter um par de elementos que reagem a uma ação.
- Os elementos dos jogos, de acordo com o jogo a ser usado, podem apresentar elementos que não se movem, bem como podem apresentar elementos que se movem, podendo ainda possuir ambos.
- As paredes de um jogo podem ou não existir;
- Os elementos que se movem em um jogo, sempre possuirão uma velocidade determinada;
- Os elementos que se movem em um jogo são um *puck* ou/e *paddle*;
- **Tipos de Regras:** é a maior diferença entre os jogos. Algumas regras estão relacionadas às leis da física (gravidade, colisões, etc) e podem ser aplicáveis a múltiplos jogos. Outras regras estão especificamente relacionadas a um jogo e podem ser usadas em todas as implementações do jogo, mas não se aplicam a outros jogos; e,
  - **Tipos de Movimentação:** em alguns jogos a movimentação é inerente à operação do jogo. Isto acontece periodicamente e é orientada pelo tempo. Em outros jogos, o jogador escolhe e inicia a movimentação, sendo ações dirigidas pelo ator.

#### II.3 Conceitos Importantes

- **Sprite:** são os elementos do jogo que os jogadores veem e com os quais eles interagem.
- **Rule:** são as regras que regem as ações dos jogos. Por exemplo, um jogo pode ter uma regra em que um objeto em movimento ao colidir com um objeto estático deve obedecer às leis da física.

### III. Casos de Uso

UC1) **Check Previous Best Score:** verifica e apresenta o melhor score registrado anteriormente.

- a) Ator seleciona a opção CHECK PREVIOUS BEST SCORE do Menu do sistema. Sistema pede para fornecer o nome do arquivo, lê o arquivo e retorna o score à caixa de diálogo.
- b) Ator seleciona OK na caixa de diálogo para continuar. Sistema retorna ao estado anterior.

UC2) **Save Score:** salva a pontuação corrente do jogador.

- a) Ator seleciona SAVE SCORE no Menu do sistema. Sistema pede um nome de arquivo ao ator (cria um novo se não existe), escreve o score no arquivo e retorna ao estado pré-salvo do jogo.

UC3) **Save Game:** salva o jogo em andamento.

- a) Ator seleciona a opção SAVE no Menu do sistema. Sistema permite ao ator especificar um nome de arquivo e, em seguida, escreve os dados do jogo no arquivo especificado e retorna ao estado pré-salvo do jogo.

UC4) **Install Game:** instala o jogo escolhido.

- a) Ator escolhe o instalador do jogo para ser executado. Sistema apresenta uma caixa de diálogo para escolher o diretório em que serão armazenados os arquivos do jogo.
- b) Ator escolhe o diretório. Sistema armazena os arquivos no diretório escolhido.

UC5) **Exit Game:** encerra o jogo em andamento.

- a) Ator seleciona EXIT no Menu do sistema. Sistema apresenta a caixa de diálogo para salvar ou sair do jogo.
- b) Ator salva o jogo. Sistema salva e sai ou cancela saída do jogo.
- c) Sistema retorna à ação suspensa.

UC6) **Uninstall Game:** remove o jogo selecionado.

- a) Ator escolhe a opção UNINSTALL do Menu do sistema. Sistema apresenta uma caixa de diálogo ao ator.
- b) Ator seleciona o diretório do jogo a ser removido. Sistema exclui os arquivos do diretório e apresenta uma caixa de diálogo de remoção concluída.
- c) Ator seleciona a opção OK da caixa de diálogo. Sistema fecha a caixa de diálogo.

UC7) **Play Selected Game:** ator seleciona o jogo e inicia a sua execução.

- a) Ator seleciona a opção PLAY a partir do Menu. Sistema inicializa o jogo e apresenta o GameBoard.
- b) Ator clica com o botão esquerdo e inicia o jogo. Sistema inicia a ação do jogo.
- c) Ator clica com o botão esquerdo ou usa o teclado para enviar comandos. Sistema responde aos comandos.
- d) Ator responde à caixa de diálogo Won/Lost/Even clicando com o botão esquerdo. Sistema retorna o GameBoard para ser inicializado.
- e) A qualquer instante o ator pode selecionar EXIT a partir do Menu.

UC8) **Play Bowling:** inicia o jogo Bowling.

- a) Ator seleciona PLAY do Menu do sistema. Sistema inicializa o jogo e apresenta o GameBoard.
- b) Ator clica com o botão esquerdo para jogar. Sistema inicia a ação do jogo.
- c) Ator repete as ações a seguir 10 vezes mais um lance bônus (opcional)
- d) Ator posiciona o mouse e clica com o botão esquerdo para lançar a Ball pela Alley (pista). Sistema move a Ball pela Alley usando um algoritmo randômico. Se há colisões com os Pins, o sistema move os Pins segundo as leis físicas de colisão. Sistema conta o número de Pins derrubados. Sistema computa o score.

UC9) **Play Brickles:** inicia o jogo Brickles.

- a) Ator seleciona PLAY do Menu do sistema. Sistema inicializa o jogo e apresenta o GameBoard.
- b) Ator clica com o botão esquerdo para iniciar o jogo. Sistema inicia a ação do jogo.
- c) Ator usa o botão esquerdo ou o teclado para enviar comandos ao jogo. Sistema move o Paddle horizontalmente, seguindo o rastro do mouse. A cada movimento do Puck, o sistema verifica se houve colisão com outro objeto. Se o Puck colide com o Ceiling (teto) ou com uma Wall o Puck volta para a área de jogo. Se o Puck colide com o Floor, ele deixa de existir. Se o número máximo de Pucks não foi alcançado, um novo é criado, senão a caixa de diálogo Lost é apresentada. Se o Puck colide com um Brick, a ação a ser tomada depende do Brick. Se for o último Brick, a caixa de diálogo Won é apresentada.
- d) Ator responde à caixa de diálogo Won/Lost clicando com o botão esquerdo. Sistema retorna o GameBoard ao estado inicializado e pronto para jogar.



UC10) **Play Pong:** inicia o jogo Pong.

- a) Ator seleciona PLAY do Menu do sistema. Sistema inicializa o jogo e apresenta o GameBoard.
- b) Ator clica com o botão esquerdo para iniciar o jogo. Sistema inicia a ação do jogo.

UC11) **Animation Loop:** executa as ações de animação dos jogos.

- a) Sistema gera periodicamente sinais e os envia para o jogo.
- b) Sistema move todos os objetos passo-a-passo de acordo com os seus algoritmos de movimentação.
- c) Sistema verifica se há colisões e executa os algoritmos de colisão dos objetos.

UC12) **Initialization:** inicializa o jogo selecionado e apresenta o GameBoard.

- a) Sistema cria as instâncias-padrão para as classes requeridas.
- b) Sistema entra no estado READY.

A Figura anexa apresenta o diagrama de casos de uso da LPS AGM. Nesse diagrama percebe-se a identificação de todas as variabilidades contidas na LPS usando a abordagem SMarty.

## IV. Características (Features)

Há nove características (*features*) presentes na LPS AGM que podem ser mapeadas nos elementos arquiteturais como interesses usando estereótipos. Nesse caso, os interesses foram estipulados observando as características descritas no modelo de características apresentado na Figura 1.

- Play: iniciar uma partida e jogar algum dos tipos de jogos disponíveis no produto da LPS;
- Save: ações relacionadas a salvar uma partida de um determinado jogo;
- Collision: quando se tratar de ações relacionadas a colisão de objetos no jogo;
- Movement: quando se tratar de ações relacionadas a movimento;
- Bowling: itens e regras relativos exclusivamente ao jogo Bowling;
- Brickles: itens e regras relativos exclusivamente ao jogo Brickles;
- Pong: itens e regras relativos exclusivamente ao jogo Pong;
- Pause: ação relacionada a pausar uma partida de jogo em andamento;
- Configuration: responsáveis por configurações como instalação;

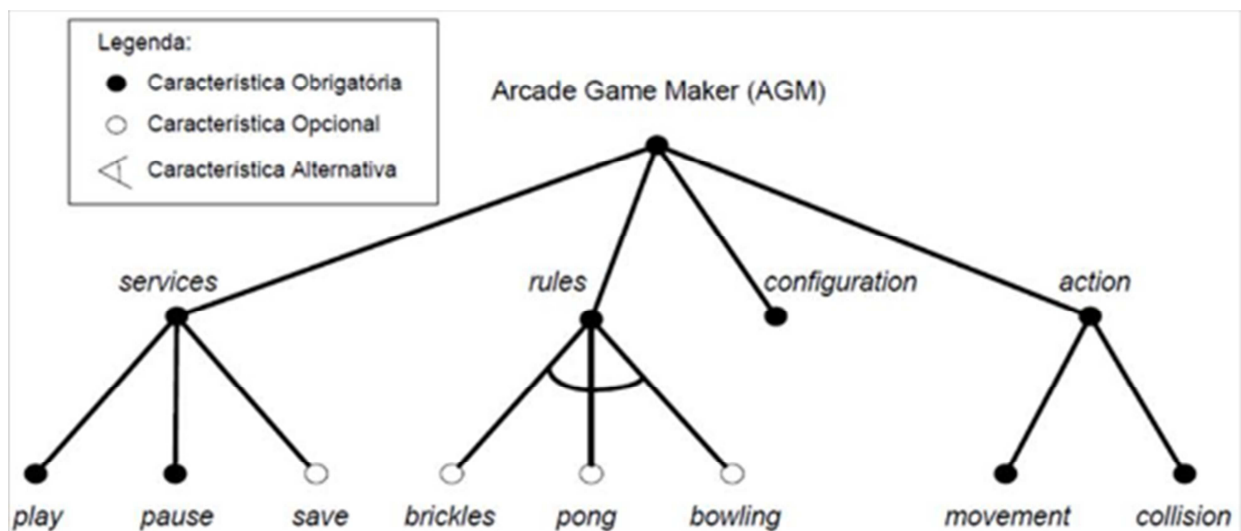
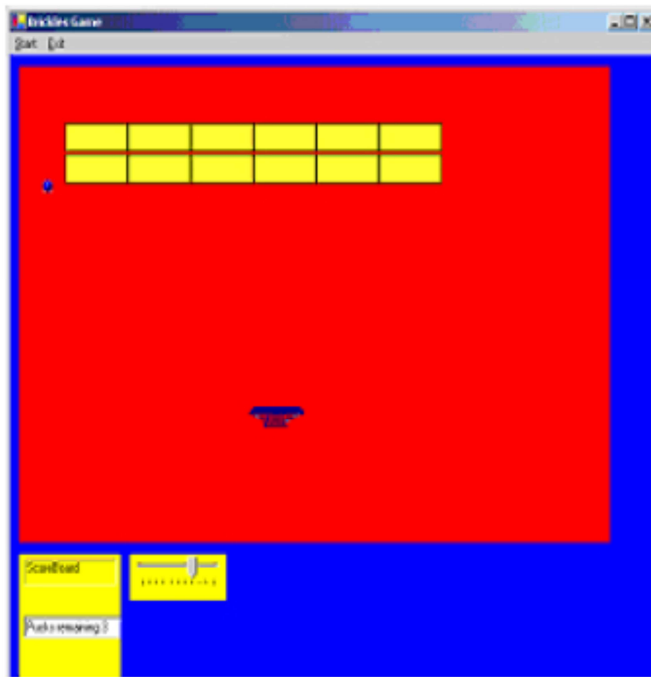


