

MARCOS DE FREITAS JUNIOR

**Melhoria na consistência da contagem de pontos de
função com base na *Árvore* de pontos de função**

São Paulo

2016

MARCOS DE FREITAS JUNIOR

**Melhoria na consistência da contagem de pontos de função
com base na *Árvore de pontos de função***

Versão corrigida

Dissertação apresentada à Escola de Artes, Ciências e Humanidades da Universidade de São Paulo para obtenção do título de Mestre em Ciências pelo Programa de Pós-graduação em Sistemas de Informação.

Área de Concentração: Metodologia e Técnicas da Computação

Versão corrigida contendo as alterações solicitadas pela comissão julgadora em 08 de dezembro de 2015. A versão original encontra-se em acervo reservado na biblioteca da EACH-USP e na Biblioteca Digital de Teses e Dissertações da USP (BDTD), de acordo com a resolução CoPGr 6018, de 13 de outubro de 2011.

Orientador: Prof. Dr. Marcelo Fantinato

São Paulo
2016

Autorizo a reprodução e divulgação total ou parcial deste trabalho, por qualquer meio convencional ou eletrônico, para fins de estudo e pesquisa, desde que citada a fonte.

CATALOGAÇÃO-NA-PUBLICAÇÃO

(Universidade de São Paulo. Escola de Artes, Ciências e Humanidades. Biblioteca)

Freitas Junior, Marcos de

Melhoria na consistência da contagem de pontos de função com base na Árvore de pontos de função / Marcos de Freitas Junior ; orientador, Marcelo Fantinato. – São Paulo, 2016
163 f. : il.

Dissertação (Mestrado em Ciências) - Programa de Pós-Graduação em Sistemas de Informação, Escola de Artes, Ciências e Humanidades, Universidade de São Paulo, em 2015

Versão corrigida

1. Engenharia de software. 2. Métricas de software. 3. Qualidade de processo de Software. 4. Análise de pontos de função. I. Fantinato, Marcelo, orient. II. Título

CDD 22.ed – 005.1

Dissertação de autoria de Marcos de Freitas Junior, sob o título “**Melhoria na consistência da contagem de pontos de função com base na Árvore de pontos de função**”, apresentada à Escola de Artes, Ciências e Humanidades da Universidade de São Paulo, para obtenção do título de Mestre em Ciências pelo Programa de Pós-graduação em Sistemas de Informação, na área de concentração Metodologia e Técnicas da Computação, aprovada em 08 de dezembro de 2015 pela comissão julgadora constituída pelos doutores:

Prof. Dr. Marcelo Fantinato

Presidente

Escola de Artes, Ciência e Humanidades – Universidade de São Paulo

Prof. Dr. Gleison dos Santos Souza

Escola de Informática - Universidade Federal do Estado do Rio de Janeiro

Prof. Dr. Anarosa Alves Franco Brandão

Escola Politécnica – Universidade de São Paulo

Prof. Dr. Marcelo Medeiros Eler

Escola de Artes, Ciência e Humanidades – Universidade de São Paulo

AGRADECIMENTOS

Agradecimentos serão, nesse momento, meu singelo reconhecimento a aqueles que, de forma direta ou indireta, me ajudaram a desenvolver este trabalho que é, para mim, o resultado de uma pós-graduação intensa e gratificante.

Em primeiro lugar, agradeço ao meu orientador Dr. Marcelo Fantinato que me acolheu sem receio e me possibilitou desenvolver um grande trabalho por meio de sua admirável e incansável dedicação em me orientar minuciosamente em todos os quesitos acadêmicos, o que exigiu paciência e, sobretudo, dedicação a sua atividade profissional. A Dr. Violeta Sun que, não só me orientou durante a graduação, como também me incentivou e encorajou a prosseguir em busca de novos desafios na área acadêmica e esteve presente durante o desenvolvimento dessa pesquisa. Ao Dr. Edmir Parada Vasques Prado por ter me recebido pacientemente na Universidade de São Paulo para me apresentar o Programa e ter me dado conselhos tão preciosos para que pudesse iniciar minha jornada na área acadêmica. Ao Dr. Luciano Araújo que auxiliou no início de minha pesquisa enriquecendo ainda mais o presente trabalho.

Também aos professores do Programa de Pós-graduação de Sistemas de Informação, que exigiram dedicação aos estudos. Aos professores da banca de qualificação, Dr. Gleison dos Santos Souza e Dr. João Bernardes, pela gentileza da participação e pelos comentários necessários. Aos professores da banca de defesa Dra. Anarosa Alves Franco Brandão, Dr. Gleison dos Santos Souza e Dr. Marcelo Medeiros Eler pela gentileza da participação e contribuição a esse trabalho.

A ti Métricas por me fornecer a flexibilidade necessária para que pudesse desenvolver este trabalho paralelamente as atividades profissionais e, particularmente, a Diana Baklizky por ter confiado no meu potencial e ter permitido que eu me desenvolvesse no assunto. Aos respondentes de ambas as organizações que se disponibilizaram em participar do experimento da abordagem.

Aos colegas e amigos de mestrado que me aconselharam e compartilharam alegrias e preocupações durante o curso, em especial, ao Ricardo Corrêa. A minha namorada Adriana Dominici Cintra, que não titubeou em nenhum segundo em me fornecer apoio, me passando sempre tranquilidade e sendo por dez anos meu pilar nos momentos mais difíceis.

*“A tarefa não é tanto ver aquilo que ninguém viu, mas pensar o que
ninguém ainda pensou sobre aquilo que todo mundo vê.”
Arthur Schopenhauer.*

Resumo

FREITAS JUNIOR, Marcos. **Melhoria na consistência da contagem de pontos de função com base na Árvore de pontos de função**. 2016. 163 f. Dissertação (Mestrado em Ciências) – Escola de Artes, Ciências e Humanidades, Universidade de São Paulo, São Paulo, 2015.

Análise de Pontos de Função (APF) é uma das medidas usadas para obter o tamanho funcional de um software. Determinou-se, no Brasil, que toda contratação pública de desenvolvimento de software deve usar APF. Entretanto, uma das principais críticas realizadas a APF diz respeito à falta de confiabilidade entre diferentes contadores em uma mesma contagem já que, segundo alguns pesquisadores, as regras de APF são subjetivas, obrigando que cada contador faça interpretações individuais a partir delas. Existem diversas propostas para que se possa aumentar a confiabilidade dos resultados gerados com APF. Em geral, as abordagens propostas realizam mapeamentos entre componentes de artefatos desenvolvidos no ciclo de vida de software com os conceitos de APF. Porém, tais propostas simplificam em mais de 50% as regras previstas em APF comprometendo a validade dos resultados gerados pelas contagens. Como o tamanho do software é usado na derivação de outras medidas, inconsistências nos tamanhos medidos podem comprometer as medidas derivadas, o que influencia negativamente nas decisões tomadas. Sem padronização dos tamanhos funcionais obtidos e – conseqüentemente – sem confiabilidade dos resultados obtidos, medidas derivadas a partir do tamanho funcional, como custo e esforço, podem estar comprometidas, fazendo com que ela não ajude a influenciar positivamente tais projetos. Diante desse contexto, o objetivo deste trabalho é desenvolver e avaliar experimentalmente uma abordagem para oferecer maior padronização e sistematização na aplicação de APF. Para isso, propõe-se incorporar o artefato “Árvore de pontos de função” ao processo de APF. Sua inclusão possibilitaria o levantamento de dados adicionais, necessários à contagem de pontos de função, reduzindo a ocorrência de interpretações pessoais do contador, e conseqüentemente, a variação de tamanho reportado. A abordagem foi denominada como Análise de Pontos de Função baseada em Árvore de Pontos de Função (APF-APF). Este trabalho baseia-se no método de pesquisa *Design Science*, cujo objetivo é estender os limites do ser humano e as capacidades organizacionais, criando novos artefatos que solucionem problemas ainda não resolvidos ou parcialmente resolvidos; que neste trabalho, trata-se da falta de confiabilidade na aplicação de APF devido à sua margem para diferentes interpretações. APF-APF foi testada com 11 Analistas de Sistemas / Requisitos que, baseados na especificação de um software de Recursos Humanos medido oficialmente pelo IFPUG com 125 pontos de função, modelaram a Árvore de pontos de função de modo manual ou automatizado via protótipo de ferramenta desenvolvido. Os resultados obtidos indicam que os tamanhos funcionais calculados com APF-APF possuem coeficiente de variação, respectivamente de 10,72% em relação a confiabilidade e 17,61% em relação a validade dos resultados de medição gerados. Considera-se que a abordagem APF-APF mostrou potencial para que melhores resultados possam ser obtidos. Verifica-se que a principal causa das variações observadas estava relacionada a ausência de informações requeridas para a Árvore de pontos de função, não tendo sido identificado nenhum problema específico em relação as regras definidas para APF-APF. Por fim, verificou-se que o uso do protótipo de ferramenta desenvolvido aumenta em até 47% a eficiência na contagem de pontos de função quando comparado com APF-APF manual.

Palavras-chave: Medições funcionais. Análise de Pontos de Função. APF. Confiabilidade de APF. Análise de Pontos de Função baseada em Árvore de Pontos de Função. APF-APF.

Abstract

FREITAS JUNIOR, Marcos. **Improvement in the consistency of function point counting based on the Function Points Tree**. 2016. 163 p. Dissertation (Master of Science) – School of Arts, Sciences and Humanities, University of São Paulo, São Paulo, 2015.

Function point analysis (FPA) is one of the measures used to achieve the functional size of software. It was determined, in Brazil, public procurement of software development should use FPA. However, one of the main criticisms made the FPA concerns the lack of reliability between different counters on the same count that, according to some researchers, the FPA rules are subjective, requiring that each counter do individual interpretations from them. There are various proposals in order to increase the reliability of the results generated with FPA. In General, the proposed approaches perform mappings between artifacts developed components in software life cycle with the concepts of FPA. However, such proposals simplify in more than 50% the rules laid down in FPA compromising the validity of the results generated by the scores. As the size of the software is used in the derivation of other measures, inconsistencies in sizes measured may compromise the measures derived, which negatively influence the decisions taken. Without standardization of functional sizes obtained and – consequently – without reliability of the results obtained, derived from measures of functional size, cost and effort, may be compromised, causing it to not help to positively influence these projects. In this context, the objective of this work is to develop and experimentally evaluate one approach to offer greater standardization and systematization in the implementation of FPA. For this, it is proposed to incorporate the artifact "function point Tree" to the FPA process. Its inclusion would allow additional data collection necessary for function point count, reducing the occurrence of personal interpretations of the counter, and consequently, the variation of size reported. The approach was called as Function Point Tree-based Function Point Analysis (FPT-FPA). This work is based on the method of Design Science research, whose goal is to extend the limits of the human and organizational capacities, creating new artifacts to troubleshoot unresolved or still partially resolved; in this work, it is the lack of reliability in application of FPA because of its scope for different interpretations. FPT-FPA were tested with 11 Systems analysts / requirements analysts, based on the specification of a human resources software measured by the IFPUG with 125 points, have modeled the function point Tree manually or via automated tool prototype developed. The results obtained indicate that the functional sizes calculated with FPT-FPA have coefficient of variation, respectively of 10.72% for reliability and 17.61% in relation to the validity of the measurement results generated. The FPA approach showed potential for better results can be obtained. It turns out that the main cause of the variations observed were related to the absence of information required for the tree of function points have not been identified any particular problem regarding the rules defined for FPT-FPA. Finally, it was found that the use of a prototype tool increases by up to 47% on efficiency function point count when compared to FPT-FPA manual.

Keywords: Functional Metrics. Function Point Analysis. FPA. Reliability of FPA. Function Point Tree-based Function Point Analysis. FPT-FPA.

Lista de figuras

Figura 1 - Requisitos medidos por APF.....	28
Figura 2 - Procedimento para a contagem de pontos de função via APF.....	29
Figura 3 - Procedimento para determinar o escopo e a fronteira da contagem.....	30
Figura 4 - Procedimento para medir as funções de dados.....	31
Figura 5 - Procedimento para medir as funções de transação.....	33
Figura 6 - Exemplo da Árvore de Refinamento de Funções do SISRH.....	41
Figura 7 - Exemplo da Árvore de pontos de função.....	68
Figura 8 - Exemplo de tela para cadastro de empregado que ilustra atributos de nó da Árvore de pontos de função.....	74
Figura 9 - Diagrama de casos de uso do protótipo de ferramenta.....	89
Figura 10 - Interface principal do protótipo de ferramenta com a modelagem de um software de gestão de RH.....	92
Figura 11 - Interface para incluir informações de uma funcionalidade modelada.....	93
Figura 12 - Interface para seleção dos rótulos de entidades de dados dependentes entre si..	94
Figura 13 - Pop-up apresentando o tamanho total do software calculado pela ferramenta...	95
Figura 14 - Interface para apresentar o detalhe do tamanho funcional do software calculado pela ferramenta.....	95

Lista de fluxogramas

Fluxograma 1	-	Processo padrão para a contagem de pontos de função.....	63
Fluxograma 2	-	Subprocesso Extrair informações necessárias.....	64
Fluxograma 3	-	Contagem de pontos de função baseada em Árvore de pontos de função.....	65
Fluxograma 4	-	Procedimento para identificar as funções de dados.....	80
Fluxograma 5	-	Detalhamento de subprocesso para Agrupar rótulos de entidade de dados.....	80
Fluxograma 6	-	Procedimento para classificar cada função de dados.....	81
Fluxograma 7	-	Procedimento para identificar os processos elementares do software...	83
Fluxograma 8	-	Detalhamento de subprocesso para Agrupar nós folha.....	84
Fluxograma 9	-	Procedimento para determinar os processos elementares únicos.....	84
Fluxograma 10	-	Procedimento para classificar cada função de transação.....	85

Lista de gráficos

- Gráfico 1 - Distribuição dos tipos de artefato produzidos por técnicas de modelagem usadas nas abordagens propostas para diminuir a subjetividade de APF..... 45
- Gráfico 2 - Perfil dos respondentes selecionados para participar do experimento..... 100
- Gráfico 3 - Motivos para a variação do coeficiente de confiabilidade e validade dos resultados de medição obtidos por APF-APF quando comparados com os coeficientes pretendidos..... 104

Lista de tabelas

Tabela 1 - Complexidade de funções de dados.....	32
Tabela 2 - Tamanho funcional de funções dados por grau de complexidade.....	33
Tabela 3 - Complexidade de funções de transação (tipo EE).....	35
Tabela 4 - Complexidade de funções de transação (tipos SE e CE).....	36
Tabela 5 - Tamanho funcional de funções de transação (tipos EE e CE) por grau de complexidade.....	36
Tabela 6 - Tamanho funcional de funções de transação (tipo SE) por grau de complexidade.....	36
Tabela 7 - Pontuação obtida pelas abordagens em relação à cobertura de APF.....	55
Tabela 8 - Resultados obtidos após a execução dos experimentos.....	101
Tabela 9 - Resultados consolidados por visões após a execução dos experimentos.....	101
Tabela 10 - Comparação dos resultados obtidos com APF-APF em relação aos resultados reportados por abordagens similares.....	108

Lista de quadros

Quadro 1 - Descrição dos marcadores de raiz usados para representar o objetivo da contagem.....	69
Quadro 2 - Descrição dos marcadores de raiz usados para representar características do software.....	70
Quadro 3 - Descrição dos marcadores de nó usados para representar características relacionadas ao comportamento das funções elementares equivalentes as lógicas de processamento de APF.....	71
Quadro 4 - Descrição dos marcadores de nó usados para representar características adicionais relacionadas ao comportamento das funções elementares sem equivalência com as lógicas de processamento de APF.....	72
Quadro 5 - Descrição do conector de inclusão.....	73
Quadro 6 - Descrição dos marcadores de rótulos de entidade de dados.....	76
Quadro 7 - Descrição do conector de exclusão.....	77

Lista de abreviaturas e siglas

AIE	Arquivos de Interface Externa
ALI	Arquivo Lógico Interno
ALR	Arquivo Lógico Referenciado
APF	Análise de Pontos de Função
APF-APF	Análise de Pontos de Função baseada em Árvore de Pontos de Função
ATM	Automatic Teller Machine
CBO	Classificação Brasileira de Ocupações
CE	Consulta Externa
CPUs	Central Processing Units
CFPS	Certified Function Point Specialist
CGS	Características Gerais do Sistema
CPM	Counting Process Manual
DER	Dado Elementar Referenciado
EE	Entrada Externa
EDPS	Electronic Data Processing Systems
ERbCO	Elicitação de Requisitos baseada em Cenários e Objetivos
ER-DFD	Entidade Relacionamento – Diagrama Fluxo de Dados
FPA	Function Point Analysis
JDBC	Java Database Connectivity
IFPUG	International Function Points Users Group
MCbER	Modelagem Conceitual baseado em Engenharia de Requisitos
MHZ	MegaHertz
OO-H	Object-oriented Hypermedia
PF	Pontos de Função
RLR	Registro Lógico Referenciado
RSL	Revisão Sistemática da Literatura
SE	Saída Externa
SISRH	Sistema de Recursos Humanos
SQL	Structured Query Language
VDM-SL	Vienna Development Method-Specification Language
UML	Unified Modeling Language
WebML	Web Modeling Language