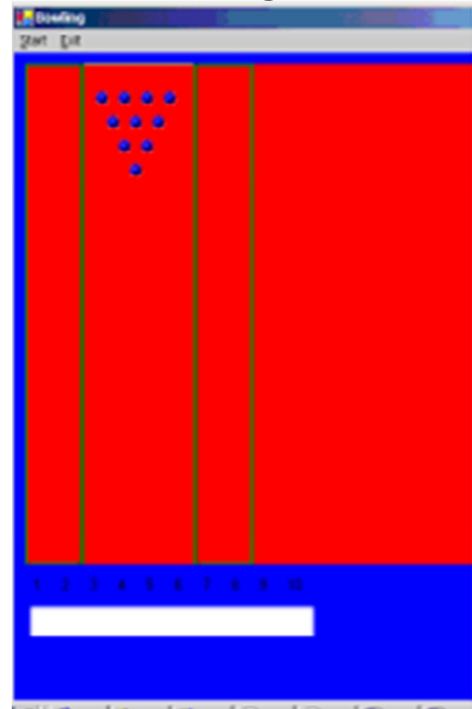
Figura 1: Modelo de Características da LPS AGM

## V. Telas dos Jogos

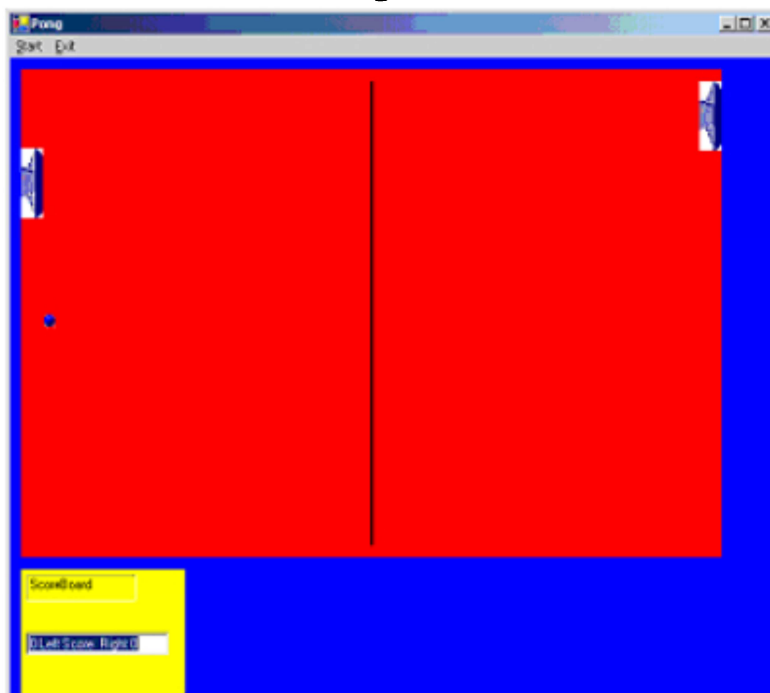
**Brickles**



**Bowling**



**Pong**



## **Referências**

SEI, Software Engineering Institute. *Arcade Game Maker Pedagogical Product Line*. Disponível em: <http://www.sei.cmu.edu/productlines/ppl/>. Acesso em: 03/07/2013. 2013.

# Banking System:

## Descrição Geral da Linha de Produto

---

### I. Identificação

Banking System é uma linha de produto de software construída com objetivo de gerenciar sistemas bancários. Um banco possui diversos caixa automáticos distribuídos e conectados a um servidor central. Ao usar a máquina ATM, um cliente pode retirar dinheiro de uma conta corrente ou de poupança, consultar o saldo de uma conta ou transferir fundos de uma conta para outra. Uma transação somente é iniciada através da utilização de um cartão magnético e uma senha denominada PIN.

### II. Descrição de Casos de Uso

Nome do caso de uso: **Validate PIN.**

Resumo: O sistema valida o PIN do cliente.

Ator: Cliente ATM.

Pré-requisito: ATM está ocioso, exibindo uma mensagem de boas-vindas.

Sequência principal:

1. O cliente insere o cartão ATM no leitor de cartões.
2. Se o sistema reconhecer o cartão, ele lê o número do cartão.
3. O sistema solicita o PIN do cliente.
4. O cliente introduz o PIN.
5. O sistema verifica a data de validade do cartão e se o cartão foram reportados como perdidos ou roubados.
6. Se o cartão for válido, o sistema verifica se o PIN introduzido pelo utilizador corresponde ao PIN do cartão mantido pelo sistema.
7. Se os números PIN coincidirem, o sistema verifica quais contas são acessíveis com o cartão ATM.
8. O sistema exibe as contas de clientes e solicita ao cliente tipo de transação: retirada, consulta ou transferência.

Sequências alternativas:

Etapa 2: Se o sistema não reconhecer o cartão, o sistema cartão.

Etapa 5: Se o sistema determinar que a data do cartão expirou, o sistema confisca o cartão.

Etapa 6: Se o sistema determinar que o cartão foi relatado como perdido ou roubado, o sistema confisca o cartão.

Passo 7: Se o PIN introduzido pelo cliente não corresponder ao número PIN esta placa, o sistema solicita novamente o PIN.

Etapa 7: Se o cliente inserir o PIN incorreto três vezes, o sistema confisca o cartão.

Etapas 4-8: Se o cliente entra em Cancelar, o sistema cancela a transação e ejeta o cartão.

Pós-condição: O PIN do Cliente foi validado.

Nome do caso de uso: **Withdraw Funds**

Resumo: O cliente retira um montante específico de fundos de um conta bancária.

Ator: Cliente ATM.

Dependência: Inclua o caso de uso do PIN de validação.

Pré-requisito: ATM está ocioso, exibindo uma mensagem de boas-vindas.

Sequência principal:

1. Inclua o caso de uso do PIN de Validar.
2. O Cliente seleciona Retirada, insere o valor e seleciona a número da conta.
3. O sistema verifica se o cliente tem fundos suficientes na conta e se o limite diário não será excedido.
4. Se todas as verificações tiverem êxito, o sistema autoriza a dispensa de dinheiro.
5. Sistema dispensa o montante em dinheiro.
6. O sistema imprime um recibo mostrando o número da transação, tipo de transação, montante retirado e saldo da conta.
7. O sistema ejeta o cartão.
8. O sistema exibe a mensagem Bem-vindo.

Sequências alternativas:

Etapa 3: Se o sistema determinar que o número da conta é inválido, ele exibe uma mensagem de erro e ejeta o cartão.

Passo 3: Se o sistema determinar que não existem fundos suficientes na conta do cliente, então ele exibe um pedido de desculpas e ejeta o cartão.

Etapa 3: Se o sistema determinar que o máximo permitido diariamente excedido, ele exibe um pedido de desculpas e cartão.

Passo 5: Se o ATM está sem fundos, o sistema exibe um pedido de desculpas, ejeta cartão, e desliga o caixa eletrônico.

Pós-condição: Os fundos de clientes foram retirados.

Nome do caso de uso: **Query Account**

Resumo: O cliente recebe o saldo de uma conta bancária válida.

Ator: Cliente ATM.

Dependência: Inclua o caso de uso do PIN de validação.

Pré-requisito: ATM está ocioso, exibindo uma mensagem de boas-vindas.

Sequência principal:

1. Inclua o caso de uso do PIN de Validar.
2. O cliente seleciona Consulta, insere o número da conta.
3. O sistema lê o saldo da conta.
4. O sistema imprime um recibo que mostra o número da transação, tipo e saldo da conta.
5. O sistema ejeta o cartão.
6. O sistema exibe a mensagem Bem-vindo.

Sequência alternativa:

Etapa 3: Se o sistema determinar que o número da conta é inválido, ele reproduz uma mensagem de erro e ejeta o cartão.

Pós-condição: A conta do cliente foi consultada.

Nome do caso de uso: **Transfer Funds**

Resumo: O cliente transfere fundos de uma conta bancária válida para outro.

Ator: Cliente ATM.

Dependência: Inclua o caso de uso do PIN de validação.

Pré-requisito: ATM está ocioso, exibindo uma mensagem de boas-vindas.

Sequência principal:

1. Inclua o caso de uso do PIN de Validar.
2. O Cliente seleciona Transferir e insere o valor, da conta e conta.
3. Se o sistema determinar que o cliente tem fundos na conta de, ele executa a transferência.
4. O sistema imprime um recibo que mostra o número da transação, tipo, quantidade transferida e saldo da conta.
5. O sistema ejeta o cartão.
6. O sistema exibe a mensagem Bem-vindo.

Banking System: Descrição Geral da Linha de Produto.

Sequências alternativas:

Etapa 3: Se o sistema determinar que o número da conta é inválido, ele exibe uma mensagem de erro e ejeta o cartão.

Etapa 3: Se o sistema determinar que o número da conta é inválido, ele exibe uma mensagem de erro e ejeta o cartão.

Passo 3: Se o sistema determinar que não existem fundos suficientes na cliente da conta, ele exibe um pedido de desculpas e ejeta o cartão.

Pós-condição: Os fundos de clientes foram transferidos.