Sumário

1. Introdução	18
1.1 Motivação e justificativa.....	18
1.2 Objetivos	21
1.3 Estrutura do documento.....	22
2. Fundamentação teórica.....	24
2.1 Medição de software	24
2.1.1 Medição, medidas e resultado da medição.....	24
2.1.2 Princípios básicos de medição	25
2.1.3 Classificação das medidas de software.....	27
2.2 Análise de pontos de função	28
2.2.1 Obter documentação disponível	29
2.2.2 Determinar escopo e fronteira da contagem.....	29
2.2.3 Medir funções de dados	31
2.2.4 Medir funções de transação.....	33
2.2.5 Calcular tamanho funcional.....	36
2.2.6 Documentar e relatar	38
2.3 Modelagem conceitual baseada em engenharia de requisitos	38
2.4 Considerações finais.....	40
3. Trabalhos relacionados.....	42
3.1 Problemas relacionados ao uso de APF	42
3.2 Melhorias propostas para diminuir a subjetividade de APF.....	44
3.3 Classificação das melhorias para diminuir a subjetividade de APF.....	55
3.4 Considerações finais.....	57
4. Método de pesquisa	58
4.1 Protocolo da revisão sistemática.....	59

5. Análise de pontos de função baseada em APF-APF	61
5.1 Visão geral da abordagem APF-APF	62
5.2 Árvore de pontos de função.....	66
5.2.1 Visão geral da estrutura da Árvore de pontos de função.....	66
5.2.2 Raiz.....	69
5.2.2.1 Marcador de raiz	69
5.2.3 Nós folha.....	70
5.2.3.1 Marcador de nó	70
5.2.3.2 Conector de inclusão.....	73
5.2.3.3 Atributo de nó	73
5.2.4 Relacionamentos entre níveis	75
5.2.4.1 Rótulo de entidade de dados	75
5.2.4.2 Marcador de rótulo de entidade	75
5.2.4.3 Conector de exclusão	75
5.2.4.4 Atributo de rótulo de entidade	76
5.3 Uso da Árvore de pontos de função na contagem de pontos de função.	77
5.3.1 Obter documentação disponível	77
5.3.2 Determinar escopo e fronteira da contagem.....	78
5.3.3 Medir funções de dados	78
5.3.3.1 Identificar as funções de dados.....	78
5.3.3.2 Classificar cada função de dados	79
5.3.3.3 Determinar complexidade funcional.....	81
5.3.3.4 Determinar a contribuição	81
5.3.4 Medir funções de transação.....	82
5.3.4.1 Identificar os processos elementares do software.....	82
5.3.4.2 Determinar os processos elementares únicos	83
5.3.4.3 Classificar cada função de transação	85
5.3.4.4 Determinar a contribuição	85
5.3.5 Calcular tamanho funcional, e documentar e relatar.....	86
5.4 Considerações finais.....	86
6. Prototipação e avaliação da abordagem APF-APF	87
6.1 Protótipo de ferramenta	87

6.1.1	Visão geral do protótipo de ferramenta	87
6.1.2	Requisitos do protótipo de ferramenta	88
6.1.3	Premissas.....	89
6.1.4	Arquitetura do protótipo de ferramenta.....	89
6.1.4.1	Componente modelo	90
6.1.4.2	Componente visão	90
6.1.4.3	Componente controle.....	91
6.1.4.4	Componente persistência	91
6.1.5	O protótipo de ferramenta desenvolvido	91
6.2	Avaliação da abordagem APF-APF	95
6.2.1	Procedimento do experimento	96
6.2.2	Ameaças à validade do experimento	98
6.2.3	Resultados do experimento	99
6.2.4	Análise qualitativa de APF-APF considerando os resultados obtidos	100
6.3	Considerações finais.....	107
7.	Conclusão	110
7.1	Contribuição científica	111
7.2	Contribuição para o mercado	111
7.3	Trabalhos futuros.....	113
	Referências.....	114
	Apêndice A – Especificação de requisitos do SISRH	123
	Apêndice B - Comparativo entre as abordagens propostas por trabalhos relacionados.....	125
	Apêndice C – Artigos publicados e relacionados à dissertação..	136
	Apêndice D – Protocolo de revisão sistemática da literatura	138
	Apêndice E – Diagramas de classe do protótipo de ferramenta.	143

Apêndice F – Modelagens da Árvore de pontos de função entregue pelos respondentes após o experimento	148
Apêndice G – E-mail enviado aos respondentes para realização de experimento	162

1. Introdução

Medições têm se tornado indispensáveis em diferentes áreas de conhecimento. É comum precisar medir, seja em atividades corriqueiras do dia-a-dia, como cozinhar ou tomar uma dose indicada de um medicamento, seja para atividades complexas, como fabricar componentes de engenharia de precisão usados em motores aeronáuticos, feitos por empresas que lidam com especificações estritas e que devem ser capazes de medir tamanho, composição e desempenho a níveis precisos, ou até mesmo exatos (NATIONAL MEASUREMENT SYSTEM, 2009).

Para a área de engenharia de software, medições são ainda mais desafiadoras, já que seu produto fim – o software¹ – não possui características tangíveis. A dificuldade em medir sua complexidade, seu tamanho, seu tempo de desenvolvimento, entre outras características é um desafio para os profissionais da área que normalmente tentam se basear em medições para transpor tais características abstratas via números palpáveis (FENTON; BIEMAN, 2014).

Dentre as medidas de software existentes, a Análise de Pontos de Função (APF), concebida por Albrecht (1979), foi a primeira medida usada para medir requisitos funcionais dos usuários, independente de tecnologia ou arquitetura usada para desenvolver o software. A desvinculação de fatores tecnológicos possibilitou que tamanhos de software – quando baseados em um mesmo conjunto de requisitos funcionais – pudessem ser comparados entre diferentes organizações (IFPUG, 2010).

APF oferece um conjunto de regras para medir um software, especificadas no Manual de Práticas de Contagem (CPM – *Counting Practice Manual*) para regulamentar a contagem. O IFPUG (*International Function Point Users Group*), fundado em 1986 com a missão de aprimorar e incentivar o uso mundial de APF é responsável por revisar e aprimorar o CPM de modo a garantir que as regras para a contagem de pontos de função sejam coesas e objetivas.

1.1 Motivação e justificativa

APF é usada para medir o tamanho de software baseado apenas em seus requisitos

¹ Para fins de padronização, neste texto, os seguintes termos são considerados sinônimos, desprezando possíveis diferenças semânticas entre eles: software, sistema, sistema de informação, sistema computacional, aplicação, aplicativo, e outros similares. Para todos esses casos, é adotado exclusivamente o termo “software”, a não ser quando necessário diferenciar semanticamente entre um dos outros termos citados.

funcionais. Ela possibilita que noções vagas sobre o tamanho de um determinado software (tais como “grande”, “pequeno”, “maior que”, etc.) sejam quantificadas. Considerando a natureza da área de engenharia de software, que lida com um produto não tangível, valores numéricos relacionados ao tamanho do software são importantes (FENTON; BIEMAN, 2014).

A partir do tamanho do software, é possível derivar outras medidas, como: produtividade de codificação de programadores; número de defeitos por pontos de função; esforço em horas usado em tarefas de desenvolvimento; custo do projeto a ser cobrado dos clientes, etc. (FENTON; BIEMAN, 2014). Tais medidas adicionais, derivadas do tamanho, são comumente usadas em contratos formais estabelecidos entre contratadas e contratantes, nos setores privado e público, para regulamentar a contratação de desenvolvimento e de manutenção de software bem como a bonificação ou a penalização das empresas contratadas. Dependendo do número de defeitos por pontos de função, por exemplo, pode ser estabelecido que a contratada deva ser multada em um determinado valor monetário (ADAMO; FABRIZI; VERGATI, 2008).

De acordo com a pesquisa “Qualidade no Setor de Software Brasileiro”, conduzida pela Secretaria de Política de Informática do Ministério da Ciência e Tecnologia (BRASIL, 2010c), verifica-se que APF vem sendo usada em várias organizações brasileiras nesse setor. Em 2001, o número de organizações brasileiras que usavam APF era de 18% do total de organizações entrevistadas. Em 2010, ano da última pesquisa encomendada pela secretaria, esse número passou para 34, 5%. Portanto, houve um aumento significativo do número de organizações brasileiras que adotam APF durante o processo de engenharia de software.

O Brasil, especificamente, apresenta um papel importante no cenário mundial de medições de software relacionado à APF. Segundo dados disponibilizados pelo IFPUG (2010), trata-se do país com o maior número de profissionais certificados em APF, contando com 37% do total. Essa alta taxa se deve a evidência dada no Brasil da importância de APF para apoiar estimativas de projetos de engenharia de software. Como exemplo, o governo federal brasileiro determinou que toda contratação pública de desenvolvimento de software deve ser balizada por uma medida objetiva (BRASIL, 2010a). Além disso, o uso de APF é explicitamente recomendado aos órgãos integrantes do Sistema de Administração dos Recursos de Informação e Informática (SISP); e a adoção do roteiro de medições de software do SISP é recomendada para contratar serviços de desenvolvimento e de manutenção de software (BRASIL, 2010b).

Apesar da importância de APF, diferentes especialistas apontam falta de confiabilidade em seus resultados. Segundo Keremer (1990) e Kampstra (2005), por exemplo, o resultado da contagem de pontos de função pode variar de 10% a 30% ao se considerar diferentes contadores². Isso ocorre pela subjetividade das regras de APF, que leva a resultados finais diferentes já que os contadores possuem interpretações individuais distintas. De acordo com Lavazza e Bianco (2008), APF é baseada em identificar componentes funcionais básicos do software, o que é realizado de acordo com regras consideradas informais e, portanto, o cálculo final do tamanho do software é também baseado em um conjunto de definições informais. Abrahão, Gomez e Insfrán (2010) concordam que existe subjetividade para aplicar APF, demandando um especialista para aplicá-la de modo menos sujeito a interpretações individuais. Huang, Ren e Yang (2012) consideram que há obscuridade nos conceitos definidos em APF. Abualkishik et al. (2012) relatam que há redundância na contagem de alguns elementos, que não é possível medir todos os tipos de software com APF, e que APF possui um tipo de transformação de escala considerada ilegal em relação à literatura de medições.

Como o tamanho do software é usado na derivação de outras medidas, inconsistências nos resultados de medição podem comprometer as medidas derivadas, o que influencia negativamente nas decisões. Esses problemas comprometem não apenas o modelo de contratação de terceiros baseados em APF estabelecidos em países como o Brasil, como também quaisquer outras medidas derivadas a partir do tamanho em APF. Sem padronização dos tamanhos funcionais obtidos e – consequentemente – sem confiabilidade dos resultados obtidos, medidas derivadas a partir do tamanho funcional, como custo e esforço, podem estar comprometidas. Embora APF possa ser usada para aumentar as chances de sucesso de um projeto de engenharia de software, os problemas relatados podem acabar fazendo com que ela não ajude a influenciar positivamente tais projetos.

Uma das causas, observadas na literatura, para as variações dos resultados de medição calculados entre distintos contadores é a ausência de informações necessárias para a contagem de pontos de função nos artefatos que compõem o ciclo de vida do software. Diante desse contexto, contadores acabam assumindo premissas para executar a contagem ou questionando analistas informalmente sem que tais informações sejam incorporadas aos artefatos do

² Por fins de simplicidade, neste texto, é usado o termo “contador” em detrimento ao uso do termo “Executor de medição”. Esse último, é recomendado por Barcellos (2009) em ontologia publicada. Porém, a semântica do termo é mantida, sendo contador um recurso humano que executa uma medição (Barcellos, 2009).

software usados para a contagem de pontos de função. Neste caso, as informações adicionais acabam descentralizadas, dificultando seu reuso por outros contadores que, futuramente, necessitem destas informações para a realização de novas contagens. Conseqüentemente, novas interpretações de outros contadores serão necessárias para o mesmo cenário, podendo levar a resultados diferentes.

Existem diversas propostas para aumentar a confiabilidade dos resultados gerados com a medida APF. Em geral, as abordagens propostas realizam mapeamentos entre componentes de artefatos desenvolvidos no ciclo de vida de software com os conceitos de APF. Porém, tais propostas simplificam demasiadamente as regras previstas em APF comprometendo a validade dos resultados gerados pelas contagens. As simplificações realizadas geram um fenômeno parecido com o relatado por alguns autores em relação a contadores: a variação do tamanho funcional medido.

1.2 Objetivos

O principal objetivo deste trabalho se refere a desenvolver e avaliar experimentalmente uma abordagem para oferecer maior padronização e sistematização na aplicação de APF. Essa abordagem propõe a extensão da técnica “Árvore de refinamento de funções”, que é parte de uma abordagem existente de engenharia de requisitos chamada “Modelagem conceitual baseada em engenharia de requisitos”, com o propósito de incorporar, de modo estruturado, informações adicionais necessárias exclusivamente à contagem de pontos de função. A técnica estendida recebe o nome de Árvore de pontos de função. Estando definidas de modo padronizado e sistemático, essas informações podem ser usadas para a contagem de pontos de função de modo automatizado. Assim, a coleta de todas as informações necessárias para a contagem é concentrada em um único artefato, diminuindo a ocorrência de interpretações individuais por parte dos contadores.

As abordagens propostas na literatura, por simplificarem demasiadamente as regras de APF original, geram resultados sem validade e não consistentes quando comparados entre si. Sendo assim, considera-se que tais abordagens não são completas o suficiente para produzirem resultados válidos e confiáveis. Deste modo, busca-se com a proposta dessa abordagem, alcançar os seguintes objetivos específicos:

- Aumentar a **confiabilidade** da contagem de pontos de função, diminuindo as variações ocasionadas por interpretações individuais de diferentes contadores;

- Manter a **validade** dessa contagem, de modo que a nova abordagem produza resultados concordantes com APF original padronizada pelo IFPUG;
- Aumentar a **eficiência** da contagem de pontos de função, por meio do uso de passos padronizados e sistemáticos que possam ser automatizados via ferramenta computacional. Espera-se assim que a nova abordagem demande menos esforço para que a medição seja realizada, e que conseqüentemente possa ser realizada de forma mais rápida, quando comparada com APF original padronizada pelo IFPUG.

Confiabilidade e validade são conceitos inter-relacionados que se referem a verificar a consistência³ e a concordância de resultados de medição obtidos a partir de medições repetidas, em condições similares ou idênticas, quando comparadas a um padrão de referência. No primeiro caso, o padrão de referência são os resultados gerados a partir de APF-APF; no segundo caso, o padrão de referência é o resultado idealmente correto a ser produzido por APF original padronizada pelo IFPUG. Assim, objetiva-se aumentar a confiabilidade ou a consistência dos resultados sendo produzidos, de forma a diminuir a variação entre diferentes contagens, desde que se mantenha a validade ou a concordância desses resultados em relação ao que APF original prevê. Ou seja, não seria desejado aumentar a confiabilidade e perder a validade dos resultados de medição gerados, o que poderia ser ocasionado, por exemplo, por um excesso de simplificações dos passos da medida.

1.3 Estrutura do documento

Este documento está dividido nas seguintes partes: capítulo 2, que contextualiza o tema em estudo via fundamentação teórica; capítulo 3, que apresenta uma análise de trabalhos relacionados; capítulo 4, que apresenta o método de pesquisa empregado para realizar o trabalho; capítulo 5, que apresenta a abordagem concebida para solução do problema abordado; capítulo 6, que apresenta a validação experimental da abordagem proposta; capítulo 7, que apresenta as conclusões desta dissertação; apêndice A, que contém a especificação de requisitos de um software de recursos humanos; apêndice B, que contém dados sistematizados sobre os trabalhos relacionados; apêndice C, que contém a lista de artigos publicados

³ De acordo com Michaelis (2011), a palavra consistência é definida como: “Estado de uma coisa que promete durar ou não ter mudança.”. No contexto desse trabalho, para fins de padronização, o termo consistência irá referir-se exclusivamente a: um resultado de medição, obtido a partir de APF, que não sofre alterações quando calculado por contadores distintos. Essa interpretação é aderente à definição fornecida em Michaelis (2011).

relacionados a este trabalho; apêndice D, que contém a descrição da revisão sistemática da literatura realizada; apêndice E, que contém a descrição do projeto lógico do protótipo da ferramenta implementada para testar a abordagem proposta; apêndice F, que contém o detalhamento dos resultados obtidos após execução do experimento e apêndice G, que contém o e-mail enviado aos respondentes com a descrição dos procedimentos para a participação do experimento de avaliação da abordagem proposta.

2. Fundamentação teórica

Neste capítulo, são apresentados os principais conceitos relacionados à abordagem proposta neste trabalho, incluindo: medição de software; medida APF; e a técnica de engenharia de requisitos usada para embasar a abordagem proposta neste trabalho.

2.1 Medição de software

Esta seção apresenta conceitos básicos sobre medição de software, incluindo: diferenças entre termos da área; princípios básicos de medição; e classificação das medidas de software.

2.1.1 Medição, medidas e resultado da medição

A área de conhecimento de medição de software tem desempenhado um papel importante na Engenharia de Software, porém, ainda apresenta problemas típicos de uma disciplina recente – a falta de padronização de terminologias, conceitos e métodos de medição, que vem sendo definidos, consolidados e acordado nas últimas décadas. Mesmo os termos mais comuns, tais como “medição”, “métrica”, “medida”, “atributo mensurável”, ainda não possuem consenso de significado entre pesquisadores e praticantes da área (GARCIA et al., 2006).

Neste trabalho, as seguintes definições apresentadas por Barcellos (2009) são adotadas:

- **Medida:** instrumento usado para associar um valor a um *elemento mensurável* usando uma *escala*, sendo que:
 - **Elemento mensurável:** propriedade mensurável de um *tipo de entidade mensurável* por meio do qual as instâncias desse tipo de *entidade mensurável* podem ser caracterizadas, sendo que:
 - **Tipo de entidade mensurável:** classificação de entidades mensuráveis que indica quais elementos mensuráveis podem ser usados para caracterizar entidades desse tipo.
 - **Entidade mensurável:** entidade que pode ser medida, representa uma instância de um tipo de entidade mensurável.

- **Escala:** estrutura de qualidade formada por valores discretos ou contínuos ou por categorias para a qual uma medida é mapeada.
- **Medição:** ação de medir um *elemento mensurável*, aplicando uma *medida* e obtendo um *resultado de medição* que define um valor em uma determinada *escala*.
- **Resultado da medição**⁴: artefato produzido em uma *medição*, contendo o valor resultante propriamente dito.

Um exemplo apresentado por Barcellos (2009) é descrito como: caso se deseje registrar o número de requisitos homologados pelo cliente que sofreram alterações em um período. Nesse cenário, *tipo de entidade mensurável* é “artefato”, *entidade mensurável* é “especificação de requisitos do projeto em questão”, e *elemento mensurável* são “requisitos alterados”. Assim, a *medição* do *elemento mensurável*, aplicando a *medida* “número de requisitos alterados”, obtém o resultado que define como *resultado da medição* o valor definido como “8”.

2.1.2 Princípios básicos de medição

Para estabelecer um processo de medição, identificam-se: os elementos mensuráveis das entidades do mundo real que se deseja medir; e as relações empíricas entre os elementos mensuráveis e suas relações numéricas correspondentes com cada relação empírica. Identificar esses elementos e dessas relações possibilita que se defina um mapeamento das entidades do mundo real para um sistema numérico de modo que se preservem as relações empíricas (FENTON; BIEMAN, 2014).

Por exemplo, caso se deseje medir a altura de uma pessoa, não basta apenas especificar um número ou um ícone. É necessário, inicialmente, perceber quais elementos mensuráveis das entidades do mundo real se deseja medir. Nesse exemplo, “altura” representa o atributo que é de interesse de alguém para que se possa realizar a medição. Após a escolha dos atributos a serem medidos, as relações existentes no mundo real devem ser observadas, de

⁴ Tradicionalmente, vários autores têm usado “métrica” como sinônimo de “resultado da medição”. Embora “métrica” seja mais antigo para esse fim, e amplamente usado e aceito por muitos profissionais e pesquisadores, seu uso pode causar confusão já que “medição” e “métrica” são bastante similares, sendo atualmente preferencial o uso dos termos “medição” e “resultado da medição”. De acordo com Garcia et al. (2006), organismos internacionais tais como ISO, IEC e IEEE já têm buscado evitar o uso do termo *métrica*. Assim, para fins de padronização, neste texto, apenas o termo “resultado da medição” é usado, em detrimento ao termo “métrica”.

modo que elas sejam mantidas no sistema numérico. No exemplo anterior, isso pode ser executado observando como o “tamanho” de uma pessoa se comporta quando comparado com o “tamanho” de outra pessoa. A partir dessa observação, é possível determinar relações empíricas tais como: “fulano” é *maior* que “beltrano”, ou “ciclano” é *muito maior* que “fulano”.

As relações empíricas observadas permitem determinar a relação numérica entre atributos. Baseado em como os atributos se comportam e se relacionam no mundo real, números ou ícones são atribuídos aos atributos das entidades. Assim, de acordo com o exemplo anterior, a relação numérica possibilita que valores sejam definidos para cada “tamanho”.

Após identificar as relações empíricas e numéricas entre os atributos, realiza-se um mapeamento entre os elementos mensuráveis das entidades e o sistema numérico. Por exemplo, se “altura” deve ser medida em polegadas, um mapeamento para o conjunto de alturas em polegadas deve ser definido. Se a opção for medir em centímetros, outro mapeamento é necessário. Porém, mesmo quando o domínio e a escala de uma medição são iguais, podem existir diferentes mapeamentos dependendo das convenções adotadas. Ainda de acordo com o exemplo anterior, ao medir a altura de uma pessoa, deve-se permitir ou não o uso de sapatos, ou pode-se medir as pessoas em pé ou sentadas. Assim, cada medição deve ser associada a um modelo que descreve como a medição mapeia as entidades e os elementos mensuráveis do mundo real para os elementos do sistema numérico. Modelos são essenciais para entender como a medição é derivada assim como para interpretar o comportamento dos elementos numéricos quando os mesmos, após a medição, são apresentados no mundo real.

O mapeamento de um elemento mensurável para um valor de medição pode ser realizado de diferentes modos, os quais representam diferentes escalas de medição. As escalas de medição são classificadas em: nominal, ordinal, de intervalo e de proporção (KAN, 2002). Além dessas, Fenton e Bieman (2014) e Haug, Olsen e Bergman (2001) reconhecem a existência da escala absoluta, todas descritas a seguir:

- **Escala nominal:** representa o modo mais primitivo de medição e propõe classificar elementos em categorias, sem considerar qualquer hierarquia de ordenação entre elas. Exemplos: sexo feminino ou masculino; e cor do cabelo.
- **Escala ordinal:** trata-se da escala nominal incrementada com mais informações que possibilitam que as categorias possam ser comparadas em uma determinada ordem. Exemplos: ordem de chegada em uma corrida; e categorias de classes sociais.

- **Escala de intervalo:** trata-se da escala ordinal incrementada com o conceito de distância, exigindo que a distância entre duas categorias consecutivas seja igual. Exemplos: escala de temperatura Fahrenheit; e escala de temperatura Celsius.
- **Escala de proporção:** semelhante à escala de intervalo, porém incluindo um ponto zero, representando a ausência total do atributo sendo medido. Exemplos: tempo para completar uma tarefa; e tamanho e/ou peso de um objeto.
- **Escala absoluta:** é a escala mais completa e mais restritiva, com a medição sendo realizada pela contagem simples do número de elementos em um conjunto de entidades, ou seja, “número de ocorrências de X na entidade”. Exemplos: número da ocorrência de defeitos em um software; e número de alunos em uma sala de aula.

2.1.3 Classificação das medidas de software

As várias medidas existentes para medir um software podem ser categorizadas, quanto sua natureza, em duas principais categorias, conforme descritas a seguir:

Medidas orientadas a tamanho: representam medições diretas do tamanho de artefatos de software associados ao processo pelo qual o software é desenvolvido (VASCONCELOS, 2005). Elas visam medir o tamanho do software, por exemplo, sob a ótica do volume das linhas de código existentes. Portanto, elas são intrinsecamente ligadas à implementação e ao fator tecnológico. Um exemplo típico é a medida LOC (*Lines of Code*), que, a partir das linhas de código de determinado software, fornece um tamanho em linhas de código. Essas medidas são objetivas e facilmente calculadas, porém, por estarem vinculadas à tecnologia usada, podem levar a variações significativas quando comparadas a outras tecnologias.

Medidas orientadas a função: representam medições que buscam quantificar as funcionalidades oferecidas pelo software (VASCONCELOS, 2005). Sua principal representante é APF, proposta por Albrecht (1979), que representa um marco para as medidas orientadas a função. Desde então, surgiram outras medidas orientadas a função, tais como: Mark II, Pontos de Caso de Uso, NESMA, COSMIC-FPP, entre outras. Esse tipo de medida, quando comparado às medidas orientadas a tamanho, apresenta como vantagem a independência da tecnologia usada, que medem o software sob a ótica dos requisitos de

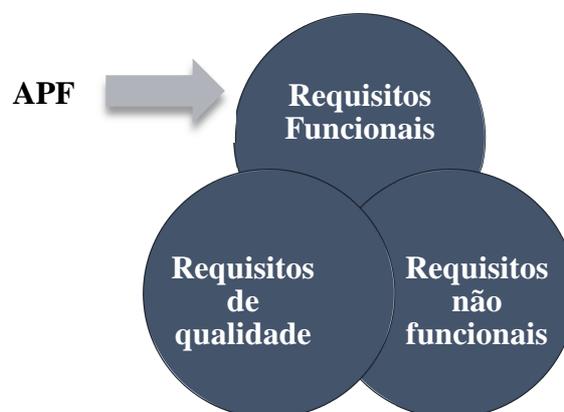
negócio. Além disso, elas fornecem um tamanho independente da produtividade do software e/ou da tecnologia.

2.2 Análise de pontos de função

A contagem de pontos de função, como medida para quantificar os requisitos funcionais do usuário de um software, foi formulada por Albrecht (1979) em seu trabalho “*A New Way of Looking at Tools*”, como resultado de um projeto na empresa IBM. Em 1986, foi criado o IFPUG (*International Function Point User Group*), uma organização sem fins lucrativos, com a missão de promover a gestão efetiva do desenvolvimento e da manutenção de software por meio de APF. Além de formular um manual para a contagem de pontos de função – o com CPM (*Counting Practice Manual*), o IFPUG certifica profissionais para realizar essa atividade.

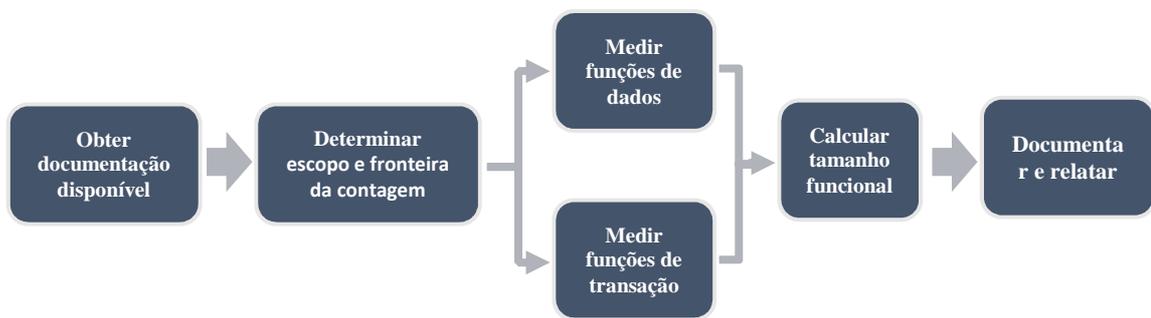
APF é uma medida orientada a função usada para medir os requisitos funcionais do usuário, independentemente do modo como esses requisitos são especificados. Os requisitos não funcionais, embora também façam parte do processo de engenharia de software, e sejam especificados pelos usuários, não são medidos diretamente por APF, conforme destacado na figura 1. APF usa dois tipos de escalas de medição: a escala ordinal, para medir os requisitos funcionais do usuário e a escala de proporção (ver seção 2.1.2), para determinar a influência dos requisitos não funcionais no tamanho medido.

Figura 1 – Requisitos medidos por APF



APF considera, na medição, as funcionalidades a serem implementadas no software (ou implementadas), que o usuário solicita e recebe, independentemente de linguagem de programação e da tecnologia usada para implementação. APF é aplicada de acordo com um conjunto de regras e de procedimentos definidos no CPM (IFPUG, 2010). Para a contagem de pontos de função, os procedimentos apresentados na figura 2 devem ser realizados. Os procedimentos são descritos de modo resumido nas próximas seções.

Figura 2 – Procedimento para a contagem de pontos de função via APF



Fonte: IFPUG (2010)

2.2.1 Obter documentação disponível

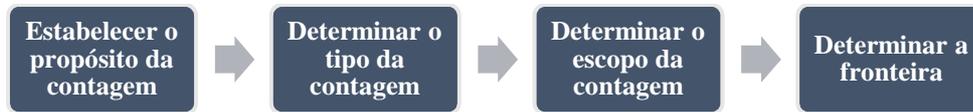
O primeiro passo é obter a documentação do software a ser medido, conforme disponível no projeto. Essa documentação deve descrever as funcionalidades entregues ou impactadas no software, podendo incluir: requisitos, modelos de dados ou de objetos, diagramas de classe, diagramas de fluxo de dados, diagramas e especificações de casos de uso, descrições procedimentais, esquemas de relatórios e de telas, manuais de usuário etc. (IFPUG, 2010).

2.2.2 Determinar escopo e fronteira da contagem

Para determinar o escopo e as fronteiras, as atividades na figura 3 devem ser realizadas. Primeiramente, o propósito da contagem deve ser estabelecido. Conforme IFPUG (2010), “o propósito da contagem representa a razão para a contagem de pontos de função”. Antes de realizar uma contagem, a necessidade exata de realizá-la deve ser definida.

Organizações não investiriam em uma contagem sem a existência de um motivo relevante para isso.

Figura 3 – Procedimento para determinar o escopo e a fronteira da contagem



Fonte: IFPUG, 2010

Depois, o tipo de contagem deve ser determinado, conforme um dos três tipos a seguir:

- **Projeto de desenvolvimento:** para desenvolver a primeira versão de um software.
- **Projeto de melhoria:** para alterar um software existente, mantendo-o funcional em um ambiente em alteração.
- **Contagem de aplicação:** Também chamado de *baseline* ou tamanho funcional instalado, fornece uma medida de funções atuais que o software fornece ao usuário.

O tipo de contagem é determinado baseado em seu propósito. Se o propósito da contagem for, por exemplo, fornecer o tamanho funcional da evolução de um software para, na sequência, derivar o custo ou o esforço do desenvolvimento dessa evolução, então uma contagem do tipo “projeto de melhoria” deve ser usada.

Na sequência, o escopo da contagem deve ser determinado. Segundo IFPUG (2010), o escopo “define o conjunto de requisitos funcionais do usuário a ser incluído na contagem de pontos de função”. O escopo pode compreender todas as funcionalidades do software ou apenas parte delas. O propósito da contagem determina o escopo. Por exemplo, caso o propósito da contagem seja “definir o custo para se desenvolver um pacote de software”, então todas as funcionalidades a serem disponibilizadas pelo pacote devem ser consideradas.

Por fim, a fronteira do software deve ser determinada. Segundo IFPUG (2010), a fronteira “é a interface conceitual entre o software sendo medido e seus usuários”. Ela é definida ao estabelecer um limite lógico entre o software em questão, seus usuários e os demais softwares com os quais há interação.

2.2.3 Medir funções de dados

Uma função de dados se refere aos requisitos funcionais do usuário sobre armazenar ou referenciar dados (IFPUG, 2010). Funções de dados são responsáveis pela retenção persistente de dados significativos aos usuários. A figura 4 ilustra as atividades a serem executadas para medir as funções de dados existentes dentro da fronteira do software.

Figura 4 – Procedimento para medir as funções de dados



Fonte: IFPUG, 2010

Inicialmente, as funções de dados devem ser identificadas por meio das seguintes ações:

1. Identificar todos os dados ou informações de controle reconhecidas pelo usuário, logicamente relacionadas no escopo da contagem.
2. Excluir entidades não mantidas por qualquer software.
3. Agrupar, em arquivos lógicos, entidades relacionadas dependentes.
4. Excluir entidades referenciadas como dados de código⁵.
5. Excluir entidades que não contenham atributos exigidos pelo usuário.
6. Remover entidades associativas que contenham atributos adicionais não exigidos pelo usuário e entidades associativas que contenham apenas chaves estrangeiras; e agrupar os atributos chave estrangeira com as entidades principais.

Assim, são consideradas funções de dados apenas entidades: reconhecidas pelo usuário (item “1”); com as quais o usuário interage por meio de funcionalidades (item “2”); e que não sejam resultado da implementação realizada (itens “4”, “5” e “6”). Após excluir todas

⁵ Entidades criadas em um modelo lógico de dados para garantir a qualidade dos dados. Dados de código fornecem uma lista de valores válidos que um atributo rótulo de entidade pode ter. Tipicamente, os atributos de dados de código são código, descrição ou outros atributos “padrão” que descrevem o código; ex.: abreviação padrão, data efetiva, data de término, dados de trilhas de auditoria, etc. O usuário nem sempre especifica explicitamente os dados de código, sendo algumas vezes referenciados como lista de dados ou dados de tradução. Em outros casos, ele é identificado pelo desenvolvedor em resposta a um ou mais requisitos técnicos do usuário (IFPUG, 2010).

as entidades não reconhecidas pelo usuário, as entidades restantes devem ser logicamente agrupadas (item “3”). Por exemplo, se determinados dados são criados ou excluídos em conjunto, então há uma forte indicação de que eles não existem de modo independente e, portanto, devem ser agrupados. As entidades devem ser agrupadas quando forem logicamente dependentes entre si, ou seja, quando uma não possui significado para o negócio sem a ocorrência de outra entidade a ela vinculada. Por exemplo, as entidades “empregado” e “dependente” ilustram tal situação. Se, ao se excluir um empregado, uma regra de negócio define que todos os dependentes a ele relacionados devem ser também excluídos, então “dependente” não possui significado independente para o negócio; assim, ambas as entidades devem ser agrupadas e consideradas como uma única função de dados.

Em seguida, as funções de dados agrupadas devem ser classificadas como: Arquivos Lógicos Internos (ALI) ou Arquivos de Interface Externa (AIE), de acordo com (IFPUG, 2010):

- **ALI:** função de dados mantida pelo próprio software;
- **AIE:** função de dados apenas referenciada (não mantida) pelo software (classificada como ALI para outros softwares com o qual o software sendo medido interage).

Na sequência, a complexidade funcional de cada função de dados deve ser determinada. Para isso, primeiramente o número de Dados Elementares Referenciados (DER) e de Registros Lógicos Referenciados (RLR) para cada função de dados devem ser calculados. A tabela 1 apresenta a complexidade das funções de dados, de acordo com os números de DERs e RLRs.

Tabela 1 – Complexidade de funções de dados

	1 a 19 DERs	20 a 50 DERs	50 ou mais DERs
1 RLR	Baixa	Baixa	Média
2 a 5 RLRs	Baixa	Média	Alta
6 ou mais RLRs	Média	Alta	Alta

Fonte: IFPUG, 2010.

Para calcular o número de DERs de uma função de dados, todos os atributos únicos das entidades agrupadas que são reconhecidos pelo usuário e não repetidos devem ser

considerados (IFPUG, 2010). As funções de dados, por representarem um agrupamento lógico de entidades, tal como percebido pelo usuário, comumente contém várias entidades agrupadas em um único arquivo lógico. Geralmente, as entidades agrupadas dentro de uma função de dados são denominadas RLRs da função de dados. Segundo IFPUG (2010), os RLRs são “um subgrupo de elementos de dados reconhecido pelo usuário em um ALI ou um AIE”.