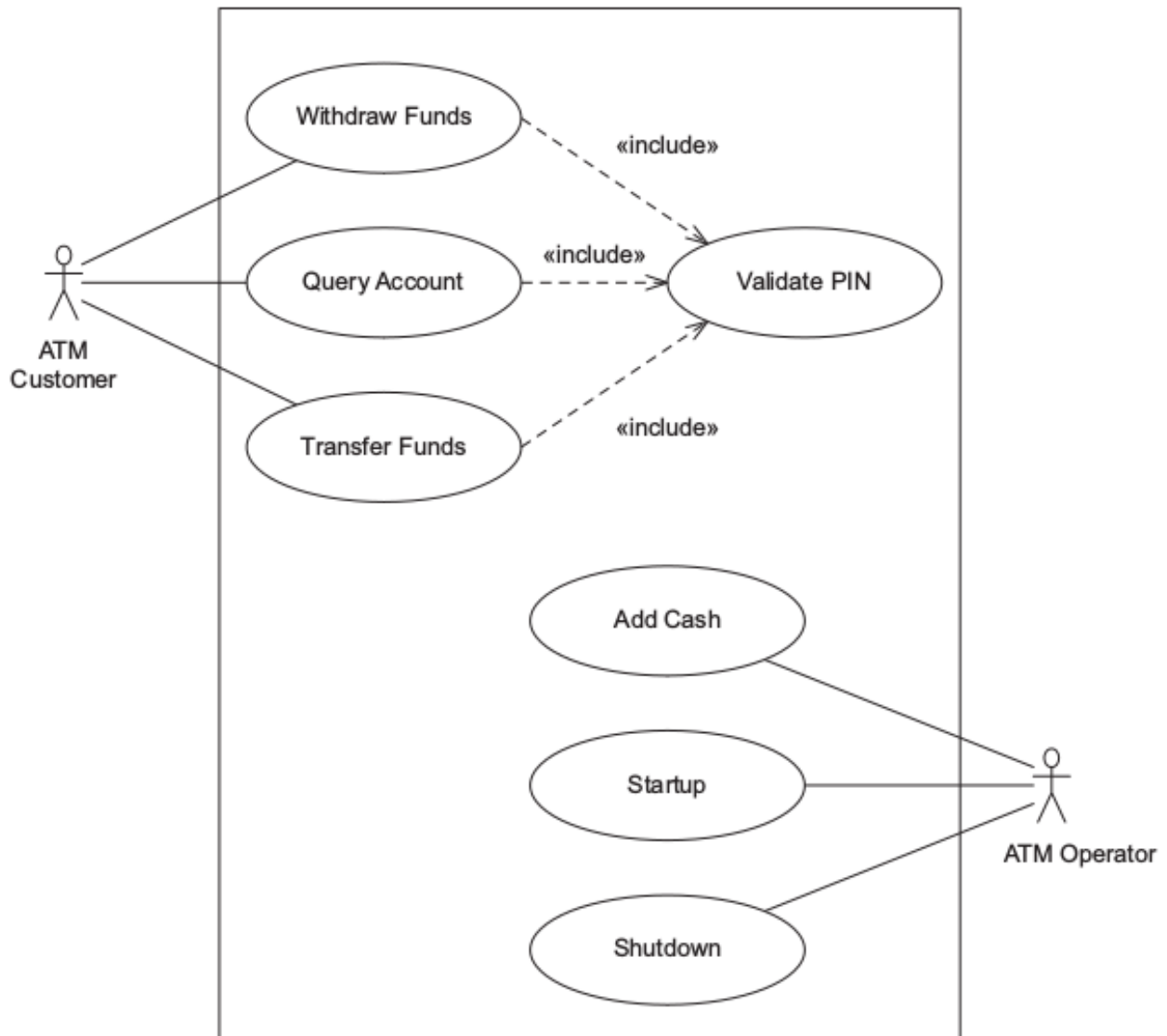


Figura 1: Diagrama de Casos de Uso.

## Referências:

GOMAA, Hassan. Software modeling and design: UML, use cases, patterns, and software architectures. Cambridge University Press, 2011.

# Mobile Media: Descrição Geral da Linha de Produto

## I. Identificação

Mobile Media (YOUNG, 2005) é uma LPS composta por aplicações que manipulam músicas, vídeos e fotos para dispositivos móveis, como celulares e palm tops. Ela provê suporte para gerenciar (criar, excluir, visualizar, executar, enviar) diferentes tipos de mídia.

A Mobile Media surgiu da extensão de uma LPS já existente denominada Mobile Photo (YOUNG, 2005), por meio da inserção de novas propriedades multimídia, como manipulação de vídeos e músicas, que somente podem ser realizados em alguns tipos de aparelhos. De certa forma, pode-se dizer que a inserção das características opcionais e alternativas a determinados aparelhos caracterizou o surgimento da Mobile Media.

## II. Similaridades e Variabilidades

Nesta seção são apresentadas as similaridades da LPS que são os aspectos que devem estar em todos os produtos desta LPS, bem como as variabilidades que a compõem. As Tabelas 1 e 2 apresentam as características da Mobile Photo. A Tabela 1 contém as características básicas, inerentes a todas as aplicações, enquanto a Tabela 2 apresenta as características variáveis da Mobile Photo. É importante ressaltar que a utilização dessas características está diretamente relacionada aos aparelhos, que podem ou não apresentar suporte para uma determinada funcionalidade. A Figura 1 apresenta um resumo dos requisitos da Mobile Media.

Tabela 1: Características básicas da Mobile Photo (YOUNG, 2005)

Característica	Descrição
Criar Álbum de Fotos	Permite ao usuário definir novos álbuns de fotos para armazenar categorias de fotos no dispositivo. A persistência da informação do álbum é realizada utilizando RMS (J2ME Record Management System).
Armazenar Foto	Gerenciar a conversão e persistência dos arquivos de foto para o sistema de arquivos do dispositivo utilizando RMS.
Adicionar/Deletar Foto	Permite ao usuário excluir fotos permanentemente do dispositivo, ou adicionar novas fotos a álbuns definidos.
Rotular Foto	Permite ao usuário determinar um texto para uma foto. Os rótulos aparecerão na lista de exibição, e podem ser utilizados para uma futura funcionalidade relacionada à busca.
Visualizar Foto	Mostra uma foto selecionada na tela do dispositivo.

Tabela 2: Variabilidades da Mobile Photo (YOUNG, 2005)

Característica	Descrição
Enviar Foto via SMS	Permite a um usuário enviar uma foto para outro via Short Messaging Service.
Relacionar Foto com Registro na Agenda	Permite ao usuário associar um registro na sua lista de contatos com a foto do álbum.
Mostrar Foto nas Chamadas Recebidas	Intercepta chamadas recebidas e mostra a foto associada ao contato.
Tocar Melodia nas Chamadas Recebidas	Intercepta chamadas recebidas e toca uma melodia personalizada para aquele contato.

- R1.** O protótipo da aplicação é um álbum apropriado para celulares, PDAs ou BlackBerrys, desde que o tal álbum não seja limitado para eles.
- R2.** A aplicação oferece um conjunto de funcionalidades listadas a seguir.
- R3.** Um usuário pode criar uma nova mídia, incluindo fotos, vídeos ou sons.
- R4.** Usuário pode então adicionar mídia criada a um álbum.
- R5.** Sistema deve armazenar mídia no formato RMS.
- R6.** O usuário pode também criar um novo álbum, permitindo o armazenamento no dispositivo de categorias relacionadas de figuras, sons ou vídeos.
- R7.** O usuário somente pode criar mídias se estiver autenticado.
- R8.** O sistema deve persistir as informações do álbum no formato RMS.
- R9.** Usuário pode também excluir uma mídia ou um álbum.
- R10.** O sistema deve verificar se o álbum a ser excluído está vazio.
- R11.** Para excluir arquivos de mídia ou álbuns, o usuário deve estar autenticado.
- R12.** Usuários devem estar aptos a listar álbuns, permitindo a visão de todos os álbuns contidos no dispositivo.
- R13.** O sistema deve permitir ao usuário listar mídia incluída no dispositivo.
- R14.** Usuário pode colocar rótulos para os arquivos de mídia.
- R15.** Rótulos aparecem na tela de exibição, e podem ser usados para futuras funcionalidades de busca.
- R16.** Usuários podem executar mídias que estão no sistema de arquivos do dispositivo.
- R17.** O usuário pode enviar uma mídia via, incluindo foto, vídeo ou áudio.
- R18.** Usuário pode ligar uma mídia com um registro na agenda, permitindo a um usuário associar um registro na lista de contatos com uma mídia no álbum.
- R19.** O sistema também permite a configuração de uma mídia para recepção de chamadas, interceptando uma chamada recebida e mostrando a foto ou executando o áudio associado àquele chamador.
- R20.** O usuário pode ouvir o toque de uma chamada, interceptando chamadas recebidas e executando um toque customizado para aquele chamador.
- R21.** O usuário pode configurar as mídias favoritas, e incluí-las em um conjunto de favoritos.
- R22.** Depois de configurar a mídia favorita, o usuário pode obter a lista de mídias favoritas.

Figura 1: Requisitos da Mobile Media (SANTOS *et al.*, 2008)

### III. Casos de Uso

**UC1:** Log in

**Categoria:** mandatory

**Resumo:** Usuário insere login e senha e o sistema valida os dados.

**Pré-Condições:** Usuário deve estar registrado no sistema.

**Cenário Principal:**

1. Sistema apresenta a tela com opção de login e senha.
2. Usuário insere login e senha.
3. Sistema verifica se os dados digitados são válidos e permite que o usuário execute ações restritas.

**Alternativos:**

- Usuário pode cancelar operação a qualquer momento.

4. Sistema rejeita os dados, uma vez que login ou senha estavam incorretos. Retorna ao passo 1.

**Pós-Condições:** Usuário foi autenticado ou houve falha na autenticação, ou ainda o usuário cancelou a operação.

**UC2:** Send Media

**Categoria:** optional

**Resumo:** Usuário seleciona e envia uma determinada mídia para outro dispositivo.

**Pré-Condições:** Deve haver mídias armazenadas no sistema.

**Cenário Principal:**

1. Usuário seleciona uma mídia. A mídia selecionada pode ser música, vídeo ou foto.
2. Usuário aciona a opção de enviar a mídia.
3. Sistema transfere o arquivo via SMS.

**Alternativos:**

- Usuário pode cancelar a operação em qualquer uma de suas ações.

**Pós-Condições:** Usuário enviou uma mídia para outro dispositivo ou a operação foi cancelada.

### **UC3: Manage Album**

**Categoria:** mandatory

**Resumo:** Usuário realiza o controle geral dos cadastros dos álbuns. Engloba as operações de criação, exclusão e listagem dos álbuns.