Por fim, a contribuição de cada função de dados, em termos de tamanho funcional, deve ser determinada com base nos dados da tabela 2, que contém o total de pontos de função a ser atribuído a cada nível de complexidade.

Tabela 2 – Tamanho funcional de funções dados por grau de complexidade

Grau de complexidade funcional	Pontos de função	
	ALI	AIE
Baixo	7	5
Médio	10	7
Alto	15	10

Fonte: IFPUG, 2010.

2.2.4 Medir funções de transação

Uma função de transação se refere à funcionalidade fornecida ao usuário para o processamento de dados por um software. A figura 5 ilustra as atividades a serem executadas para medir as funções de transação existentes dentro da fronteira do software.

Figura 5 – Procedimento para medir as funções de transação



Fonte: IFPUG, 2010

Inicialmente, os processos elementares do software devem ser identificados. Conforme IFPUG (2010), um processo elementar “(i) é a menor unidade de atividade significativa para os usuários; (ii) constitui uma transação completa; (iii) é autocontido; e (iv) deixa o negócio

do software em um estado consistente”. Todo processo elementar é composto por um conjunto de lógicas de processamento, que são: “qualquer requisito especificamente solicitado pelo usuário para completar um processo elementar, tais como validações, algoritmos ou cálculos, e leitura ou manutenção de uma função de dados” (IFPUG, 2010). Lógicas de processamento permitem identificar os procedimentos necessários para executar um processo elementar. Existem 13 lógicas de processamento reconhecidas pelo CPM 4.3.1, considerando: validações efetuadas; cálculos matemáticos efetuados; valores equivalentes convertidos; dados filtrados e selecionados por critérios específicos para comparar vários grupos de dados; condições analisadas para determinar quais se aplicam; pelo menos um ALI atualizado; pelo menos um ALI ou um AIE referenciado; dados ou informações de controle recuperados; dados derivados criados; comportamento do software alterado; informações preparadas e apresentadas para fora da fronteira; dados ou informações de controle aceitos entrando pela fronteira do software; e, dados reclassificados ou reorganizados (IFPUG, 2010).

Para identificar os processos elementares, os requisitos funcionais do usuário devem ser observados para delimitar as menores unidades de atividades que satisfaçam as outras três restrições apresentadas na definição de processo elementar acima. Por exemplo, em uma analogia com uma situação do mundo real: “quando se deseja lavar um quintal, é necessário molhá-lo, ensaboar e esfregar o chão, enxaguá-lo e secá-lo”. Quaisquer das atividades mencionadas acima que sejam executadas independentemente não satisfariam a tarefa “lavar o quintal”, sendo esta, portanto, a menor unidade possível de atividade. Apesar disso, a tarefa “lavar o quintal” poderia ser interpretada de modo diferente por usuários ou tipos de negócio diferentes; de modo que identificar processos elementares está, na verdade, vinculada ao entendimento de um determinado usuário acerca de um determinado negócio.

Após identificar os processos elementares do software, apenas processos elementares únicos (ou seja, processos não duplicados entre si) devem ser classificados como função de transação. Portanto, cada processo elementar identificado deve ser comparado com todos os outros processos elementares identificados para verificar se contém os mesmos DERs, Arquivos Lógicos Referenciados (ALR⁶) e lógicas de processamento. Caso possuam, então eles devem ser classificados como uma única função de transação; caso contrário, cada processo elementar deve ser classificado com uma função de transação independente.

Cada função de transação deve então ser classificada em “Entrada Externa (EE)”,

⁶ Os ALRs são descritos oportunamente ainda nesta seção.

“Consulta Externa (CE)” ou “Saída Externa (SE)”. Conforme IFPUG (2010):

- **EE:** processa dados ou informações de controle que vêm de fora da fronteira do software; sua intenção primária é dar manutenção a um ou mais ALIs ou alterar o comportamento do software.
- **CE:** envia dados ou informações de controle para fora da fronteira do software; sua intenção primária é apresentar informações ao usuário via recuperação de dados ou de informações de controle.
- **SE:** envia dados ou informações de controle para fora da fronteira do software, incluindo um processamento adicional ao de uma CE; sua intenção primária é apresentar informações ao usuário via lógica de processamento que não seja apenas recuperação de dados ou de informações de controle.

Na sequência, a complexidade funcional de cada função de transação deve ser determinada. Para isso, primeiramente, o número de ALRs e DERs para cada função de transação deve ser calculado. As tabelas 3 e 4 apresentam as complexidades das funções de transação, de acordo com o número de DERs e ALRs, dependendo do tipo: EE ou CE e SE.

Para calcular o número de DERs de uma função de transação, todos os atributos únicos que cruzam a fronteira do software, que são reconhecidos pelo usuário e não repetidos devem ser considerados (IFPUG, 2010). Para cada função de transação, funções de dados podem precisar ser referenciadas ou mantidas; cada entidade referenciada ou mantida durante uma função de transação é denominada um ALR.

Tabela 3 – Complexidade de funções de transação (tipo EE)

	1 a 4 DERs	5 a 15 DERs	16 ou mais DERs
0 a 1 ALRs	Baixa	Baixa	Média
2 ALRs	Baixa	Média	Alta
3 ou mais ALRs	Média	Alta	Alta

Fonte: IFPUG, 2010

Por fim, a contribuição de cada função de transação, em termos de tamanho funcional, deve ser determinada com base nos dados das tabelas 5 ou 6, que contém o total de pontos de função a ser atribuído a cada nível de complexidade, dependendo do tipo: EE e CE, ou SE.

Tabela 4 – Complexidade de funções de transação (tipos SE e CE)

	1 a 5 DERs	6 a 19 DERs	20 ou mais DERs
0 a 1 ALRs	Baixa	Baixa	Média
2 a 3 ALRs	Baixa	Média	Alta
4 ou mais ALRs	Média	Alta	Alta

Fonte: IFPUG, 2010

Tabela 5 – Tamanho funcional de funções de transação (tipos EE e CE) por grau de complexidade

Grau de complexidade funcional	Pontos de função
Baixo	3
Médio	4
Alto	6

Fonte: IFPUG, 2010

Tabela 6 – Tamanho funcional de funções de transação (tipo SE) por grau de complexidade

Grau de complexidade funcional	Pontos de função
Baixo	4
Médio	5
Alto	7

Fonte: IFPUG, 2010

2.2.5 Calcular tamanho funcional

Para o cálculo final do tamanho funcional, as seguintes fórmulas devem ser aplicadas, dependendo do tipo de contagem: (1) projeto de desenvolvimento; (2) projeto de aplicação; ou (3) projeto de melhoria (IFPUG, 2010).

$$DFP = ADD + CFP, \text{ onde} \quad (1)$$

- **DFP** (*Development Functional Point*) é a contagem de pontos de função do software de “projeto de desenvolvimento”.

- **ADD** (*ADDED*) é o tamanho resultante da soma das funções de dados e das funções de transação a serem entregues ao usuário pelo projeto de desenvolvimento.
- **CFP** (*Conversion Functionality Point*) é o tamanho resultante da soma das funções de dados e das funções de transação para converter dados (por exemplo: migração de dados de aplicações antigas para novas).

$$AFP = ADD, \text{ onde} \quad (2)$$

- **AFP** (*Application Function Point*) é a contagem de pontos de função do software de “projeto de aplicação”.
- **ADD** (*ADDED*) é o tamanho resultante da soma das funções de dados e das funções de transação sendo desenvolvidas para a primeira versão do software.

$$EFP = ADD + CHGA + CFP + DEL, \text{ onde} \quad (3)$$

- **EFP** (*Enhancement Function Point*) é a contagem de pontos de função do software de “projeto de melhoria”.
- **ADD** (*ADDED*) é o tamanho resultante da soma das funções de dados e das funções de transação a serem incluídas no software pelo projeto de melhoria.
- **CHGA** (*CHAnGed After*) é o tamanho resultante da soma das funções de dados e das funções de transação a serem alteradas no software pelo projeto de melhoria.
- **CFP** (*Conversion Functionality Point*) é o tamanho resultante da soma das funções de dados e das funções de transação para converter dados (por exemplo: migração de dados de aplicações antigas para novas).
- **DEL** (*DELETED*) é o tamanho resultante da soma das funções de dados e das funções de transação a serem excluídas do software pelo projeto de melhoria.

A aplicação das fórmulas apropriadas resulta em um número de pontos de função, que demonstra o **tamanho funcional não ajustado** do software, ou seja, que não considera qualquer influência tecnológica (IFPUG, 2010). Esse resultado pode ser opcionalmente ajustado considerando a influência de requisitos não funcionais no tamanho funcional. Para isso, o tamanho funcional calculado deve ser multiplicado por um Fator de Ajuste de Valor (VAF – *Value Adjustment Factor*), que é calculado baseado em 14 Características Gerais do Sistema (CGS), relacionadas a: comunicação de dados; processamento distribuído; desempenho; configuração intensamente usada; volume de transações; entrada de dados on-line; eficiência do usuário final; atualização on-line; processamento complexo; reusabilidade; facilidade de instalação; facilidade de operação; múltiplos locais; e facilidade de mudança.

Para cada CGS, o contador deve pontuar seu Grau de Influência (DI – *Degree of Influence*) no projeto de 0 a 5, de acordo com descrições específicas descritas no CPM 4.3.1. A soma de todos os DIs, das 14 CGSs, leva ao o TDI (*Total Degree of Influence*), usado no cálculo do VAF, de acordo com a fórmula (4).

$$\text{VAF} = (\text{TDI} * 0.01) + 0.65, \text{ onde} \quad (4)$$

- **VAF** (*Value Adjustment Factor*) é o fator de ajuste a ser realizado no tamanho funcional do software considerando a influência de requisitos não funcionais.
- **TDI** (*Total Degree Of Influence*) é o somatório dos graus de influência definido para cada uma das 14 CGSs.

2.2.6 Documentar e relatar

A contagem de pontos de função deve ser documentada com as seguintes informações, nesta ordem (IFPUG, 2010): (i) propósito e tipo da contagem; (ii) escopo da contagem e fronteira do software; (iii) data da contagem; (iv) lista das funções de dados e funções de transação consideradas, incluindo tipos, complexidades e números de pontos de função atribuídos; (v) resultado da contagem; e (vi) quaisquer suposições feitas e questões resolvidas.

A documentação assegura a rastreabilidade do tamanho funcional encontrado ao final da contagem e relata possíveis premissas adotadas e suposições realizadas. Além disso, a documentação torna a medida mais consistente e pode indicar possíveis melhorias a serem realizadas na documentação de apoio fornecida (IFPUG, 2010).

Premissas adotadas, quando bem discriminadas, podem evidenciar possíveis erros de interpretação individual dos insumos disponibilizados para a contagem. Sem a correta documentação dessas informações, o tamanho funcional se torna apenas um número, sem agregar, necessariamente, informações valiosas ao processo de medição (IFPUG, 2010).

2.3 Modelagem conceitual baseada em engenharia de requisitos

Esta seção apresenta a Modelagem Conceitual baseada em Engenharia de Requisitos (MCbER) (INSFRÁN, PASTOR E WIERINGA, 2002), cuja uma de suas partes é usada para embasar a abordagem proposta neste trabalho. MCbER inclui técnicas para especificar claramente requisitos funcionais do software, via decomposição sistemática dos requisitos a partir do mais alto nível até uma especificação mais detalhada. Ela é composta por três fases:

- Modelagem de requisitos: cujo propósito é capturar de modo preciso o que o usuário deseja construir no software.
- Esquema conceitual: cujo propósito é representar os requisitos do usuário em uma especificação descrevendo o que o software faz, como se houvesse uma tecnologia perfeita disponível para implementar o software.
- Modelo de execução: cujo propósito é obter uma aplicação específica em um ambiente de desenvolvimento imperativo de software.

Na modelagem de requisitos, três técnicas são combinadas: (i) Declaração de missão do software; (ii) Árvore de refinamento de funções; e (iii) Modelo de caso de uso:

- i. **Declaração da missão do software**: descreve as funcionalidades necessárias ao software no nível mais alto de abstração. Ela demonstra a razão pela existência do software e o associa aos requisitos de negócio, resumindo a razão pela qual algumas partes interessadas, do ambiente de negócios, dispenderiam recursos para sua aquisição. É constituída por uma descrição de uma frase seguida por uma descrição das principais responsabilidades do software. Opcionalmente, ela pode indicar uma lista de atividades que o software não desempenhará (WIERINGA, 1998).
- ii. **Árvore de refinamento de funções**: é uma técnica para análise de funcionalidades disponibilizadas por um software. A Árvore de refinamento de funções é uma lista hierárquica de funções desejadas do software. Essas funções do software podem ser refinadas uma ou mais vezes, em diferentes níveis de abstração, para um mesmo software. Todos os nós da árvore devem se relacionar entre si para justificar a existência de todos seus níveis. Os níveis superiores explicam por que uma função de nível inferior deve estar presente e as funções de nível inferior explicam o que uma função de nível superior faz para o ambiente (WIERINGA, 1998).
- iii. **Modelo de caso de uso**: contempla o Diagrama de casos de uso para demonstrar a comunicação entre o ambiente (atores) e o software, e a especificação dos casos de uso para especificar com detalhes as interações externas.

Segundo Insfrán, Pastor e Wieringa (2002), a combinação da Declaração da missão do software com a Árvore de refinamento de funções foi proposta com o intuito de fornecer um bom nível de abstração para o Modelo de casos de uso. Dentre essas técnicas, a Árvore de refinamento de funções é usada para embasar a abordagem proposta neste trabalho.

A Árvore de refinamento de funções possui a seguinte estrutura (WIERINGA, 1998; INSRÁN, PASTOR E WIERINGA, 2002):

- **Nível 0 – Raiz da árvore:** contém a declaração da “missão” do software, o que demanda um alto grau de abstração de suas funcionalidades.
- **Nível 1 – Nós intermediários:** representam “tipos de atividade” realizadas pelo software ou “áreas de negócio” em que o software atua, o que demanda um grau de abstração médio sobre como as funcionalidades do software devem ser agrupadas.
- **Nível 2 – Nós folhas:** representam “funções elementares” do software, ou seja, aquelas iniciadas por eventos enviados por seus usuários ou pela ocorrência de eventos temporais, o que demanda um baixo grau de abstração, comparado com os anteriores. Uma função é dita elementar quando é iniciada por um evento enviado pelo usuário (ator) ou pela ocorrência de um evento temporal.

Um exemplo de uma Árvore de refinamento de funções é exibido na figura 6. Os requisitos funcionais do software, usados como base para a elaboração da Árvore de refinamento de funções é descrito no apêndice A.

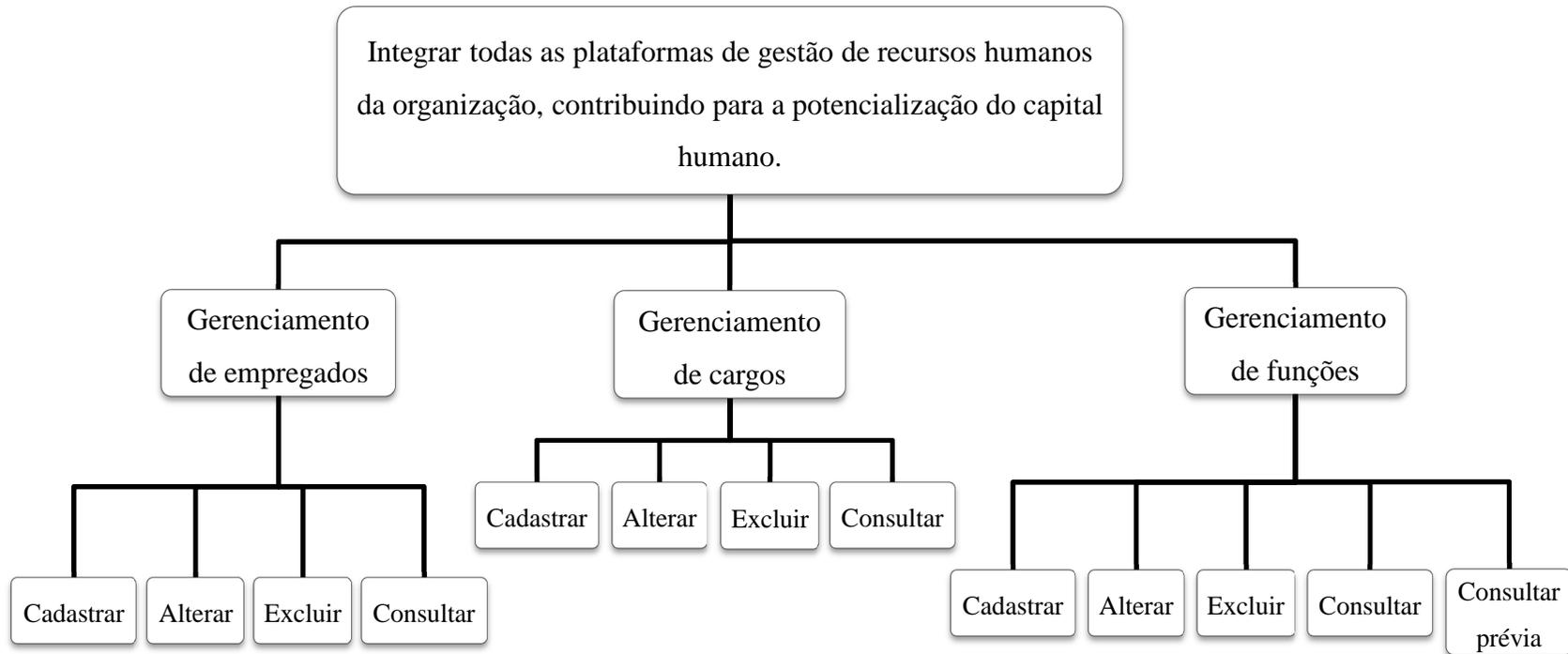
Funções representadas nos nós folha devem ser posteriormente detalhadas por meio de especificações de caso de uso. Os proponentes da abordagem MCbER propõem um modelo específico para isso.

MCbER foi usada para embasar uma abordagem para a contagem de pontos de função, proposta por Abrahão e Insfrán (2008). Porém, essa abordagem apresenta problemas que a abordagem proposta neste trabalho visa solucionar, conforme destacados na seção 3.3. Assim, existem distinções significativas entre a abordagem proposta neste trabalho e aquela proposta por Abrahão e Insfrán (2008), conforme descrito na seção 5.1.

2.4 Considerações finais

Neste capítulo, foram apresentados os principais aspectos referentes a medição de software, com enfoque em APF. Os conceitos acima destacados são essenciais para o entendimento integral do trabalho realizado. No próximo capítulo são apresentados os trabalhos desenvolvidos que estão relacionados com essa pesquisa, de modo que seja possível situar este trabalho com pesquisas relacionadas.

Figura 6 – Exemplo da Árvore de Refinamento de Funções do SISRH⁷



Fonte: Marcos Freitas Junior, 2015

⁷ O desnível das funcionalidades relacionadas ao “Gerenciamento de cargos” foi realizado apenas para melhorar a visualização desta figura.

3. Trabalhos relacionados

Este capítulo analisa trabalhos anteriores que abordaram problemas relacionados ao uso de APF e apresenta sugestões de melhoria para solucioná-los. Os problemas decorrentes da existência de subjetividade na aplicação das regras da técnica foram comentados mais detalhadamente, pois estão diretamente ligados ao contexto da abordagem proposta neste trabalho.

Embora o objetivo deste trabalho seja desenvolver uma abordagem automatizável, são apresentados aqui trabalhos que não lidam necessariamente com abordagens automatizadas, já que elas também poderiam conter informações úteis para esse contexto.

As próximas seções abordam os seguintes itens: problemas relacionados ao uso de APF; melhorias propostas para diminuir a subjetividade de APF; e análise das melhorias propostas.

3.1 Problemas relacionados ao uso de APF

Uma revisão sistemática da literatura (cujo método empregado está apresentado no capítulo 4) foi conduzida para identificar trabalhos que apresentassem problemas relacionados ao uso de APF. Os itens seguintes trazem um resumo dos problemas encontrados:

1. **Regras que requerem interpretação individual:** o uso de APF é subjetivo, requerendo o julgamento individual por parte dos contadores, o que impossibilita sua aplicação de modo padronizado. Isso ocorre devido à ausência de regras bem definidas, claras e sistemáticas no CPM (HARPUT; KAINDL; KRAMER, 2005; LAVAZZA; BIANCO; CARAVAGLIA, 2008; RAO et al., 2008; UEMURA; KUSUMOTO; INOUE, 2001; KUSUMOTO et al., 2002; ABRAHÃO; INSFRÁN, 2008; DEL BIANCO; LAVAZZA; MORASCA, 2012; LAVAZZA; MORASCA; ROBILOLO, 2013). Em estudos anteriores, Kemerer (1993) relatou uma variação de 12% entre contadores que mediram o mesmo software na mesma organização; Low e Jeffery (1990) relataram uma variação de 30% entre contadores da mesma organização.

2. **Aplicação tardia:** Arnuphaptrairong (2013) considera que, para aplicar APF, exige-se uma grande quantidade de informações, sendo possível usá-las apenas quando a maior parte dos artefatos já tenha sido desenvolvida. Assim, a utilidade de APF seria baixa quando aplicada em uma fase avançada do ciclo de vida, durante a qual não é tão importante realizar estimativas.

3. **Complexidades e números de pontos de função imprecisos:** falhas na quantificação correta de funções de dados e funções de transação foram relatadas. O modo de estabelecer complexidades e números de pontos de função (conforme tabelas 1 a 5) não é adequado, já que funções com variadas combinações de DERs e RLRs/ALRs podem receber complexidades iguais. Além disso, em alguns casos, funções com DERs e ALRs/RLRs parecidos (com diferença de apenas uma unidade, por exemplo) podem ser classificadas com complexidades diferentes e, conseqüentemente, receber tamanhos diferentes (CHEN et al., 2010; YA-FANG et al., 2010; KRALJ et al., 2005; MARZOUGHY et al., 2010; RAO et al., 2011; XIA et al., 2008).

4. **Independência tecnológica:** APF está desatualizada frente a novas tecnologias. Para Calazans et al. (2004) e Gerlec, Krajnc e Goljat (2010), elas possuem particularidades específicas que deveriam ser consideradas nas medições atuais; em 1979, isso não foi feito devido ao contexto tecnológico do período. Ferreira e Marques-Neto (2013) destacam que *data marts* são exemplos de software que possuem características tecnológicas específicas que deveriam ser consideradas. Por outro lado, Xia et al. (2008) e Ya-fang et al. (2010) acreditam que APF não é mais apropriada para aumentar a acurácia de estimativas por ser antiga e não refletir o atual processo de engenharia de software, incluindo orientação a objetos.

5. **CGSs desatualizadas e de difícil uso:** para Bharadwaj e Nair (2009), o tamanho funcional ajustado não considera requisitos não-funcionais importantes para o contexto atual, tais como eficiência, usabilidade, manutenibilidade e portabilidade. Por outro lado, a CGS “volume de transações” não é mais aplicável atualmente devido à evolução tecnológica, altas taxas de velocidade de internet, alta velocidade de acesso em disco e CPUs com processamento MHZ alto. Abdulah et al. (2009) acreditam que as CGS não contemplam características para atender requisitos não-funcionais focadas no contexto de segurança de software, como criptografia, troca de dados sensíveis e proteção da tecnologia de segurança. De acordo com Matijevic et al. (2012), determinar o grau de influência de cada CGS é uma tarefa difícil e, na prática, o contador não consegue identificar com facilidade esses valores. Considerando que um requisito não-funcional pode influenciar a complexidade de um projeto de engenharia de software mais que um requisito funcional, esse problema deveria ser resolvido.

6. **Aplicação lenta, cara e dependente de especialistas:** APF é aplicada muito lentamente e exige profissionais especialistas, o que encarece seu uso (LAVAZZA; BIANCO;

CARAVAGLIA, 2008; FRATERNALI; BONGIO, 2006; ADEM; KASIRUN, 2010; DEL BIANCO; LAVAZZA; MORASCA, 2012; LAVAZZA; MORASCA; ROBIOLO, 2013).

7. **Domínio original de software obsoleto:** APF foi concebida para softwares eletrônicos de processamento de dados (EDPS – *Electronic Data Processing Systems*), capturando adequadamente o tamanho funcional das funções de dados e das funções de transação. Porém, a técnica não foi adaptada para representar a complexidade de outros tipos atuais de software, tais como os embarcados e os de tempo real (LAVAZZA; CARAVAGLIA, 2009).

8. **Uso inconsistente de escalas de medição:** Fenton e Bieman (2014) argumentam que APF combina de maneira inconsistente várias escalas de medição, considerando a teoria de medição. Enquanto os graus de complexidade e a classificação de CGS estão em escala ordinal, a contagem de pontos de função é baseada em escala de proporção. De acordo com esses autores, essa combinação linear não tem sentido do ponto de vista matemático.

3.2 Melhorias propostas para diminuir a subjetividade de APF

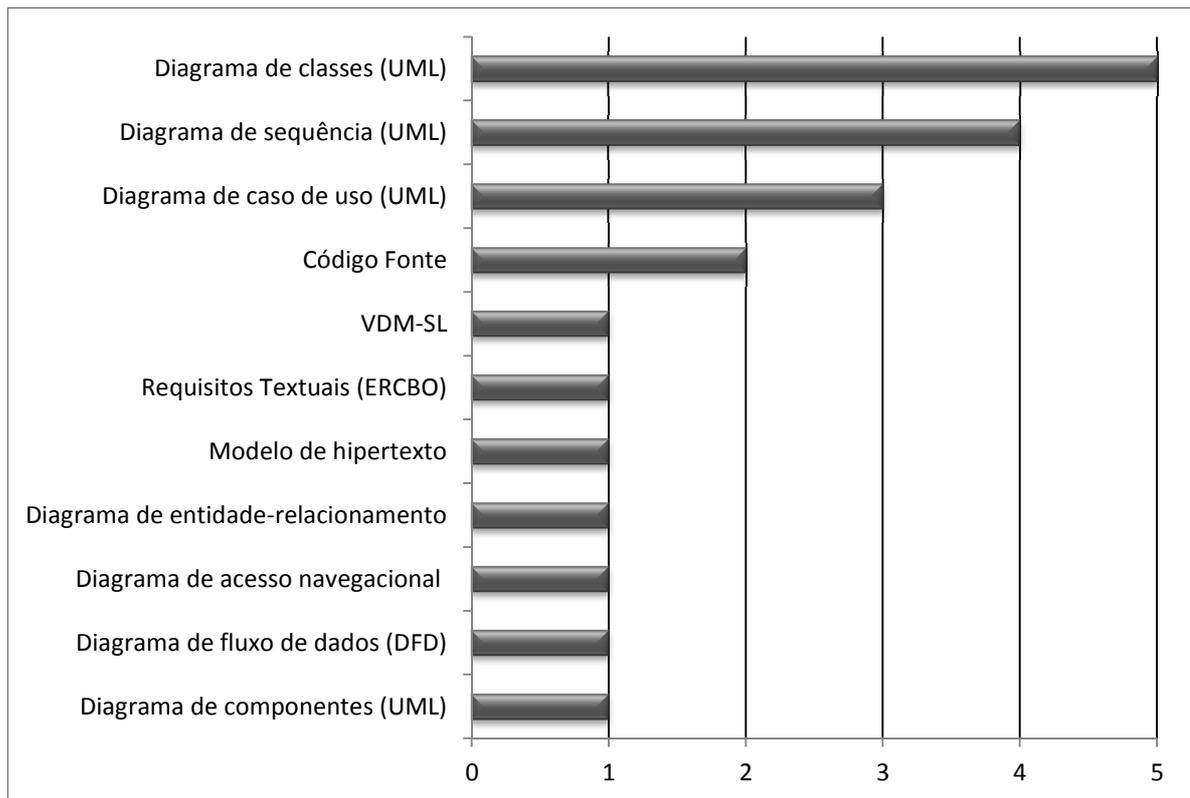
Considerando o principal problema relatado para APF, “1 – Regras que requerem interpretação individual”, esta seção apresenta um resumo das melhorias sugeridas para tentar mitigá-lo. As propostas dividem-se em dois tipos:

1. Associação de conceitos de APF a artefatos produzidos no ciclo de vida de software, incluindo aqueles produzidos pelas seguintes técnicas: diagramas UML, ERbCO, especificação VDM-SL, MCbER, código fonte, ER-DFD, WebML e Modelo Conceitual OO-H.
2. Diretrizes para a decodificação dos conceitos de APF a partir de artefatos elaborados em tecnologias como *data warehouse* (FERREIRA; MARQUES-NETO, 2013).

A maior parte das abordagens propostas para mitigar esse problema usa artefatos produzidos no ciclo de vida de software a fim de tornar a medição mais sistemática, já que os contadores podem usar um direcionamento padronizado para encontrar as informações necessárias para a contagem, o que torna as interpretações individuais em procedimentos consistentes e repetíveis. Além disso, por serem sistemáticos, esses mapeamentos podem apoiar a automação do processo de medição. O gráfico 1 apresenta a proporção de artefato produzidos por técnicas de modelagem de software usadas nos mapeamentos propostos. Diagramas UML representam a maior parte, o que pode ser explicada por: (i) eles

potencialmente possuírem mais elementos adequados para representar informações necessárias à contagem de pontos de função ou (ii) serem mais conhecidos e usados no ambiente corporativo, o qual é o principal alvo de APF.

Gráfico 1 – Distribuição dos tipos de artefato produzidos por técnicas de modelagem usadas nas abordagens propostas para diminuir a subjetividade de APF



Fonte: Marcos Freitas Junior, 2015

A seguir, estão resumidas as propostas e análise de lacunas identificadas considerando as oito abordagens usadas nos mapeamentos propostos. Os quadros 13 e 14 (no apêndice B) apresentam resumidamente uma comparação entre as abordagens avaliadas nesta seção, considerando a completude de tais abordagens em relação à cobertura de passos de APF. Apenas os principais passos de APF foram considerados na comparação, ou seja, aqueles necessários para a contagem de funções de dados e funções de transação.

1. **Diagramas UML:** sete abordagens foram propostas para a contagem de pontos de função a partir de diagramas UML. Dentre elas, seis propõem regras para mapear elementos de diagramas UML para conceitos de APF e uma propõe incluir informações

complementares, via estereótipos, ao Diagrama de componentes, de modo que a modelagem do diagrama fosse realizada com auxílio de um especialista em APF (LAVAZZA; BIANCO; GARAVAGLIA, 2008). Em conjunto, as abordagens preveem a utilização de apenas quatro dos 14 diagramas que compõem a UML: Diagrama de casos de uso, Diagrama de sequência, Diagrama de classes e Diagrama de componentes. Os quadros 8, 9, 10 e 11 (do apêndice B) resumem as associações propostas entre elementos de diagramas UML e conceitos de APF.

Cada uma das sete abordagens apresentam a mesma limitação: a ausência de todas as informações necessárias para a aplicação das regras de APF original. Isso ocorre porque os Diagramas de casos de uso, de sequência, de classes e de componentes não são representados em um nível de abstração ideal e não contemplam todas as informações necessárias para a contagem. Por exemplo, os diferentes níveis de abstração possíveis para representar um caso de uso dificultam o estabelecimento de regras de conversão para conceitos de APF. Segundo Lavazza, Bianco e Caravaglia (2008), durante a modelagem de casos de uso para softwares complexos, pode ocorrer de um único caso de uso representar diversas transações em um nível de abstração maior do que o exigido para a contagem via APF, por exemplo. Quando acompanhados de uma especificação que ofereça mais detalhamento em relação a um caso de uso, esta é realizada em linguagem natural, de modo não sistemático, o que dificulta a sistematização da contagem. O Diagrama de sequência é representado em um nível de abstração menor do que o exigido para APF, já que são desenvolvidos a partir de casos de uso. Eles representam informações mais específicas (em menor nível de abstração) a respeito da interação entre objetos. O mesmo é observado para os Diagramas de classes e de componentes.

Além dos limitantes apresentados para cada diagrama constituinte da UML, destacam-se a seguir problemas adicionais encontrados em relação às abordagens que propõem a medição de APF a partir de diagramas UML: *(i)* ausência de padronização entre as abordagens no que diz respeito a quais diagramas são mais adequados para a medida APF. Por exemplo, para identificar as funções de dados de um software Rao; Nagaraj; Ahuja (2008) e Uemura; Kusumoto; Inoue (2001) propõem o uso de Diagrama de sequência, enquanto que Chamundeswari e Babu (2008) e Harput; Kaindl; Kramer (2005) propõem o uso de Diagrama de classes); *(ii)* ausência de padronização de regras para as abordagens que se baseiam nos mesmos diagramas UML. Por exemplo, Chamundeswari e Babu (2008) e Harput; Kaindl; Kramer (2005) propõem que as funções de dados sejam representadas por classes do Diagrama de classes, entretanto, para Uemura; Kusumoto; Inoue (2001) não existe uma relação direta entre classes e funções de dados; *(iii)* identificação imprecisa de constituintes de

APF baseados nos diagramas selecionados. Por exemplo, Rao; Nagaraj; Ahuja (2008) e Uemura, Kusumoto; Inoue (2001) usam as mensagens trocadas entre objetos em um Diagrama de sequência para identificar funções de dados, enquanto Rao; Nagaraj; Ahuja (2008) e Harput; Kaindl; Kramer (2005) usam essas mensagens para identificar funções de transação; (iv) simplificação de APF original (ver quadros 13 e 14 do apêndice B); e (v) manutenção da dependência de especialistas em contagem para a execução da mesma. Por exemplo, na abordagem de Lavazza; Bianco; Garavaglia (2008), foram necessários participantes com conhecimento das regras de APF na elaboração de diagramas UML.

2. **Descrição textual de requisitos funcionais:** uma abordagem propõe a contagem de pontos de função baseado na técnica de Elicitação de Requisitos baseada em Cenários e Objetivos (ERbCO), concebida originalmente por Kim, Park, Sugumaran (2004). ERbCO apresenta uma atividade complementar de levantamento de requisitos que se baseia na Análise dirigida a casos de uso. O objetivo dessa técnica é solucionar o problema da falta de apoio para um processo sistemático de elicitação de requisitos que resulta muitas vezes em um conjunto “ad hoc” de casos de uso gerados sem uma lógica explícita. ERbCO (assim como MCbER) propõe o desenvolvimento de níveis de abstração distintos para os cenários e objetivos, sendo eles: Nível de Negócio (NN), Nível de Serviço (NS), Nível de Interação (NI) e Nível Interno (NIN). NN identifica o propósito do software como um todo (por exemplo, “melhorar os serviços disponibilizados para os clientes bancários”); NS identifica os serviços que um software deve disponibilizar a uma organização (por exemplo, “retirar dinheiro”); NI prioriza as interações entre o software e seus agentes (por exemplo, “retirar dinheiro da ATM”); e NIN descreve as necessidades do software para realizar as interações selecionadas em NI. A todos os níveis são associados objetivos, definidos como “algo sobre qual a parte interessada (*stakeholder*) espera alcançar no futuro” e cenários, que representam “um possível comportamento limitado a um conjunto de interações intencionais que ocorrem entre vários agentes”. Para cada cenário, existe um conjunto de regras que descreve como os cenários devem ser especificados (por exemplo, “o cliente [agente] + deposita [verbo] + dinheiro [alvo] + no ATM [direção]”).