**Pré-Condições:** Para a operação de exclusão, o sistema deve possuir pelo menos um álbum cadastrado, e o álbum a ser excluído deve estar vazio. Para a listagem, devem existir mídias armazenadas no sistema.

#### **Cenário Principal:**

1. Usuário seleciona a opção de gerenciamento de álbuns.
2. Sistema apresenta as operações disponíveis para o gerenciamento do tipo de mídia selecionado.
3. Usuário seleciona uma operação.
  - Create Album: ver item 3.1.
  - Delete Album: ver item 3.2.
  - List Album: ver item 3.3.
4. Sistema conclui a operação.

#### **Alternativos:**

- Usuário pode cancelar a operação em qualquer uma de suas ações.

#### **Itens:**

##### **3.1. Create Album**

- 3.1.1. Usuário acessa a opção de criar um álbum.
- 3.1.2. Sistema apresenta a tela para criação de um álbum.
- 3.1.3. Usuário entra com os dados do álbum.
- 3.1.4. Sistema verifica se já não existe algum álbum com as características informadas pelo usuário.
- 3.1.5. Sistema cria o álbum.

#### **Alternativos desse item:**

- Usuário pode cancelar a operação em qualquer uma de suas ações.  
3.1.4. Já existe um álbum com as configurações estabelecidas. Sistema apresenta uma mensagem de erro para o usuário e cancela a operação.

##### **3.2. Delete Album**

- 3.2.1. Usuário aciona a opção de excluir um álbum.
- 3.2.2. Sistema exige confirmação do usuário sobre a exclusão permanente do álbum.
- 3.2.3. Usuário confirma a exclusão.
- 3.2.4. Sistema verifica se o álbum selecionado está vazio.
- 3.2.5. Sistema exclui o álbum selecionado pelo usuário.

#### **Alternativos desse item:**

- Usuário pode cancelar a operação em qualquer uma de suas ações.  
3.2.3. Usuário não confirma a exclusão, e a operação é cancelada.  
3.2.4. Sistema verifica que o álbum selecionado possui mídias. Dessa forma, alerta o usuário que o álbum não está vazio e cancela a operação.

##### **3.3. List Album**

- 3.3.1. Usuário seleciona a opção de visualizar todos os álbuns presentes no dispositivo.
- 3.3.2. Sistema apresenta uma lista com todos os álbuns disponíveis.

#### **Alternativos desse item:**

3.3.2. Não existem álbuns criados. O sistema exibe uma mensagem de erro ao usuário e cancela a operação.

**Pós-Condições:** Sistema realizou a operação desejada ou a ação foi cancelada.

---

### **UC4: Manage Media**

**Categoria:** mandatory



**Resumo:** Usuário realiza o controle geral dos cadastros de mídia. Engloba as operações de criação, exclusão e listagem dos registros.

**Pré-Condições:** Para a criação, o usuário deve estar logado no sistema e deve haver espaço em memória suficiente para o armazenamento da mídia criada. Para a exclusão, usuário deve estar logado e deve haver mídias armazenadas no sistema. Para a listagem, devem existir mídias armazenadas no sistema.

**Cenário Principal:**

1. Usuário seleciona a opção de gerenciamento de mídias.
2. Sistema apresenta os tipos de mídia disponíveis. Os tipos são foto, vídeo ou música.
3. Usuário seleciona um tipo de mídia.
4. Sistema apresenta as operações disponíveis para o gerenciamento do tipo de mídia selecionado.
5. Usuário seleciona uma operação.
  - Create Media: ver item 3.1.
  - Delete Media: ver item 3.2.
  - List Media: ver item 3.3.
6. Sistema conclui a operação.

**Alternativos:**

- Usuário pode cancelar a operação em qualquer uma de suas ações.
- 3. Caso a mídia escolhida seja uma foto, <<extend>> Manage Photo. Caso seja um vídeo, <<extend>> Manage Video. Caso seja uma música, <<extend>> Manage Music.

**Itens:**

3.1. Create Media

- 3.1.1. Usuário acessa a opção de criar uma mídia.
- 3.1.2. <<include>> Log in.
- 3.1.3. Sistema apresenta a tela e as opções para a criação da mídia desejada.
- 3.1.4. Usuário utiliza as opções apresentadas para criar uma mídia.
- 3.1.5. Usuário escolhe salvar a mídia criada.
- 3.1.6. Sistema verifica se é possível salvar a mídia.
- 3.1.7. Sistema armazena a mídia criada pelo usuário.

**Alternativos desse item:**

- Usuário pode cancelar a operação em qualquer uma de suas ações.
- 3.1.6. Sistema não consegue encontrar espaço suficiente para armazenamento da mídia criada. Sistema apresenta uma mensagem de erro ao usuário e a operação é cancelada.

3.2. Delete Media

- 3.2.1. Usuário indica a ação de excluir uma mídia.
- 3.2.2. <<include>> Log In.
- 3.2.3. Sistema exige confirmação sobre a exclusão permanente.
- 3.2.4. Usuário confirma sua opção de excluir.
- 3.2.5. Sistema verifica se a mídia excluída não está associada a um contato.
- 3.2.6. Sistema exclui a mídia da memória do dispositivo.

**Alternativos desse item:**

- Usuário pode cancelar a operação em qualquer uma de suas ações.
- 3.2.4. Usuário não confirma a exclusão, e a operação é cancelada.
- 3.2.5. A mídia está sendo utilizada, podendo estar, por exemplo, vinculada a um determinado contato. Neste caso, sistema alerta o usuário sobre a utilização e cancela a exclusão.

3.3. List Media

- 3.3.1. Usuário acessa a opção de listar mídias disponíveis.
- 3.3.2. Sistema exibe uma lista de todas as mídias ordenadas pelas mais vistas.

**Alternativos desse item:**

- 3.3.2. Não existem mídias armazenadas. O sistema exibe uma mensagem de erro ao usuário e cancela a operação.

**Pós-Condições:** Sistema realizou a operação desejada ou a ação foi cancelada.

## **UC5: Manage Favourite Media**

**Categoria:** optional

**Resumo:** Usuário realiza o controle acerca de mídias favoritas. Engloba as operações de definição de mídia favorita e de listagem das mídias favoritas.

**Pré-Condições:** Para a operação de marcar mídia como favorita, deve haver mídias armazenadas no sistema. Para a listagem, devem existir mídias armazenadas no sistema e marcadas como favorita.

### **Cenário Principal:**

1. Usuário seleciona a opção de gerenciamento de mídias favoritas.
2. Sistema apresenta as operações disponíveis para o gerenciamento do tipo de mídia selecionado.
3. Usuário seleciona uma operação.
  - Set Media as Favourite: ver item 3.1.
  - List Favourite Media: ver item 3.2.
4. Sistema conclui a operação.

### **Alternativos:**

- Usuário pode cancelar a operação em qualquer uma de suas ações.

### **Itens:**

#### 3.1. Set Media as Favourite

- 3.1.1. Usuário seleciona a opção de marcar a mídia como favorita.
- 3.1.2. Sistema verifica se a mídia já não está marcada como favorita.
- 3.1.3. Sistema seta a mídia como favorita.

### **Alternativos desse item:**

- Usuário pode cancelar a operação em qualquer uma de suas ações.  
3.1.2. A mídia já estava marcada como favorita. Sistema exibe uma mensagem de erro para o usuário e cancela a operação.

#### 3.2. List Favourite Media

- 3.2.1. Usuário acessa a opção de visualizar suas mídias favoritas.
- 3.2.2. Sistema exibe uma lista com todas as mídias marcadas como favoritas.

### **Alternativos desse item:**

3.2.2. O usuário ainda não marcou nenhuma mídia como favorita. Sistema exibe uma mensagem de erro para o usuário. A operação é cancelada.

**Pós-Condições:** Sistema realizou a operação desejada ou a ação foi cancelada.

---

## **UC6: Nome do Caso de Uso: Play Media**

**Categoria:** mandatory

**Resumo:** Usuário executa uma mídia de acordo com as possibilidades do produto.

**Pré-Condições:** Devem existir mídias no sistema de arquivos do dispositivo.

### **Cenário Principal:**

1. Usuário seleciona uma mídia de acordo com as possibilidades do produto (a mídia pode ser música, foto ou vídeo).
2. Usuário seleciona a opção de executar a mídia.
3. Sistema executa a mídia selecionada pelo usuário.

### **Alternativos:**

- Usuário pode cancelar a operação em qualquer uma de suas ações.

3. Caso a mídia escolhida seja uma foto, <<extend>> Visualize Photo. Caso seja um vídeo, <<extend>> Play Video. Caso seja uma música, <<extend>> Play Music.

**Pós-Condições:** Usuário executou uma mídia ou a operação foi cancelada.

---

#### **UC7: Add Media to Album**

**Categoria:** mandatory

**Resumo:** Usuário seleciona uma mídia e a vincula a um determinado álbum armazenado no sistema.

**Pré-Condições:** Devem existir mídias e álbuns armazenados no sistema.

**Cenário Principal:**

1. Usuário seleciona uma mídia.
2. Usuário aciona a opção de adicionar essa mídia a um álbum.
3. Sistema verifica se a mídia já está vinculada a um álbum.
4. Sistema exibe as opções de álbuns disponíveis.
5. Usuário seleciona um dos álbuns disponíveis.
6. Sistema adiciona a mídia ao álbum selecionado.

**Alternativos:**

- O usuário pode cancelar a operação em qualquer uma de suas ações.
3. Usuário seleciona uma mídia já vinculada a um determinado álbum. Sistema verifica se o usuário deseja criar uma cópia da mídia para o álbum a ser selecionado. Caso o usuário confirme, uma cópia é criada. Caso contrário o sistema move a mídia para o álbum a ser escolhido, eliminando-a do álbum antigo.
  4. Não existe qualquer álbum. O sistema exibe uma mensagem de erro para o usuário e cancela a operação.
  6. A mídia já se encontra no álbum selecionado. O sistema exibe uma mensagem de erro para o usuário e cancela a operação.

**Pós-Condições:** Usuário adicionou uma mídia a um álbum ou a operação foi cancelada.

---

#### **UC8: Link Media with Address Book Entry**

**Categoria:** optional

**Resumo:** Usuário vincula uma mídia a um determinado contato para que ela seja reproduzida quando há uma chamada desse contato.

**Pré-Condições:** Devem existir mídias e contatos armazenados no dispositivo.

**Cenário Principal:**

1. Usuário seleciona uma música armazenada no sistema.
2. Usuário aciona a opção de relacionar essa mídia a um contato.
3. Sistema exibe uma lista com os contatos contidos no dispositivo.
4. Usuário seleciona um determinado contato na lista.
5. Sistema verifica se o contato selecionado já não possui uma mídia vinculada do mesmo tipo da mídia selecionada.
6. Sistema vincula a mídia ao contato selecionado.

**Alternativos:**

- Usuário pode cancelar a operação em qualquer uma de suas ações.
1. A mídia selecionada pode ser uma foto.
  3. Não existem contatos registrados no dispositivo. Sistema apresenta uma mensagem de erro ao usuário e cancela a operação.
  5. O contato selecionado já apresenta uma mídia do tipo da selecionada vinculada. O sistema informa o usuário e exibe a opção de vincular a nova mídia ou manter a que já estava. Retorna ao passo 6.

**Pós-Condições:** Usuário vinculou uma mídia a um contato ou a operação foi cancelada.

---

**UC9: View/Hear Media from Incoming Caller****Categoria:** optional**Resumo:** Ao receber uma chamada, o sistema reproduz a mídia vinculada ao contato.**Pré-Condições:** O contato deve ter alguma mídia vinculada.**Cenário Principal:**

1. Usuário recebe uma chamada de um determinado contato.
2. Sistema verifica se esse contato está vinculado a uma determinada mídia (música ou foto).
3. Sistema intercepta a chamada e reproduz a mídia personalizada.

**Alternativos:**

2. O contato não apresenta nenhuma mídia vinculada. O sistema reproduz o toque e a imagem padrão de chamadas.

**Pós-Condições:** Sistema reproduziu uma mídia personalizada para um contato ou reproduziu a mídia padrão para todos os contatos.**UC10: Label Files****Categoria:** optional**Resumo:** Usuário insere um rótulo para uma determinada mídia, podendo ser vídeo, foto ou música.**Pré-Condições:** Devem existir mídias no sistema de arquivos do dispositivo.**Cenário Principal:**

1. Usuário seleciona uma mídia. A mídia selecionada pode ser música, vídeo ou foto.
2. Usuário acessa a opção de inserir um rótulo para aquela mídia.
3. Sistema verifica se a mídia já possuía um rótulo associado.
4. Sistema apresenta a tela para a inserção do rótulo.
5. Usuário insere um rótulo para aquele arquivo.
6. Sistema verifica se algum arquivo já possui aquele rótulo.
7. Sistema vincula o rótulo ao arquivo.

**Alternativos:**

- Usuário pode cancelar a operação em qualquer uma de suas ações.

3. O arquivo já possuía um rótulo. Sistema informa ao usuário se ele deseja substituir o rótulo existente.
5. Usuário deixou o rótulo em branco. Sistema informa o erro ao usuário. Retorna ao passo 4.
6. Sistema verifica que já existe um arquivo com aquele rótulo e apresenta uma mensagem de erro para o usuário. Retorna ao passo 4.

**Pós-Condições:** Usuário inseriu um rótulo para uma mídia ou a operação foi cancelada.

A Figura 2 apresenta o diagrama de casos de uso da LPS Mobile Media. Nesse diagrama percebe-se a identificação de todas as variabilidades contidas na LPS usando a abordagem SMarty.

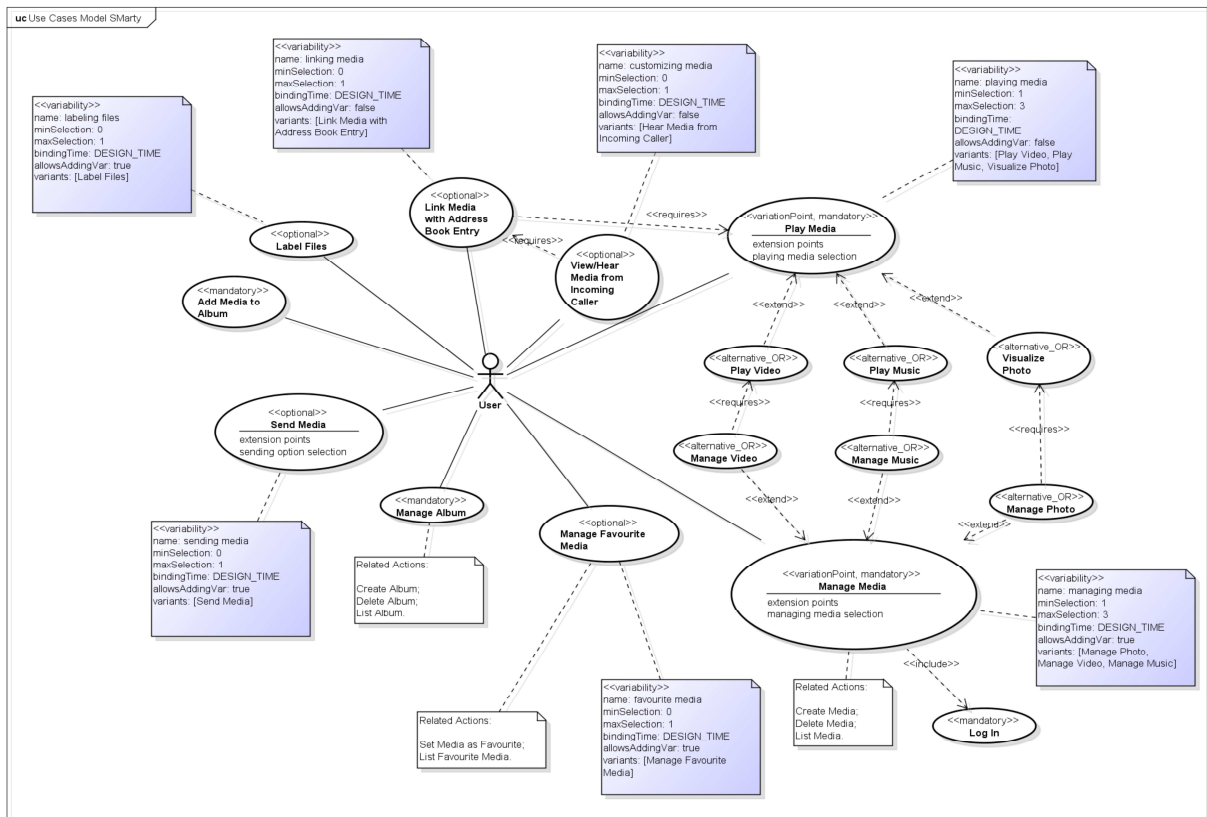


Figura 1: Diagrama de Casos de Uso utilizando Smarty (CONTIERI JUNIOR, 2010)

## IV. Características (*Features*)

Há doze características (*features*) presentes na LPS Mobile Media que podem ser mapeadas nos elementos arquiteturais como interesses usando estereótipos. Elas foram extraídas do modelo de características da LPS apresentado na Figura 4.

- *Media Management*: refere-se ao gerenciamento de mídias;
- *Copy Media*: refere-se à ação de fazer cópias de uma determinada mídia;
- *Create/Delete*: refere-se às operações básicas de mídia como criar ou excluir;
- *Favourites*: itens relacionados ao gerenciamento de mídias favoritas;
- *View/Play Media*: itens relacionados a visualização de quaisquer tipos de mídias;
- *Label Media*: ação referente à nomeação ou renomeação de mídias;
- *SMS Transfer*: ação referente à transferência de mídia;
- *Video*: itens relativos exclusivamente ao tipo de mídia vídeo;
- *Music*: itens relativos exclusivamente ao tipo de mídia música;
- *Photo*: itens relativos exclusivamente ao tipo de mídia foto;
- *Media*: quando se tratar de mais de um tipo de mídia ao mesmo tempo;
- *Link Media*: quando se trata de vincular uma mídia a um contato.

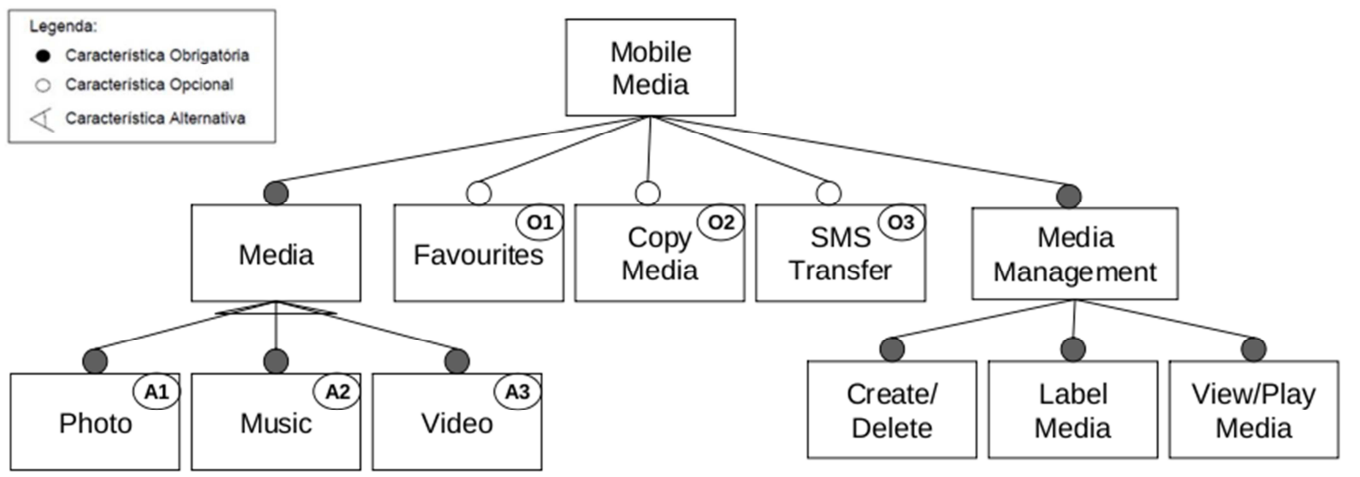


Figura 4: Modelo de Características da LPS Mobile Media (CONTIERI JUNIOR, 2010)

## Referências

CONTIERI JUNIOR, A. C.; **Aplicação de Métricas em Arquiteturas de Linhas de Produto de Software**. 2010. 73 f. Monografia de TCC. Universidade Estadual de Maringá, 2010.

SANTOS, J. P.; et al. Generating Requirements Analysis Models from Textual Requirements, **MARK'08**, Workshop of 16th RE (IEEE International Requirements Engineering Conference), Barcelona, Sep 2008.