Na abordagem de Choi, Park e Sugumaran (2006) para o uso de ERbCO na contagem de pontos de função, funções de dados e funções de transação devem ser identificadas a partir do NI, sendo que “objetivos” representam funções de transação e “alvos” representem funções de dados. Essa abordagem foi implementada em uma ferramenta por Adem e Kasirun (2010), que também a compararam, via um experimento, à APF original. O experimento realizado, que considerou a medição de requisitos funcionais de um único software, apresentou um

coeficiente de variação de 2,35% do resultado produzido pela abordagem de Choi, Park e Sugumaran (2006) em relação à APF original.

Entretanto, o uso de ERbCO não é ideal para a contagem de pontos de função sem a incorporação de informações adicionais. Isso ocorre porque ERbCO não prevê um nível de abstração ideal para representar funcionalidades a serem desempenhadas pelo software, tal como exigido por APF (ver seção 2.2.4). Embora essa técnica defina níveis de abstração de modo mais objetivo que UML, há a possibilidade de um processo elementar ser representado tanto em NS quanto em NI. Com isso, o estabelecimento de regras de conversão para conceitos de APF a partir do ERbCO é dificultado, pois demanda regras adicionais para identificar os casos em que os processos elementares foram representados em nível de serviço ou em nível de interação. Além disso, ERbCO não contém todas as informações necessárias para a contagem de pontos de função, o que torna necessário o levantamento adicional de informações por parte do contador.

Além dos limitantes apresentados para ERbCO, destacam-se a seguir problemas adicionais encontrados em relação à abordagem de Choi, Park e Sugumaran (2006): (i) uso exclusivo do nível de interação para identificar funções de transação, já que a abordagem não previu regras para englobar possíveis processos elementares especificados em nível de serviço; (ii) classificação de funções de transação baseadas em uma lista de palavras pré-definidas, assim, caso o usuário especifique uma palavra que não esteja presente na lista, a abordagem não consegue identificar os tipos de função de transação; e (iii) simplificação de APF original (ver quadros 13 e 14 do apêndice B).

3. **Especificação VDM-SL:** uma abordagem propõe a contagem de pontos de função baseado em especificações VDM-SL (*Vienna Development Method-Specification Language*). VDM-SL é uma linguagem de especificação formal usada para modelar, analisar e desenvolver sequencialmente um software (FITZGEALD, 2009). Um modelo VDM-SL é constituído por: *tipos de dados*, que representa o universo de possíveis estados para um software; *invariantes*, que expressam propriedades que não devem mudar ao longo do tempo; *funções/operações*, pelas quais os dados são acessados e manipulados; *pré-condições*, que representam uma instrução ou um conjunto de declarações verdadeiras quando a operação é chamada (a operação não é garantida para executar como deveria se não satisfeitas as condições prévias); e *pós-condições*, que representam uma instrução ou um conjunto de declarações que serão verdadeiras quando a operação for concluída.

Miyawaki, Iijima e Ho (2008) propuseram um mapeamento entre conceitos de VDM-SL e conceitos de APF por meio de uma notação matemática. Esse mapeamento foi

implementado em uma ferramenta que recuperava e convertia automaticamente especificações VDM-SL em uma contagem de pontos de função. O experimento realizado, que considerou a medição de requisitos funcionais de um único software, apresentou um coeficiente de variação de 6,66% em relação à APF original.

Entretanto, o uso de VDM-SL não é ideal para a contagem de pontos de função sem a incorporação de informações adicionais. Isso ocorre porque VDM-SL não prevê um nível de abstração ideal para representar funcionalidades a serem desempenhadas pelo software, tal como exigido por APF (ver seção 2.2.4). Além disso, VDM-SL não fornece todas as informações necessárias para a contagem.

Além dos limitantes apresentados para VDM-SL, destacam-se a seguir problemas adicionais encontrados em relação à abordagem de Miyawaki, Iijima e Ho (2008): *(i)* classificações incorretas para funções de transação, já que a abordagem não incorpora todas as lógicas de processamento previstas por APF original; *(ii)* menor atribuição de DERs e ALRs quando comparada à APF original, por usar um artefato que não contemple informações que possibilitem quantificar adequadamente esses elementos; *(iii)* funções de transação classificadas, concomitantemente, de modos distintos, já que as decisões subjetivas necessárias para a determinação do tipo de função de transação não são refletidas nesse artefato; e *(iv)* simplificação de APF original (ver quadros 13 e 14 do apêndice B).

4. Modelo conceitual baseado em engenharia de requisitos: uma abordagem propõe a contagem de pontos de função baseado no Diagrama de casos de uso e o Diagrama de sequência, também empregados na abordagem MCbER (ver seção 2.3). O Diagrama de sequência foi estendido com os seguintes estereótipos: «signal», que representa mensagens trocadas entre ator e software; «service», que representa mensagens para atualizar o estado dos objetos; «query», que representa mensagens para consultar o estado do objeto; «select», que representa mensagens para selecionar as instâncias das classes de servidor; «service/new», que representa a criação de uma instância de uma classe alvo; «service/update», que representa a alteração do estado de uma instância de uma classe alvo; «service/destroy», que representa a destruição de um objeto de uma classe alvo; e «connect», que representa interações usadas para estabelecer uma relação estrutural entre objetos e interações. O quadro 12 (no apêndice B) resume o mapeamento proposto entre estereótipos do Diagrama de sequência e conceitos de APF. Esse mapeamento foi implementado em uma ferramenta, denominada REST tool, que a partir da especificação de requisitos elaborada na ferramenta RETO tool, calculava automaticamente a quantidade de pontos de função. O

experimento realizado, que considerou a medição de requisitos funcionais de um único software, apresentou um coeficiente de variação de 6,15% em relação à APF original.

Entretanto, o uso de MCbER não é ideal para a contagem de pontos de função sem a incorporação de informações adicionais. Isso ocorre porque MCbER não contém todas as informações necessárias para a contagem de pontos de função, demandando levantamento adicional de informações por parte do contador.

Além dos limitantes apresentados para MCbER, destacam-se a seguir problemas adicionais encontrados em relação à abordagem de Abrahão e Insfrán (2008): *(i)* consideração de cada mensagem do Diagrama de sequência equivalente a um processo elementar; *(ii)* classificação de funções de dados e de transação a partir dos estereótipos do Diagrama de sequência; e *(iii)* a simplificação de APF original (ver quadros 13 e 14 do apêndice B). Esses problemas ocorrem porque: *(i)* não é ideal considerar que cada mensagem do Diagrama de casos de uso representa um processo elementar, já que a mensagem de um Diagrama de sequência, por definição, não atende aos critérios necessários para ser classificada como um processo elementar (ver seção 3.2); e *(ii)* a abordagem simplifica demasiadamente os conceitos de função de transação ao restringir a classificação das funções de transação aos estereótipos previstos no Diagrama de sequência.

5. **Código fonte:** duas abordagens propõe a contagem de pontos de função baseado em código fonte. Edagawa et al. (2011) propuseram usar códigos fontes de softwares web (escritos em Java apoiado no framework *Struts*) com base em transações de telas e acesso ao banco de dados (via *JDBC* ou *iBATIS*). Essa abordagem prevê inicialmente a extração do código fonte de todas as ações disponíveis em tela. Para cada ação extraída, uma árvore SQL é construída, contendo três tipos de nó: *(i)* *nó de código*, que representa o intervalo de código fonte relacionado a uma determinada ação; *(ii)* *nó de ramo*, que representa as condições contidas no intervalo de código; e *(iii)* *nó de chamada SQL*, que representa a chamada SQL. As informações necessárias são extraídas dessa estrutura para aplicar APF. Nessa abordagem, uma função de transação equivale a diversas instruções SQL executadas simultaneamente e uma função de dados equivale a tabelas lidas ou referenciadas pelo software. A abordagem foi implementada em uma ferramenta, que a partir da análise do código fonte do software, extraia as informações necessárias para calcular automaticamente a quantidade de pontos de função. O experimento realizado, que considerou a medição de requisitos funcionais para sete softwares, apresentou um coeficiente de variação de 5,01% em relação à APF original. Klusener (2007) propõe que a contagem de pontos de função seja realizada via código fonte desenvolvido em Cobol e JCL (Job Control Language). Tal contagem prevê que os arquivos

lógicos sejam identificados a partir de arquivos Cobol, tabelas DB2 e data sets encontrados em jobs JCL e programas Cobol. As funções de transação devem ser identificadas a partir de declarações de E/S, declarações SQL e declarações de chamadas. Essa abordagem não foi aplicada a nenhum projeto, não sendo possível identificar se apresenta resultados próximos aos medidos pelos contadores.

Entretanto, o uso do código fonte de um software não é ideal para a contagem de pontos de função sem a incorporação de informações adicionais. Isso ocorre porque o código fonte detalha as funcionalidades existentes no menor nível de abstração possível, não sendo possível obter informações em maior nível de abstração, necessárias para a contagem de pontos de função. Além disso, as diversas sintaxes existentes para linguagens de programação aumentam a complexidade da definição de um mapeamento único de código fonte (em quaisquer linguagens de programação) para conceitos de APF. Adicionalmente, seu uso impossibilitaria que fossem realizadas medições no início do ciclo de vida do software, levando a uma consequência negativa, já que essas são necessárias principalmente no início do ciclo de vida.

Além dos limitantes apresentados para código fonte, destacam-se a seguir problemas adicionais encontrados em relação à abordagem de Edagawa et al. (2011): *(i)* consideração de que cada tabela do software equivale a uma função de dados (a identificação do significado das tabelas para um usuário de processo de negócio não é possível a partir desse artefato); *(ii)* identificação incorreta de processos elementares, que são identificados a partir de informações em nível de abstração inadequado; *(iii)* classificação incorreta das funções de transação, já que a abordagem falha ao diferenciar SE de CE; *(iv)* aplicação da abordagem limitada a um único contexto tecnológico, ou seja, softwares web desenvolvidos com tecnologias Java, struts e JDBC ou iBATIS; e *(v)* simplificação de APF original (ver quadros 13 e 14 do apêndice B). Em relação à abordagem de Klusener (2003), três problemas foram identificados: *(i)* aplica-se apenas a projetos de manutenção; *(ii)* aplicação da abordagem é limitada a um único contexto tecnológico, ou seja, softwares desenvolvidos com tecnologias COBOL, SQL e JCL Jobs; e *(iii)* simplificação de APF original (ver quadros 13 e 14).

6. **ER-DFD**: uma abordagem propõe a contagem de pontos de função baseado em um novo artefato, denominado ER-DFD (Entidade Relacionamento – Diagrama Fluxo de Dados). ER-DFD integra o Diagrama Entidade-Relacionamento (ER) e o Diagrama de Fluxo de Dados (DFD) em um único artefato de forma simples. Em ER-DFD, transações do software são representadas via DFD, porém, o armazenamento de dados (usualmente também expresso via DFD), é representado via ER. Para permitir o uso dos dois diagramas em conjunto, a

representação gráfica de alguns de seus elementos foi alterada, já que há sobreposição de formas entre eles.

Lamma, Mello e Riguzzi (2004) propuseram um mapeamento entre os componentes de ER-DFD e conceitos de APF. Em ER-DFD, cada processo modelado é classificado como uma função de transação e cada entidade modelada é classificada como uma função de dados, sendo necessário que o responsável por essa tarefa tenha conhecimento das regras de APF para modelar corretamente os processos. A abordagem proposta por Lamma, Mello e Riguzzi (2004) foi implementada em uma ferramenta denominada FUN, que a partir do artefato ER-DFD, extraia as informações necessárias para calcular automaticamente a quantidade de pontos de função. A elaboração do ER-DFD foi realizada a partir da especificação do IFPUG de um software de recursos humanos (também usado para testar a abordagem proposta nesse trabalho – ver seção 6.2). O experimento realizado, que considerou a medição de requisitos funcionais para um único software, apresentou um coeficiente de variação de 15,91% em relação à APF original.

Entretanto, o uso de ER-DFD não é ideal para a contagem de pontos de função sem a incorporação de informações adicionais. Embora a integração de ER e DFD possibilite o uso de um maior conjunto de informações para a contagem de pontos de função, por outro lado isso não elimina problemas em potencial, quando os dois artefatos são observados de modo isolado. Para Jilani et al. (2008), DFD é usualmente gráfico, de modo que praticantes e ferramentas interpretam suas notações de modo distinto, sendo necessários maiores formalismos para seu uso. Assim, apenas a integração desse último com ER não é suficiente para que possíveis ambiguidades inerentes a esse artefato possam ser eliminadas.

Além dos limitantes apresentados para ER-DFD, destacam-se a seguir problemas adicionais encontrados em relação à abordagem de Lamma, Mello e Riguzzi (2004): (i) exigência de que o responsável pela elaboração do ER-DFD tenha conhecimento prévio em APF, já que a abordagem define que as funcionalidades e os atributos das entidades devam ser representados nesse artefato de modo que atendam aos conceitos do CPM de processo elementar, respectivamente; e (ii) simplificação de APF original (ver quadros 13 e 14 do apêndice B).

7. Web Modeling Language (WebML): uma abordagem propõe a contagem de pontos de função baseado em artefatos da técnica WebML. Essa abordagem é empregada no desenvolvimento de softwares web e prevê a geração de código fonte a partir da elaboração de quatro modelos: (i) *modelo de dados*, representado pelo modelo Entidade-Relacionamento, ODMG (*Object Data Management Group*) ou diagramas UML; (ii) *modelo de hipertexto*,

composto por modelo de composição (que contém a especificação das páginas que estão no *website*) e modelo de navegação (que define a navegação de *links* entre as páginas), além de contar com os seguintes elementos principais – “páginas web”, estruturadas em “subpáginas web” e “unidades de conteúdo”, que representam um componente responsável pela publicação de conteúdo na página web; (iii) *modelo de apresentação*, que contém a especificação do posicionamento dos elementos que representam as informações descritas no modelo de dados e a aparência gráfica do software; e (iv) *modelo de personalização*, que contém a especificação de usuários e de grupos de usuários.

Na abordagem de Fraternali, Tisi e Bongio (2006) que contempla o uso de WebML para a contagem de pontos de função, os “modelos de dados” (representados por um Diagrama de classes) são usados para identificar as funções de dados e o “modelo de hipertexto” é usado como especificação dos usuários. Inicialmente, essa abordagem prevê identificar as funções de dados a partir das classes do Diagrama de classes; na sequência, verifica-se os relacionamentos entre as classes ou as funções que acessam essas entidades para determinar se caracterizavam arquivos lógicos e se eram ALIs ou AIEs. As funções de transação são identificadas a partir do modo como os usuários interagem com as páginas web, ou seja, páginas acessadas sequencialmente ou que demandam obrigatoriamente o retorno do usuário para a tela anterior são classificadas como uma única função de transação. A abordagem proposta por Fraternali, Tisi e Bongio (2006) foi implementada em uma ferramenta, que a partir de WebML, extraia as informações necessárias para calcular automaticamente a quantidade de pontos de função. O experimento realizado, que considerou a medição de requisitos funcionais para cinco softwares, apresentou um coeficiente de variação de 4,08% em relação à APF original.

Entretanto, o uso de WebML não é ideal para a contagem de pontos de função sem a incorporação de informações adicionais. O Modelo de hipertexto, por exemplo, armazena apenas dados sobre páginas que compõe um website e como tais páginas e unidades de conteúdo se relacionam para formar um hipertexto, não sendo um bom indicativo dos requisitos funcionais do usuário sobre quais funcionalidades devem ser desempenhadas pelo software.

Além dos limitantes apresentados para WebML, destacam-se a seguir problemas adicionais encontrados em relação à abordagem de Fraternali, Tisi e Bongio (2006): (i) processos elementares são identificados exclusivamente a partir da navegação do usuário no website, levando em consideração apenas a implementação realizada para a interface. Assim, um único requisito funcional implementado com navegação distinta poderia ser pontuado de

modos distintos por essa abordagem; *(ii)* aplicação é limitada a um único contexto tecnológico, ou seja, software web; e *(iii)* simplificação de APF original (ver quadros 13 e 14 do apêndice B).

8. **Modelo conceitual OO-H (Object-oriented Hypermedia)**: uma abordagem propõe a contagem de pontos de função baseado no modelo conceitual OO-H (Object-oriented Hypermedia). Esse modelo é empregado no desenvolvimento de softwares web e combina o Diagrama de classes e o DAN (Diagrama de Acesso Navegacional), que capturam a estrutura e as visões de navegação de software web, respectivamente. DAN possibilita representar como um visitante de um *website* navega e acessa a informação do software e mostrar quais eventos acionam diferentes regras e quais tipos de critérios são satisfeitos para atender aos requisitos (GARRIGOS; CASTELEYN; GÓMEZ, 2005). De acordo com Abrahão et al. (2007), essas visões contêm os elementos que contribuem com o tamanho funcional do software web e, assim, podem ser usados para embasar a contagem de pontos de função.

Na abordagem de Abrahão et al. (2007), Diagramas de classes são usados para classificar o tipo de funções de dados e DAN são usados para classificar o tipo de funções de transação. Por exemplo, classes que compõem hierarquias de agregação e de composição e classes que não fazem parte dessas hierarquias são classificadas como ALIs. Em outro exemplo, cada método do Diagrama de classes é classificado como um EE, enquanto cada alvo de navegação é classificado como CE ou SE. Essa abordagem não foi aplicada a nenhum projeto, não sendo possível identificar se apresenta resultados próximos aos medidos pelos contadores.