YOUNG, T. **Using AspectJ to Build a Software Product Line for Mobile Devices**. MSc Dissertation, University of British Columbia, 2005.

Questionário 01 do experimento qualitativo

Acesse o link:

<https://goo.gl/forms/acYY5wDnrFidpEuL2>

1. Qual a sua opinião com relação à coesão relacional desse projeto de PLA?
2. Qual a sua opinião com relação ao acoplamento desse projeto de PLA?
3. Qual a sua opinião com relação ao tamanho das interfaces desse projeto de PLA?
4. Qual a sua opinião com relação à modularização de características desse projeto de PLA?
5. Qual a melhor alternativa de projeto sob o ponto de vista de modularização de características?
6. Quais propriedades você julga importantes para se avaliar em um projeto de PLA?

Questionário 02 do experimento qualitativo

Acesse o link:

<https://goo.gl/forms/Wrb4FHfwFsP119PE2>

1. Qual a sua opinião com relação à coesão relacional desse projeto de PLA?
2. Qual a sua opinião com relação ao acoplamento entre classes desse projeto de PLA?
3. Qual a melhor alternativa de projeto sob o ponto de vista de coesão relacional?
4. Qual a melhor alternativa de projeto sob o ponto de vista de acoplamento de classes?
5. Qual a melhor alternativa de projeto sob o ponto de vista de modularização de características?
6. Quais propriedades você julga importantes para se avaliar em um projeto de PLA?



## Características da PLA AGM

As características da PLA podem ser identificadas por meio dos esteriótipos a seguir:

play  
persistence  
exception handling  
logging  
action  
rule  
configuration  
service  
save  
bowling  
pong  
brickles  
movement  
collision  
pause  
ranking

## Características da PLA BANK

As características da PLA podem ser identificadas por meio dos esteriótipos a seguir:

user\_interaction  
io  
max\_pin\_attempts  
output  
external\_output\_device  
state\_dependent\_control  
statement  
maintenance  
deposit  
external\_io\_device  
greeting  
external\_user  
german  
spanish  
french  
english

## Características da PLA MOM

As características da PLA podem ser identificadas por meio dos esteriótipos a seguir:

AlbumManagement

Create\_Delete

ViewPlayMedia

SMSTransfer

LabelMedia

LinkMedia

Favourites

MM

Video

Music

Photo

CopyMedia

Media

GUI

---

## Corpus construídos no Exp 03

---

Na sequência são apresentados os três corpus utilizados na ferramenta Iramuteq. Para facilitar a compreensão, a cada bloco de respostas foi inserido a pergunta correspondente.

### C.1 Corpus 1

O Corpus 1 é composto pelas respostas do Questionário 1.

#### 1. Qual a sua opinião com relação à coesão relacional desse projeto de PLA?

\*\*\*\* \*n\_1 De forma geral, não existe forte coesão entre as classes, interfaces ou pacotes. A PLA possui 7 pacotes que poderiam ser simplificados (unidos) com relação as suas interfaces e classes para tornar o projeto mais coeso. Assim, é possível observar 4 pacotes que, por sua vez, modelam a maioria da PLA, que não são muito coesos. Além disso, para cada um dos 3 pacotes restantes, cada um possui apenas uma interface, tornando o projeto coeso mais sem sentido com relação a sua viabilidade. É necessário repensar a modelagem de tal PLA. Um exemplo de problema seria manter dependência da classe CashReader com as interfaces interface1609 e a interface1553, sendo que as quais, não têm responsabilidades muito claras em todo o projeto.

\*\*\*\* \*n\_2 Dado o número de relacionamentos internos de cada pacote e o número de pacotes, entendo que esta PLA tem baixa coesão. Cinco pacotes não possuem relacionamento entre classes e interfaces ou os relacionamentos são com interfaces de outros pacotes.

\*\*\*\* \*n\_3 Pouco coeso, pois dá para fazer uma redistribuição mais compreensível entre as classes dos pacotes (ATMClient). No entanto prejudicaria o acoplamento.

\*\*\*\* \*n\_4 De um modo geral, algumas coisas que não gostei na arquitetura. Vários atributos tem nomes genéricos, attr5 por exemplo. O mesmo acontece com o nome dos pacotes. Isso é ruim e não faz sentido quando falamos de Orientação a Objeto. O nome das classes, operações, métodos e pacotes devem representar o conceito / função / característica que eles mapeiam ou representam. A notação da arquitetura gerada também dificulta alguns entendimentos. Por exemplo, a classe game, é abstrata ? Porque ocorrem atributos repetidos nas classes que herdam dela (Por exemplo o atributo gameCode repete nas subclasses Bowling, Pong e Brickles).

\*\*\*\* \*n\_5 O projeto tem um grande volume de ligações tendo como um dos extremos somente um pacote, o que pode indicar que o mesmo pacote possui muitas responsabilidades, viola pelo menos um dos princípios do SOLID e indica um alto acoplamento e, por consequência, uma baixa coesão.

\*\*\*\* \*n\_6 Na minha opinião a coesão da estrutura apresenta um nível médio de coesão relacional.

\*\*\*\* \*n\_7 Em geral a coesão relacional apresentada é simples. Salvo pelas ligações da classe MediaCtrl, mas não vejo problema nenhum nessas ligações.

\*\*\*\* \*n\_8 A PLA possui coesão alta, já que os pacotes estão associados a poucas características, mantendo a mesma características entre as classes.

\*\*\*\* \*n\_9 Considero nem alta nem baixa a coesão relacional desta PLA, pois somente as classes MediaCtrl e Media possuem mais relacionamentos se comparadas com as outras classes da PLA.

## **2. Qual a sua opinião com relação ao acoplamento desse projeto de PLA?**

\*\*\*\* \*n\_1 O acoplamento de toda a PLA poderia ser melhorado levando em consideração, principalmente, o pacote ATMClient1 que possui 13 classes com dependências, heranças e realizações confusas com relação a suas reais responsabilidades. Isso é consequência de um alto acoplamento. Suas interfaces também estão muito simples e necessitam ser repensadas.

\*\*\*\* \*n\_2 Entendo que esta PLA esteja altamente acoplada com base no número de pacotes, classes e interfaces e dos seus relacionamentos entre si. Note que 3 pacotes não dependem de outros pacotes, mesmo assim o número de relacionamentos com relação ao número de pacotes é alto.

\*\*\*\* \*n\_3 O acoplamento esta ok. Baixo acoplamento, porém é notável que o baixo acoplamento prejudicou a coesão da arquitetura.

\*\*\*\* \*n\_4 Encontrei algumas coisas que me parece inconformidades. Não entendi muito bem porque uma parte da arquitetura é desconexa da outra. Por exemplo, porque o GameBoardMgr não tem relação com o GameBoardCtrl? Porque algumas classes não

se relacionam com outras? A classe jogador estar relacionada com uma subclasse do Package15224Ctr também não parece fazer sentido. Tem interface dentro de pacotes sem nenhum acoplamento. Isso me parece estranho. Isso acontece no Package15137Mgr e no 14949Mgr. Se pensarmos de uma maneira geral a arquitetura inicial que parecia ser projetada com MVC ficou um tanto descaracterizada. Porque o GameBoardGui não tem relações com o GameBoardCtrl e Mgr?

\*\*\*\* \*n\_5 Alto acoplamento e, por consequência, uma baixa coesão. Já especificado na primeira resposta, o alto número de ligações de um mesmo módulo indica alto acoplamento.

\*\*\*\* \*n\_6 Na minha opinião existe um grau alto de acoplamento apresentado no projeto da PLA.

\*\*\*\* \*n\_7 Achei que poderia ser um pouco mais limpo. Exemplo: Se todas as classes que gerenciam o comportamento de uma entidade ficassem no mesmo pacote talvez diminuísse o acoplamento. Desse forma haveria apenas uma interface para acessar os recursos do pacote, diminuindo algumas ligações.

\*\*\*\* \*n\_8 A PLA possui um acoplamento elevado, principalmente devido a uma forte dependência da classe MediaMgr.

\*\*\*\* \*n\_9 Considero alto o acoplamento entre as classes, pois o pacote Media, o pacote que contém a classe Media e o pacote que contém a classe MediaMgr possuem relacionamentos com a maioria dos pacotes presentes na PLA.

### **3. Qual a sua opinião com relação ao tamanho das interfaces desse projeto de PLA?**

\*\*\*\* \*n\_1 As responsabilidades das interfaces são muito simples e possuem poucos métodos realmente úteis. Além disso, seus relacionamentos e dependências poderiam ser modelados de outra maneira para melhorar a coesão e o acoplamento em toda a PLA. Também existem 3 interfaces únicas para 3 pacotes únicos de forma separada. Por que não criar apenas uma pacotes para as 3 interfaces? Além disso, por exemplo, a interface IReceiptPrinter possui apenas um método.

\*\*\*\* \*n\_2 Somente 2 pacotes não possuem interfaces. Entendo que o tamanho das interfaces em função do número de operações parece ser suficiente para reutilização, porém a classe ATMTransactor depende da classe Account, o que parece precisar de uma interface para fornecer os serviços ou ao menos uma classe DAO para Account.

\*\*\*\* \*n\_3 Tamanho das interfaces ok.

\*\*\*\* \*n\_4 Me parece OK. No geral as interfaces apresentam menos de 5 métodos. A maior interface IGameBoardMgt que é a interface gerenciadora do jogo tem a maior quantidade de métodos (10) e faz sentido já que ele gerencia o jogo.

\*\*\*\* \*n\_5 As interfaces são pequenas, o que pode influenciar positivamente no reuso das mesmas e na facilidade de substituições de suas implementações.

\*\*\*\* \*n\_6 Na minha opinião o tamanho das interfaces desse projeto de PLA é alto.

\*\*\*\* \*n\_7 Achei bom.

\*\*\*\* \*n\_8 As interfaces em geral são pequenas, com algumas exceções sendo um pouco maiores, algo que pode afetar um pouco a reusabilidade, mas não drasticamente.

\*\*\*\* \*n\_9 Considero baixo, pois a maioria das interfaces contém entre 1 à 4 operações e a maior (IManageMedia) teve 16 operações, não prejudicando a manutenção e o reuso da PLA.

#### **4. Qual a sua opinião com relação à modularização de características desse projeto de PLA?**

\*\*\*\* \*n\_1 Todas as características do projeto possuem modificadores de acesso públicos. Acredito que esse fato pode impactar no que realmente está encapsulado ou protegido na PLA. Tais fatos tornam as características dessa PLA difíceis de ser modularizadas. Até que ponto tais características e / ou métodos podem estar relacionadas com pacotes, interfaces ou classes nesse projeto? Tudo está acessível a outras interfaces ou classes? Os seus relacionamentos poderiam se utilizar de anotações ou estereótipos mais claros que para estabelecer o que é obrigatório, o que varia ou o que é opcional.

\*\*\*\* \*n\_2 Os pacotes DisplayPromptsClient1 e ATMClient1 são os que possuem menos modularização de características por causa do elevado número de difusão de características e e coesão baseada em características. De forma geral, os demais pacotes possuem alta modularidade. Entendo que os 2 pacotes citados precisam ser mais modularizados, pois representam o maior número de classes interfaces da PLA.

\*\*\*\* \*n\_3 Tive dificuldade de avaliar a modularização na arquitetura. A forma como foi apresentado exemplos de modularização nos slides, não está compatível com a forma que ela deve estar na arquitetura, por este motivo não consegui identificar. No entanto dá para notar que a difusão de características é um pouco alta para algumas classes específicas e coesão baseada em características é baixa pois há pacotes pouco coesos.

\*\*\*\* \*n\_4 Me parece que tem coisas estranhas (não só com a modularização, mas com o projeto em si). Do ponto de vista da modularização, alguns exemplos de coisas estranhas. A maior interface, só é implementada por uma classe, que tem um único atributo com nome genérico. Não dá pra inferir o que isso significa.

\*\*\*\* \*n\_5 O projeto possui funcionalidades que estão devidamente isoladas em pequenos módulos com poucas responsabilidades. Não são todos, mas algumas funcionalidades estão devidamente isoladas em módulos.

\*\*\*\* \*n\_6 Não consegui identificar quais estereótipos estão relacionados às características . Para evitar o prejuízo de fazer uma avaliação equivocada não opinei a respeito da modularização com base em características.

\*\*\*\* \*n\_7 Acho que pode ser mais modularizado. Alguns pacotes poderiam conter toda a lógica da entidade.

\*\*\*\* \*n\_8 Levando em conta o acoplamento elevado e o tamanho das interfaces, a modularização da PLA não é o ideal e poderia ser melhorada.

\*\*\*\* \*n\_9 Considero nem alta nem baixa, pois não são todas as características da PLA contêm alta difusão, interação e coesão.

### **5. Qual a melhor alternativa de projeto sob o ponto de vista de modularização de características?**

\*\*\*\* \*n\_1 Dentre as alternativas analisadas, a melhor alternativa de projeto é a banking.pdf. Tal escolha foi devido ao fato dessa alternativa definir estereótipos e notações para as interfaces e classes do projeto (opcional, mandatory, dentre outros). Acredito que isso torna as características mais consistentes e modulares, propiciando a forte coesão e o baixo acoplamento. Além disso, a quantidade de dependências entre interfaces e classes nesta alternativa de projeto foi minimizada, tornando as características mais consistentes e modulares.

\*\*\*\* \*n\_2 O pacote ATMClient é o mais problemático das PLAs. Porém, acho que VAR\_All\_banking-7294316121 é a que apresenta melhor modularização em termos da distribuição de características nas classes e pacotes em função da DC, EC e CC.

\*\*\*\* \*n\_3 Acredito que a VAR\_All\_banking-8489715735, no entanto não está muito claro para mim.

\*\*\*\* \*n\_4 VAR\_All\_agm-4953622311 eu ficaria entre ela e a AGM. Me incomodou ter um pacote na AGM com muitas classes. Entretanto, as outras arquiteturas criaram pacotes com nomes estranhos, e que não pareciam fazer muito sentido.

\*\*\*\* \*n\_5 VAR\_1\_agm-1137411359.pdf

\*\*\*\* \*n\_6 Pela dificuldade de identificação das características nas PLAs apresentadas, não consigo apontar uma alternativa tendo como base a modularização de características,. Mais levando-se em conta as métricas tradicionais e relacionais acredito que uma alternativa interessante seja a arquitetura apresentada no arquivo VAR\_1\_agm-1137411359.pdf

\*\*\*\* \*n\_7 MobileMedia. Achei que as responsabilidades ficaram melhor divididas em seus pacotes. Fica mais fácil identificar onde as responsabilidades estão.

\*\*\*\* \*n\_8 VAR\_All\_MobileMedia-7172446371, pois apesar de apresentar uma coesão ligeiramente mais baixa, possui um baixo acoplamento.



\*\*\*\* \*n\_9 O projeto VAR\_All\_MobileMedia-3748334128, pois levando em consideração a característica AlbumManagement está difusa em apenas dois pacotes da PLA.

## C.2 Corpus 2

O Corpus 2 é composto pelas respostas do Questionário 2.

### 1. Qual a sua opinião com relação à coesão relacional desse projeto de PLA?

\*\*\*\* \*n\_10

Pude perceber que esse projeto há um número considerável de relacionamento interno no pacote BankingServiceServer1. O pacote ATMClient1 possui um número médio de relacionamentos internos. Já o pacote e DisplayPromptsClient1 não possui nenhum relacionamento interno. Em relação as características, os pacotes ATMClient1 e DisplayPromptsClient são pouco coesos, já os pacotes BankingServiceServer1 e CardServiceServer2 são altamente coesos. Portanto, de maneira geral pude perceber que há um nível de coesão relacional médio para baixo nesse projeto.

\*\*\*\* \*n\_11 Entendo que esse projeto de PLA apresenta as seguintes características: Pacotes que apresentam uma única classe e / ou interface, que são utilizadas / implementadas por elementos externos. Isso pode ser observado nos pacotes Package13C, Package4E e Package86; Pacotes cujas classes não se relacionam, como é o caso do pacote DisplayPromptsClient1, mas que apresentam muitos relacionamentos externos; Pacotes com uma única interface, que é implementada por um elemento externo ao pacote. Isso acontece por exemplo, com o pacote Package41; Pacotes que apresentam um alto número de relacionamentos internos, como é o caso do pacote ATMClient1; Dadas as características apresentadas acima, entendo que a maioria dos pacotes apresentam um número maior de relacionamentos externos, em comparação com o número de relacionamentos internos. Uma exceção é feita ao pacote ATMClient1, que apresenta uma situação oposta. Por conta disso, entendo que a coesão relacional desse projeto de PLA é baixa.

\*\*\*\* \*n\_12 Analisando o número médio de relacionamentos internos, é possível notar que houveram melhorias no sentido de diminuição desse número em alguns pacotes. Porém, em outros, o número permaneceu o mesmo. Isto leva a crer que alguns pacotes já estavam coesos (ex pacote BankingServiceServer1) e outros poderiam (e acredito que foram) ser melhorados (ex. pacote ATMClient1). Com relação ao pacote CardServiceServer2, notei que a classe CardAccount simplesmente sumiu. Apesar de que, com isso, a coesão relacional tenha melhorado nesse pacote, achei estranho não ver os métodos ou atributos dessa classe na classe que ela depende (DebitCard).

\*\*\*\* \*n\_13 Alta coesão.

\*\*\*\* \*n\_14 O pacote GameController achei pouco coeso, a característica de save foi separada em outro pacote mas as características de configuration e de play ficaram na mesma classe e elas não se relacionam. Apesar disso, se tivessem sido separadas, o GameGUI iria depender de mais pacotes, ou seja, foi centralizada a dependência da tela para uma classe, o que é justificável pois estes elementos não são variáveis. Os outros pacotes possuem classes que se relacionam melhor, reúnem características em comum e estão organizadas de uma forma lógica.

\*\*\*\* \*n\_15 Se eu entendi correto, esse projeto está pouco coeso, pois há muito características repetidas em várias classes, dentro do mesmo pacote e entre os pacotes.

\*\*\*\* \*n\_16 Com relação à coesão relacional, esse projeto de PLA é pouco ou fracamente coeso, uma vez que apresentam, em sua maioria, um elevado número médio de relacionamentos.

\*\*\*\* \*n\_17 Achei de acordo, foi de fácil entendimento.

\*\*\*\* \*n\_18 A MM apresenta boa coesão relacional, pois o relacionamento de suas classes e interfaces são claros e fáceis de entender.

## **2. Qual a sua opinião com relação ao acoplamento entre classes desse projeto de PLA?**

\*\*\*\* \*n\_10 É evidente nesse projeto o alto nível de acoplamento, principalmente entre os elementos dos pacotes BankingServiceServer1, CardServiceServer2 e DisplayPromptsClient1. No pacote BankingServiceServer1 o nível de elementos que tem dependências de outras classes chega a ser de 100%, ou seja, todas as classes desse pacote fornecem interfaces para outros pacotes isolados. No pacote ATMClient1 é baixo o número de dependências. Portanto, de uma maneira geral pude perceber que há um nível alto de acoplamento entre as classes desse projeto.

\*\*\*\* \*n\_11 Considerando o relacionamento entre classes e interfaces, é possível observar que apenas os pacotes BankingServiceServer e ATM Client1 apresentam um número de relacionamentos internos igual ou superior ao número de relacionamentos externos. Nos demais pacotes, as classes / interfaces apresentam mais relacionamentos externos que internos. Além disso, foi observado que muitos pacotes apresentam relacionamentos de generalização, realização e dependência. Esses relacionamentos evidenciam um certo acoplamento entre os elementos. Por conta da existência de mais relacionamentos externos que internos, além dos relacionamentos de dependência, generalização e realização, se entende que tal projeto de PLA possui um alto acoplamento.

\*\*\*\* \*n\_12 Eu acho que as classes e as interfaces que estão nos pacotes DisplayPromptsClient1, Package86, Package46 e Package13c estão demasiadamente conectadas gerando

relacionamentos que ao meu ver estavam melhores relacionadas no modelo original. Analisando somente as classes não consegui ver grandes diferenças. As diferenças para mim ficaram mais evidentes quando envolvem as interfaces.

\*\*\*\* \*n\_13 Baixo acoplamento.

\*\*\*\* \*n\_14 Para esse quesito devo destacar que o diagrama parece estar incompleto, a implementação das interfaces da característica play do pacote GameController, que o pacote PlayGameGUI dependeria e que poderia se relacionar também com os pacotes GameBoardCtr, GameBoardMgr e GameMgr não existe. O mesmo vale para implementação da interface do pacote GameMgr. Acredito que essas classes faltantes impactariam fortemente no acoplamento, e sem elas alguns pacotes foram totalmente isolados. Avaliando só as classes apresentadas o acoplamento é baixo, mas o projeto parece quebrado.