Entretanto, o uso do Modelo conceitual OO-H não é ideal para a contagem de pontos de função sem a incorporação de informações adicionais. Isso ocorre porque esse modelo representa apenas a navegação do usuário no software sem refletir inteiramente os requisitos funcionais do usuário.

Além dos limitantes apresentados para Modelo conceitual OO-H, destacam-se a seguir problemas adicionais encontrados em relação à abordagem de Abrahão et al. (2007): *(i)* parte dos processos elementares são identificados a partir da navegação do usuário no website, tal como observado para WebML; *(ii)* a aplicação da abordagem é limitada a um único contexto tecnológico, ou seja, softwares web; *(iii)* há maior atribuição de DERs para as funções de transação quando comparadas à APF original, já que campos técnicos ou que não cruzam a fronteira dos softwares são indevidamente considerados como DERs; *(iv)* simplificação de APF original (ver quadros 13 e 14 do apêndice B).

3.3 Classificação das melhorias para diminuir a subjetividade de APF

Uma classificação entre as abordagens está apresentada na tabela 7. Dois itens dos procedimentos de contagem de pontos de função foram avaliados para cada abordagem: (i) Medir funções de dados e (ii) Medir funções de transação (ver seção 2.2). Para cada item, foi avaliado se a abordagem contemplava todas as regras previstas no CPM. Se a abordagem previa algum dispositivo para atender a regra analisada, a regra era assinalada como “sim”; caso contrário, era assinalada como “não”. Caso a abordagem atendesse parcialmente a regra, era assinalado “parcialmente”. A avaliação de cada abordagem não levou em consideração se o dispositivo previsto para atender a uma determinada regra era suficiente para a identificação correta dos elementos, tal como em APF original. Desse modo, foi realizada apenas uma análise quantitativa das regras cobertas pelas abordagens.

A pontuação final para cada abordagem foi determinada baseada na soma dos pontos de: “sim” (1,0 ponto cada); “não” (0 pontos cada); e “parcialmente” (0,5 ponto cada) – com exceção dos passos 1, 2, 4, 6 e 7 (dos quadros 13 e 14 do apêndice B), que não foram somados para evitar dupla valoração.

Tabela 7 – Pontuação obtida pelas abordagens em relação à cobertura de APF⁸

	A	B	C	D	E	F	G	
Funções de transação	2, 0	2, 5	5, 5	2, 0	3, 0	5, 5	0, 0	
Funções de dados	1, 5	4, 0	6, 5	1, 0	4, 0	6, 5	5, 5	
Pontuação geral	3, 5	6, 5	12, 0	3, 0	7, 0	12, 0	5, 5	
Percentual atendido	16%	30%	55%	14%	32%	55%	42%	
	H	I	J	K	L	M	N	O
Funções de transação	3, 0	3, 0	7, 0	5, 0	2, 0	1, 0	3, 0	4, 0
Funções de dados	1, 0	1, 0	1, 0	4, 0	1, 0	0, 0	8, 5	5, 0
Pontuação geral	4, 0	4, 0	8, 0	9, 0	3, 0	1, 0	11, 5	9, 0
Percentual atendido	18%	18%	36%	41%	14%	5%	52%	41%

Fonte: Marcos Freitas Junior, 2015

⁸ A legenda das siglas designadas para as abordagens propostas pode ser consultada nos quadros 13 e 14 do apêndice B.

A abordagem C, proposta por Harput, Kaindl e Kramer (2005), e a abordagem F, proposta por Lavazza, Bianco e Caravaglia (2008), apresentaram os melhores resultados comparativamente, com uma cobertura de 55% do total de passos de APF considerados na análise. Na sequência e com uma cobertura próxima, a abordagem N, proposta por Abrahão et al. (2007), com 52%. As demais abordagens apresentaram coberturas que variam de 5% a 42%. No geral, as 15 abordagens apresentaram uma cobertura média de 31%.

Esses números evidenciam que, em geral, as propostas de melhoria identificadas apresentam muitas simplificações em relação às regras originais de APF. Se, por um lado, elas podem contribuir, dentro de um contexto tecnológico específico (para diminuir as variações de contagens entre diferentes contadores) por outro lado, elas alcançam esse benefício usando a estratégia nociva de simplificar as regras de APF. Desse modo, tais abordagens não mantêm a validade com as regras originais de APF e aumentam a possibilidade de variação do tamanho calculado por cada abordagem em relação as demais. Assim, as simplificações propostas por essas abordagens acabam desfavorecendo suas adoções por organizações que usam APF original já que não há garantias de que seus resultados sejam válidos, embora possam ser confiáveis. Consequentemente, os resultados produzidos por essas abordagens, para essas organizações, até seriam confiáveis entre seus diferentes contadores, produzindo resultados padronizados, mas não seriam válidos para comparação com outras organizações que usam APF original.

Onze artefatos distintos foram usados pelas abordagens propostas e, para todos os casos, houve simplificação das regras de APF. As simplificações dos procedimentos para a contagem de pontos de função, contrastados com os distintos artefatos usados, indicam que a falta de validade das abordagens propostas (quando comparadas a APF original) pode ser ocasionada pela falta de informações (necessárias para a contagem de pontos de função) nos artefatos do software usados. A abordagem F proposta por Lavazza, Bianco, Caravaglia (2008), por exemplo, previu a incorporação de novas informações nos artefatos de UML e apresentou os melhores resultados, embora também não tenha incorporado todos os procedimentos necessários para a contagem o que pode indicar que outras informações se faziam necessárias para que todos os procedimentos fossem incorporados.

Em relação à simplificação das regras de APF, por exemplo, nenhuma das abordagens prevê um passo para excluir entidades classificadas como “dados de código” ao identificar funções de dados, por representarem detalhes de implementação, sendo esse um dos passos integrantes das regras de APF. Segundo IFPUG (2010), dados de código costumam representar 50% do total de entidades de um modelo de dados na 3ª Forma Normal. A não

exclusão desses pode inflar o tamanho das funções de dados e, conseqüentemente, de outras medidas derivadas. Os quadros 13 e 14 (no apêndice B) apresentam, em maiores detalhes, as regras de APF não incorporadas pelas abordagens propostas.

3.4 Considerações finais

O objetivo deste capítulo era apresentar relatos na literatura que indicassem os problemas no uso de APF, sobretudo relacionados à validade das medições; as melhorias propostas para mitigar esses problemas e as principais limitações das abordagens encontradas.

Para isso, realizou-se uma RSL (conforme detalhado no capítulo 4) na qual foram identificadas 15 abordagens analisadas neste capítulo. O alto número de abordagens propostas para incrementar a confiabilidade dos resultados de medição gerados indica que melhorias nesse quesito se fazem necessárias. Foi possível verificar que as abordagens propostas simplificam demasiadamente as regras estabelecidas no CPM 4.3.1 para determinar o tamanho funcional do software. Como consequência disso, o uso de uma abordagem em determinada organização geraria um resultado distinto de outra abordagem, já que ambas não englobam a totalidade de procedimentos previstos no CPM 4.3.1 para a contagem de pontos de função.

Portanto, conclui-se que as abordagens resolvem parcialmente o problema, já que esse não é eliminado, quando observado a partir de outra escala. Como exemplo, pode-se comparar analogamente cada uma dessas abordagens com um contador: ele possui certo conjunto de procedimentos para executar uma contagem de pontos de função, mas os procedimentos adotados por ele não são os mesmos que os adotados por outro contador, devido a interpretações pessoais distintas ou desconhecimento da medida de APF, provocando variação nos resultados de medição gerados. O mesmo efeito é observado nas abordagens: elas definem um conjunto de procedimentos (objetivos e passíveis de automatização) para a contagem de pontos de função, porém, ao adotarem apenas parcialmente as regras previstas para a contagem, geram resultados distintos entre si e, principalmente, diferente do tamanho real do software, quando calculado por meio da aplicação de todas as regras previstas no CPM.

Assim, conclui-se que não há uma abordagem que aplique efetivamente todas as regras de APF e, conseqüentemente, elimine o problema de modo satisfatório. No próximo capítulo, informações sobre o método de pesquisa aplicado neste trabalho são apresentadas.

4. Método de pesquisa

Este trabalho de pesquisa se caracteriza como uma pesquisa do tipo “ciência de projeto” (*design science*) cujo propósito é solucionar necessidades organizacionais, tendo o respaldo científico para embasar a proposição de soluções para problemas identificados (HEVNER et al., 2004). Esse tipo de pesquisa tem como objetivo resolver problemas práticos das organizações, estabelecendo uma ponte entre a ciência e ação prática. Busca-se solucionar problemas identificados pela própria pesquisa, ao invés de realizar apenas uma análise descritiva extensa do problema que não leva necessariamente à proposição de melhorias para ele, por exemplo (HEVNER et al., 2004). Assim, ciência de projeto mostra-se adequada a este trabalho, já que seu objetivo é solucionar o problema da falta de padronização e sistematização na aplicação de APF, descrito por diferentes pesquisadores conforme apresentado no capítulo 3, e que tem impacto direto na acurácia dos resultados calculados por contadores.

Esse projeto de pesquisa foi realizado via as seguintes atividades: (i) estudo exploratório dos fundamentos da área; (ii) estudo sistemático do estado da arte; (iii) definição dos objetivos e requisitos buscados; (iv) projeto e desenvolvimento da abordagem proposta; (v) prototipação de ferramenta de apoio; e (vi) experimentação e avaliação de resultados obtidos.

Primeiramente, um estudo exploratório considerando o contexto-base deste trabalho foi realizado por meio de revisão de livros e artigos científicos, e de reuniões com o professor orientador. A fundamentação teórica deste estudo compreendeu: conceitos básicos de medidas e medição; regras que permeiam a medida APF; e Modelagem conceitual baseada em engenharia de requisitos, incluindo a técnica Árvore de refinamento de funções. Os resultados desse estudo exploratório serviram para embasar a obtenção do conteúdo necessário para a realização deste trabalho, conforme apresentado no capítulo 2.

Na sequência, de forma mais aprofundada, uma revisão sistemática da literatura foi realizada para investigar trabalhos relacionados aos objetivos deste trabalho. Com isso, foi possível compreender as principais críticas realizadas a APF e as principais abordagens propostas para sua melhoria. Além disso, também foi possível identificar lacunas ainda existentes no contexto avaliado. Informações mais completas sobre a revisão sistemática realizada estão apresentadas a seguir, na seção 4.1. Os resultados desse estudo sistemático também serviram para embasar a obtenção do conteúdo necessário a este trabalho, conforme apresentado no capítulo 3, além de terem sido usados na elaboração de um artigo científico

publicado no periódico internacional *IEEE Transactions of Engineering Management*, com mais informações apresentadas no apêndice C deste documento.

Com base na análise dos trabalhos relacionados e, principalmente, nas lacunas identificadas para o problema em questão, houve o refinamento dos objetivos buscados e dos requisitos a serem tratados com este trabalho, e na sequência iniciou-se o projeto e o desenvolvimento da abordagem propriamente dita, apresentada no capítulo 5 deste documento. Em suma, as lacunas encontradas indicaram a ausência de abordagens efetivas para melhorar a confiabilidade das medições de pontos de função ao mesmo tempo em que mantém sua validade, já que, em geral, elas falham em manter a completude das regras de APF original. As lacunas encontradas indicaram também a ausência de propostas de melhoria de confiabilidade das medições que sejam acompanhadas de melhoria de eficiência por meio de automação.

Também foi parte do desenvolvimento da abordagem proposta sua automação por meio de um protótipo de ferramenta que visasse demonstrar sua aplicabilidade computacional. Por fim, a abordagem proposta foi experimentada e avaliada, considerando duas ações principais: (a) comparação sistemática entre a abordagem proposta e os trabalhos relacionados considerando o mecanismo de comparação apresentado na seção 3.3 (ver quadros 13 e 14 e tabelas 7 e 8 do apêndice B); (b) experimento em que um grupo de analistas de requisitos ou analistas de sistema realizou a contagem manual de pontos de função usando a abordagem proposta e outro grupo de analistas realizou a contagem, via protótipo de ferramenta desenvolvida, tendo seus resultados comparados entre si para avaliar a eficiência, a validade e a confiabilidade da abordagem proposta neste trabalho.

A seção a seguir apresenta mais informações a respeito da revisão sistemática realizada como parte do método de pesquisa deste trabalho.

4.1 Protocolo da revisão sistemática

Revisão Sistemática da Literatura (RSL) é usada para identificar, avaliar e interpretar toda a pesquisa disponível relevante para uma questão de pesquisa específica, uma área temática, ou um determinado fenômeno de interesse (KITCHENHAM, 2004). RSL vai um passo além de uma simples revisão bibliográfica, já que ela visa integrar a pesquisa empírica para criar generalizações. Isso permite realizar uma análise crítica dos dados coletados, para resolver problemas detectados na literatura e identificar problemas que possam ser estudados

futuramente (BIOLCHINI; MIAN; NATALI, 2005). A seguir, são apresentadas as informações mais relevantes a respeito do protocolo e da condução da RSL realizada.

Com a RSL, buscou-se responder às seguintes questões de pesquisa: (1) Existem problemas sendo relatados para APF? (2) Quais melhorias estão sendo propostas para APF em termos de confiabilidade? (3) As melhorias propostas se mostraram efetivas? (4) Quais limitações existem para as melhorias propostas?

As seguintes bases de dados foram usadas para a busca: (i) Scopus; (ii) IEEEExplore; (iii) ACM Digital Library; (iv) Science Direct. A *string* de busca genérica criada para a busca dos estudos primários foi: “function point” OR “function points” OR “function-point” OR “function-points”. Essa *string* precisou ser adaptada para cada uma das quatro bases de dados.

Os critérios de inclusão visaram incluir artigos: (i) com foco em APF cujo principal objetivo seja identificar alguma limitação ou deficiência nessa medida em termos de acurácia; (ii) com foco em APF cujo principal objetivo seja propor melhorias nos procedimentos existentes nessa medida para solucionar os problemas de acurácia identificados; (iii) publicados e disponíveis integralmente *online* em bases de dados científicas; (iv) publicados a partir de 1 de janeiro de 2002, já que a última versão do Manual de Práticas de Contagem (CPM), na versão 4.X, foi publicada pelo IFPUG em 2002; (v) publicados em periódico ou em anais de evento, com revisão por pares; (vi) publicados inteiramente no idioma inglês. Os critérios de exclusão visaram excluir artigos que: (i) propõem novas medidas para a medição funcional de software alternativa a APF; (ii) apresentam um estudo secundário; (iii) investigam a relação de APF (ou compará-la) com outras medidas funcionais; (iv) investigam a conversibilidade do tamanho funcional medido com APF com o tamanho medido por outras medidas funcionais; e (v) realizam melhorias nos passos de APF para aumentar a rapidez de cálculo ou quaisquer outros fatores sem que esses tornem a aplicação de APF mais acurada.

Como resultado final, 15 estudos primários foram selecionados, sendo três publicados em periódicos e 12 em anais de conferência. O protocolo completo de revisão sistemática para a condução da RSL está apresentado no apêndice D deste documento.

5. Análise de pontos de função baseada em APF-APF

Visando resolver o problema relacionado à interpretação individual na aplicação de APF (ver seção 3.1), é proposta uma abordagem, denominada Análise de Pontos de Função baseada em Árvore de Pontos de Função (APF-APF), projetada para ser usada como apoio sistemático à coleta das informações necessárias à aplicação de APF original, sem que nenhum de seus passos seja excluído ou simplificado. Para isso, propõe-se que a coleta das informações seja realizada em um formato pré-determinado, denominado Árvore de pontos de função, para incorporar todas as informações necessárias à aplicação de APF. Feito isso, uma ferramenta pode calcular automaticamente o número de pontos de função

A Árvore de pontos de função é uma extensão da técnica de Árvore de refinamento de funções usada na abordagem MCbER (ver seção 2.3). MCbER foi escolhida para apoiar APF-APF por estabelecer níveis de abstração para os requisitos funcionais do usuário próximos aos requeridos para a contagem de pontos de função. Diferentes níveis de abstração para os requisitos funcionais do usuário possibilitam que tais requisitos sejam especificados desde o início do Processo de engenharia de requisitos de modo mais semelhante com o requerido para a contagem. Assim, a contagem de pontos de função é realizada com menor grau de interpretação individual por parte dos contadores, já que esses recebem uma especificação de requisitos mais adequada para a contagem, podendo realizar medições com maior grau de confiabilidade e validade quando comparadas à APF original.

Além de MCbER, outra abordagem que estabelece diferentes níveis de abstração para requisitos é ERbCO, conforme apresentado na seção 3.3. Entretanto, MCbER foi escolhida para apoiar APF-APF por possuir um nível de abstração para os requisitos mais próximo do necessário para identificar os processos elementares do software.

MCbER já foi usada para apoiar a contagem de pontos de função na abordagem proposta por Abrahão e Insfrán (2008). Entretanto, existem diferenças significativas entre a abordagem desses autores e APF-APF no que diz respeito a: *(i)* quais técnicas componentes de MCbER foram selecionadas para apoiar a contagem de pontos de função; *(ii)* existência ou não de adaptações nas técnicas selecionadas; e *(iii)* mapeamentos realizados de MCbER para APF. Enquanto Abrahão e Insfrán (2008) usam Diagramas de sequência e Modelos de caso de uso, sem qualquer adaptação ou extensão, a abordagem APF-APF usa a técnica Árvore de refinamento de funções adaptada, cuja extensão levou à criação de uma nova técnica – a Árvore de pontos de função. As adaptações se fizeram necessárias, já que, após análise exploratória inicial, verificou-se que, embora a abordagem MCbER seja em geral a mais

adequada para apoiar a contagem de pontos de função, suas técnicas componentes não continham todas as informações necessárias para oferecer o apoio de forma completa. Assim, alterações nas técnicas componentes de MCbER foram necessárias. Além disso, os mapeamentos realizados por Abraão e Insfrán (2008) – conforme destacado na seção 3.3 – apresentam inconsistências que culminam em um resultado final incorreto. Em APF-APF, o estabelecimento de novos mapeamentos permitiu a correção desse problema.