\*\*\*\* \*n\_15 Há muito acoplamento entre as classes e interfaces. Muitas características sendo repetida nessas classes.

\*\*\*\* \*n\_16 Com relação ao acoplamento desse projeto de PLA, acredito que ele apresente um grau elevado de acoplamento, uma vez que os relacionamentos entre classes são abundantes.

\*\*\*\* \*n\_17 Bem definido.

\*\*\*\* \*n\_18 A MM apresenta pouco acoplamento, pois a maioria das classes não dependem de outras para funcionar.

### **3. Qual a melhor alternativa de projeto sob o ponto de vista de coesão relacional?**

\*\*\*\* \*n\_10 A PLA banking e a PLA VAR\_28\_banking-7114272521 apresentam um alto nível de coesão relacional, porém na PLA banking pude perceber que a coesão relacional é maior entre as classes, visto que todas as classes possuem algum tipo de relacionamento, já na PLA VAR\_28\_banking-7114272521 algumas classes estão isoladas ou sem nenhum tipo de relacionamento. No meu ponto de vista, diante das PLAs expostas a melhor alternativa de projeto sob o ponto de vista de coesão relacional é a PLA banking.

\*\*\*\* \*n\_11 Entendo que a alternativa nomeada banking.pdf apresenta a melhor situação de coesão relacional, pois a maioria dos relacionamentos entre classes e / ou interfaces se concentra no pacote. Entende-se que tal situação favorece a coesão e a modularidade do projeto de PLA.

\*\*\*\* \*n\_12 VAR\_28\_banking-7114272521.

\*\*\*\* \*n\_13 O projeto VAR\_21\_agm-8156195255.

\*\*\*\* \*n\_14 Apesar de ter classes que parecem estar faltando, o que deixou algumas classes isoladas, o projeto do arquivo VAR\_All\_agm-4444523912.pdf é o que parece mais coeso.

\*\*\*\* \*n\_15 O arquivo agm.pdf parece ter uma melhor coesão, pois está entrelaçado com a mesma característica.

\*\*\*\* \*n\_16 MobileMedia. Considerando que a coesão relacional é diretamente relacionada a reusabilidade do produto, de acordo com a função CM; a opção citada anteriormente se apresenta como a melhor alternativa de projeto.

\*\*\*\* \*n\_17 Var\_28\_mobile.

\*\*\*\* \*n\_18 A melhor opção é a MobileMedia, pois a mesma possui menos classes e as mesmas possuem menos responsabilidades que não são suas.

#### **4. Qual a melhor alternativa de projeto sob o ponto de vista de acoplamento de classes?**

\*\*\*\* \*n\_10 No meu ponto e vista as PLAs banking e a PLA VAR\_28.banking-7114272521 possui menor acoplamento de classes, porém a PLA banking aparenta ter um menor acoplamento de classes.

\*\*\*\* \*n\_11 Entendo que a alternativa nomeada banking.pdf apresenta o grau mais baixo de acoplamento de classes, entre as alternativas. Tal entendimento se justifica pelo fato de que alguns tipos de relacionamentos, tais como o de dependência, são reduzidos em comparação com as outras alternativas.

\*\*\*\* \*n\_12 VAR\_28\_banking-7114272521.

\*\*\*\* \*n\_13 O projeto VAR\_All\_agm-4444523912.

\*\*\*\* \*n\_14 O projeto do arquivo VAR\_21\_agm-8156195255.pdf é o que parece menos acoplado, e também não parece ter classe faltando (o que poderia indicar relacionamentos faltando que piorariam o acoplamento caso fossem completados).

\*\*\*\* \*n\_15 Var\_ALL parece ter menor acoplamento.

\*\*\*\* \*n\_16 VAR\_All\_MobileMedia-6111288751. Considerando que o baixo acoplamento de classes é um fator que corrobora para a reusabilidade e manutenibilidade do código desenvolvido, a opção citada anteriormente se apresenta como a melhor alternativa de projeto.

\*\*\*\* \*n\_17 Var\_28\_mobile.

\*\*\*\* \*n\_18 Seria a MobileMedia, pois a mesma possui menos classes dependentes umas das outras. Chegando a ter pouco mais da metade de dependências em relação a VAR\_All\_MobileMedia-8589111691.

#### **5. Qual a melhor alternativa de projeto sob o ponto de vista de modularização de características?**

\*\*\*\* \*n\_10 Em relação a difusão de características, a PLA banking aparenta ter o menor índice de difusão. Em relação ao entrelaçamento de características a PLA banking aparenta ter o menor índice de entrelaçamento. Já em relação a coesão baseada em

características, os pacotes da PLA VAR\_All\_banking-4119842252 aparentam ser altamente coesos por apresentar menos características associadas. No geral, sob o ponto de vista e modularização de características, a PLA banking seria a melhor alternativa de projeto sob o ponto de vista de modularização de características.

\*\*\*\* \*n\_11 Entendo que a modularidade da alternativa nomeada banking.pdf é a mais adequada, pois entendo que foi a alternativa que apresentou a maior coesão e o menor nível de acoplamento, considerando aqui que coesão e acoplamento são bons indicadores de modularidade. Entende-se que as características estão bem organizadas, exceção feita às características user\_interaction e external\_user, que apesar de tratarem da mesma entidade (usuário) aparecem em pacotes diferentes.

\*\*\*\* \*n\_12 VAR\_28\_banking-7114272521.

\*\*\*\* \*n\_13 O projeto VAR\_All\_agm-4444523912.

\*\*\*\* \*n\_14 Não achei que algum dos projetos conseguiu separar uma variância de forma que ajudasse na modularização. Porém gostaria de destacar novamente que o projeto do arquivo VAR\_21\_agm-8156195255.pdf é o que parece mais completo em termos de ter as classes que atenderiam aos casos de uso.

\*\*\*\* \*n\_15 agm.pdf parece ser a melhor alternativa.

\*\*\*\* \*n\_16 VAR\_28\_MobileMedia-6161996518. Considerando a distribuição das classes e interfaces dentro dos pacotes e as respectivas características de cada um deles, a alternativa citada no início da resposta se apresenta como a melhor alternativa sob o aspecto da modularização de características. Numa análise visual, essa alternativa me pareceu ser a mais modularizada (ou a mais próxima da modularizada).

\*\*\*\* \*n\_17 Var\_28\_mobile.

\*\*\*\* \*n\_18 VAR\_All\_MobileMedia-8589111691, pois a mesma possui mais módulos em relação as demais e se subentende que sendo assim a PLA está mais dividida e mais legível com uma melhor manutenção e melhor desempenho.

### C.3 Corpus 3

O Corpus 3 é composto pelas respostas alcançadas pela questão 6 nos Questionário 1 e 2.

#### 6. Quais propriedades você julga importantes para se avaliar em um projeto de PLA?

\*\*\*\* \*n\_1 Julgo mais essenciais a coesão e o acoplamento. Sempre procuro analisar e definir as responsabilidades para os pacotes, interfaces, classes e características que serão modeladas e são realmente fundamentais para um bom projeto. Deste modo, é

possível tornar o projeto mais consistente, simples, reutilizável e viável, minimizando os seus principais problemas.

\*\*\*\* \*n\_2 Coesão, acoplamento, reusabilidade em termos de elementos abstratos, tamanho e estabilidade.

\*\*\*\* \*n\_3 Coesão, acoplamento e manutenibilidade.

\*\*\*\* \*n\_4 Não dá pra generalizar em uma propriedade só. A arquitetura tem que ser avaliada de diferentes perspectivas. Acoplamento, coesão, modularização. Entretanto é importante observar o nome dos métodos, atributos, classes pacotes. Isso orienta o programador a encontrar e navegar na arquitetura.

\*\*\*\* \*n\_5 Dependências entre classes e pacotes, quantidade de funcionalidades pelas quais classes e pacotes diferentes são responsáveis, tamanho de interfaces e o uso das mesmas sendo feito de maneira a simplificar o desenvolvimento.

\*\*\*\* \*n\_6 Julgo importante a coesão e o acoplamento.

\*\*\*\* \*n\_7 Pacotes bem definidos. Classes que só façam o que é pertinente a elas. Uso de interfaces quando necessário.

\*\*\*\* \*n\_8 Primeiramente o acoplamento, por segundo a coesão e então as demais propriedades.

\*\*\*\* \*n\_9 Além das propriedades utilizadas neste experimento, acredito que a extensibilidade e a manutenibilidade sejam também importantes, já que é uma forma de garantir uma maior qualidade, respectivamente, ao expandir/gerar novos produtos e na manutenção da PLA.

\*\*\*\* \*n\_10 Acredito que para um projeto de PLA seus elementos devem ter o maior número de independência possível de outros elementos, a fim de melhorar a manutenção e também sua reutilização (o que é considerado no ponto de vista de modularidade). Acredito que o conteúdo deste estudo é relevante para se avaliar projetos de PLA, considerando os níveis de coesão alto e de acoplamento baixo, e também a avaliação de modularização das características.

\*\*\*\* \*n\_11 Modularidade, reusabilidade e testabilidade.

\*\*\*\* \*n\_12 Coesão, acoplamento, tamanho e difusão de características.

\*\*\*\* \*n\_13 Difusão de características, coesão baseada em características, modularização de características.

\*\*\*\* \*n\_14 Organização das classes de forma lógica do ponto de vista de casos de usos porém se atentando a classes que participam da mesma variação. Não criar pacotes muito grandes que abriguem características diferentes mas também não criar muitos pacotes que partilham do mesmo módulo.

\*\*\*\* \*n\_15 Acredito que tendo uma alta coesão é melhor, pois permite centralizar as propriedades.

\*\*\*\* \*n\_16 Considerando-se que a PLA visa gerar produtos especificações particulares com base num núcleo de artefatos comuns, tal como numa linha de produção manufatureira; acredito que seja importante a definição de componentes bem delimitados; a documentação dos requisitos atendidos, sejam parcial ou totalmente, por cada componente; a formalização dos relacionamentos entre os componentes, independentemente da notação adotada; e a mitigação de esforços e custos para a obtenção de um novo produto (talvez o conceito da reusabilidade).

\*\*\*\* \*n\_17 Especificação das entidades e das funcionalidades.

\*\*\*\* \*n\_18 Uma arquitetura madura e sustentável, apresenta alta coesão e baixo acoplamento, pois cada um deles tem um propósito específico que visa melhorar o projeto da PLA.