Este capítulo apresenta as seguintes informações relacionadas à APF-APF: (i) visão geral da abordagem; (ii) adaptações propostas à Árvore de refinamento de funções; e (iii) mapeamento de elementos da Árvore de pontos de função para conceitos de APF.

5.1 Visão geral da abordagem APF-APF

A abordagem APF-APF propõe que os requisitos funcionais do usuário sejam especificados – obrigatoriamente – em formato de Árvore de pontos de função. A partir dessa especificação, a contagem de pontos de função pode ser feita de forma manual ou automática (com o apoio de uma ferramenta) dispensando em ambos os casos o papel de contador. Também é proposto que a Especificação dos requisitos funcionais do usuário, por meio da Árvore de pontos de função, seja incorporada ao processo tradicional de especificação de requisitos sendo, portanto, realizada por um analista de requisitos.

Por representar os requisitos funcionais do usuário, a Árvore de pontos de função pode ser utilizada para contagem de pontos de função de qualquer tipo de software independente de sua tecnologia, arquitetura e/ou qualquer outro aspecto não relacionado aos requisitos funcionais do usuário. Também não existem restrições para a utilização de APF-APF para softwares, com maior ou menor, complexidade.

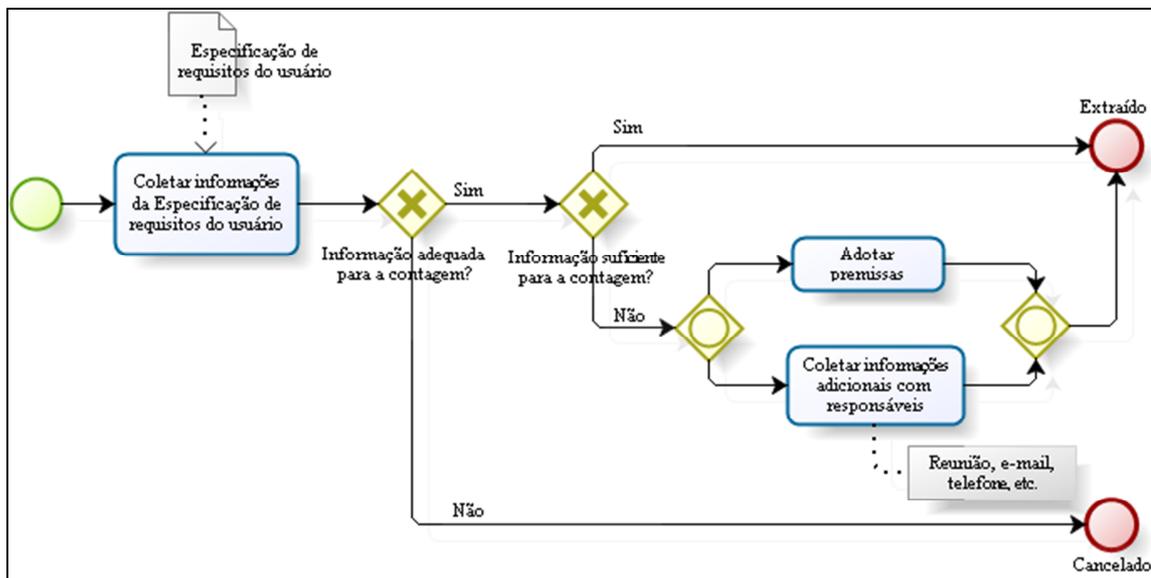
O processo padrão para a contagem de pontos de função (ver fluxograma 1) se inicia com a disponibilização da Especificação de requisitos do usuário (não limitada somente a esse artefato), elaborada pelo Analista de requisitos como resultado do Processo de engenharia de requisitos⁹, para o qual nenhum formato específico é requerido. Este, é usado pelo contador para Extrair as informações necessárias para a contagem de modo que todas as Informações coletadas são usadas para que se possa paralelamente Realizar a contagem de pontos de

⁹ O detalhamento do subprocesso “Processo de engenharia de requisitos” está fora do escopo deste trabalho.

de requisitos do usuário não é suficiente, fazendo com que o contador opte por: (i) Adotar premissas que darão subsídio a Realizar a contagem de pontos de função, futuramente ou (ii) Coletar informações adicionais com responsáveis, de modo que todas as informações necessárias estejam disponíveis. Como essa coleta pode ser realizada por diversos meios, como reunião, e-mail, telefone etc., as Informações coletadas podem variar de acordo com o tipo de levantamento realizado. Nesses casos, o resultado gerado pela medição também pode variar, comprometendo diretamente sua confiabilidade.

Diante desse contexto, a abordagem APF-APF propõe a incorporação da macroatividade Elaborar Árvore de pontos de função como um passo adicional ao processo padrão de contagem de pontos de função (conforme etapas apresentadas nos fluxogramas 1 e 2), a ser desempenhado também pelo papel Analista de requisitos. Com a incorporação desse subprocesso, elimina-se a necessidade do papel contador Extrair as informações necessárias à contagem, já que todas as informações estarão concentradas necessariamente em um único artefato – a Árvore de pontos de função. Como consequência, as atividades anteriormente existentes para o papel contador são eliminadas, enquanto as atividades atribuídas ao papel analista de requisitos são ampliadas.

Fluxograma 2 – Subprocesso Extrair as informações necessárias



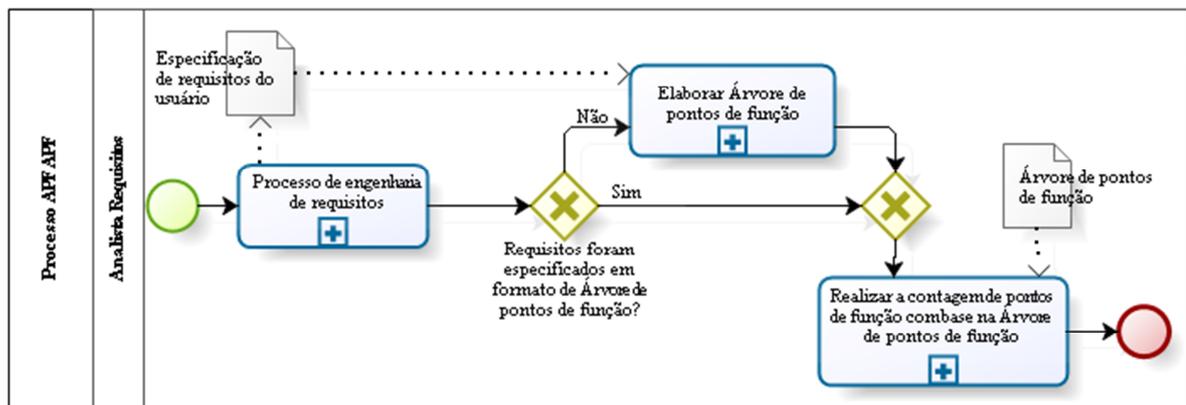
Fonte: Marcos Freitas Junior, 2015

O processo APF-APF (ver fluxograma 3) inicia-se com a elaboração da Especificação de requisitos do usuário, realizada pelo Analista de requisitos, como resultado do Processo de

engenharia de requisitos, e para a qual nenhum formato específico é requerido. Se os requisitos foram elaborados em formato de Árvore de pontos de função, então ela já pode ser usada diretamente para Realizar a contagem de pontos de função; caso contrário, o Analista de requisitos deve Elaborar Árvore de pontos de função para se prosseguir com a contagem.

Em ambos os casos – abordagem padrão e APF-APF –, os pesos dos papéis de Analista de requisitos e contador são invertidos: enquanto o contador é eliminado, o Analista de requisitos passa a ter mais atividades em APF-APF devido à inclusão de um novo artefato obrigatório no processo de especificação: a Árvore de pontos de função. Como consequência, a importância do Analista de requisitos dentro do processo de medição é maximizada.

Fluxograma 3 – Contagem de pontos de função baseada em Árvore de pontos de função



Fonte: Marcos Freitas Junior, 2015

APF-APF foi projetada visando às seguintes vantagens em relação ao processo padrão para a contagem de pontos de função: (i) oferecer uma técnica para coletar todas as informações necessárias para a contagem de pontos de função, possibilitando que as informações necessárias sejam documentadas desde o Processo de engenharia de requisitos, eliminando possíveis interpretações por parte dos contadores; (ii) documentar as informações necessárias para a contagem de pontos de função de modo centralizado em um único artefato (a Árvore de pontos de função), ao contrário do processo padrão, no qual premissas ou levantamentos adicionais normalmente são documentados de modo descentralizado em e-mails, planilhas de contagem, entre outros; e (iii) documentar as informações em momento apropriado no ciclo de vida do software, quando se está definindo e analisando os requisitos.

5.2 Árvore de pontos de função

Esta seção descreve a técnica Árvore de pontos de função, proposta neste trabalho como uma extensão da técnica Árvore de refinamento de funções e usada para embasar a contagem de pontos de função. Primeiramente, é apresentada uma visão geral das adaptações propostas à técnica original em relação a sua estrutura. Depois, são apresentadas mais informações das adaptações em relação aos novos elementos propostos.

5.2.1 Visão geral da estrutura da Árvore de pontos de função

Conforme proposto em MCbER, a Árvore de refinamento de funções padrão não contém todas as informações necessárias para a contagem de pontos de função. Por isso, informações e regras adicionais para modelagem de requisitos de software foram definidas especificamente para esse fim, originando a Árvore de pontos de função. Essas regras são válidas tanto para restringir algumas ações possíveis na Árvore de refinamento de funções, de modo a simplificá-la, quanto para possibilitar novas ações, dependendo da necessidade.

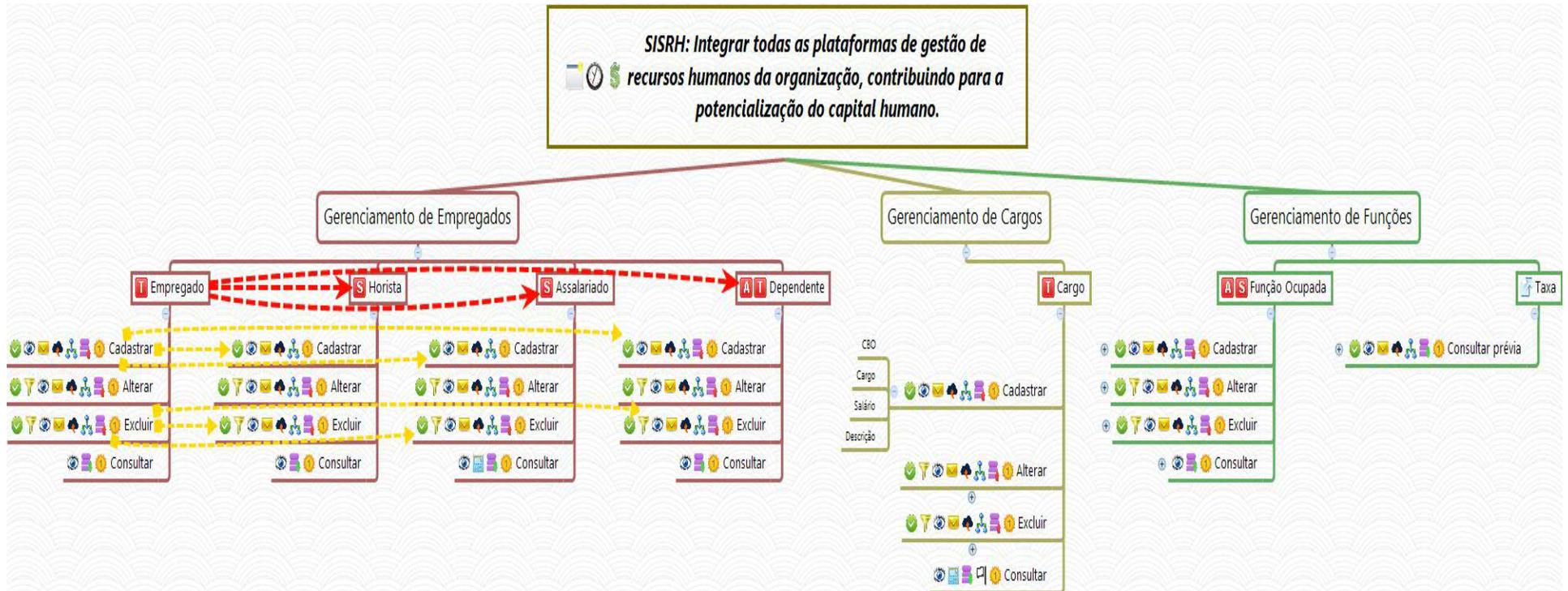
Um exemplo ilustrativo de Árvore de pontos de função é apresentado na figura 7. Com base nesse exemplo, as regras definidas são apresentadas a seguir e descritas nas seções subsequentes. A apresentação dessas regras está relacionada à estrutura de níveis do artefato a ser gerado, comparando com as regras da Árvore de refinamento de funções da seção 2.3:

1. **Nível 0 – Raiz:** toda raiz deve ser composta pelo nome do software e pela descrição da missão do software, separados por dois pontos (<NOME>: <Missão do software>), visando a padronização das informações textuais representadas. Adicionalmente, para cada raiz, o seguinte elemento deve ser considerado:
 - a. **Marcador de raiz:** a raiz da árvore deve possuir ao menos dois marcadores de raiz para possibilitar identificar o propósito e o tipo da contagem. Cada marcador de raiz só pode ocorrer uma única vez na raiz. Exemplo: ícone  na raiz da árvore.
2. **Nível 1 – Nós intermediários:** sem alteração em relação à árvore original.
3. **Nível 2 – Nós folha:** os seguintes elementos devem ser considerados:
 - a. **Marcador de nó:** os nós folha devem possuir um ou mais marcadores de nó para possibilitar identificar as lógicas de processamento executadas por cada

função elementar, e cada marcador de nó só pode ocorrer uma única vez em um mesmo nó. Exemplo: ícone  no nó folha “Consultar Empregado”.

- b. **Conector de inclusão:** se aplicável, os nós folha podem possuir um ou mais conectores de inclusão que os relacionem a outros nós folha para representar a interdependência entre as funções elementares modeladas. Exemplo: ícone  conectando o nó folha “Cadastrar Empregado” com “Cadastrar Horista”, “Cadastrar Assalariado” e “Cadastrar Dependente”.
 - c. **Atributo de nó:** os nós folha devem possuir um ou mais atributos de nó que demonstram quais campos são exibidos ao usuário ou informados por eles, durante o processamento da função elementar. Exemplo: atributos CBO, Cargo, Salário e Descrição relacionados ao nó folha “Cadastrar Cargo”.
4. **Relacionamentos entre níveis:** as seguintes regras devem ser consideradas:
- a. **Rótulo de entidade de dados:** os relacionamentos entre nós intermediários e nós folha devem possuir, quando aplicável, rótulo(s) associado(s) que descreva(m) qual entidade de dados do software é mantida ou referenciada pelas funções elementares relacionadas. Exemplo: rótulo “Cargo” para o relacionamento entre o nó intermediário “Gerenciamento de Cargos” e o nó folha “Cadastrar”.
 - b. **Marcador de rótulo de entidade:** os rótulos de entidade devem possuir um ou mais marcadores de rótulo de entidade para possibilitar identificar os RLRs dos Arquivos Lógicos e auxiliar na classificação das funções de dados, e marcador de rótulo de entidade só pode ocorrer uma vez em um mesmo rótulo de entidade. Exemplo: ícone  no rótulo de entidade de dados “Taxa”.
 - c. **Conector de exclusão:** se aplicável, os rótulos de entidade de dados podem possuir um ou mais conectores de exclusão que os relacionem a outro rótulo de entidade de dados para representar a dependência entre os rótulos de entidade modelados. Exemplo: ícone  conectando o rótulo de entidade de dados “Empregado” com “Horista”, “Assalariado” e “Dependente”.
 - d. **Atributo de rótulo de entidade:** os rótulos de entidade devem possuir dois ou mais atributos de rótulo de entidade que representam as informações específicas armazenadas na entidade de dados do software que são mantidas ou referenciadas pelos nós folha relacionados. Exemplo: o rótulo de entidade de dados “Cargo” deve armazenar os dados: CBO, Cargo, Salário e Descrição mantidos pela função elementar “Cadastrar Cargo”.

Figura 7 – Exemplo da Árvore de pontos de função



Fonte: Marcos Freitas Junior, 2015

5.2.2 Raiz

Esta seção apresenta a raiz da Árvore de pontos de função. A raiz é uma representação gráfica que indica o software modelado e sua missão dentro da organização. A raiz é o primeiro elemento a ser inserido na árvore e deve ter a ela associada os Marcadores de raiz, obrigatoriamente. O elemento marcador de raiz é descrito a seguir.

5.2.2.1 Marcador de raiz

Esta seção apresenta os diferentes tipos de marcador de raiz a serem associados à raiz da Árvore de pontos de função. Trata-se de informações que indicam: (i) o objetivo para realizar a contagem de pontos de função em uma organização; e (ii) as características do software modelado no momento da contagem. A raiz sempre deve ter a ela associada ao menos dois marcadores para representar cada tipo de informação.

Quadro 1 – Descrição dos marcadores de raiz usados para representar o objetivo da contagem

Nome	Ícone	Definição	Exemplo
ESFO		Estimar o esforço requerido para o desenvolvimento do software.	Usar o tamanho funcional de um projeto como uma entrada para o processo de estimativa a fim de determinar o esforço para desenvolver a primeira versão de um software.
CUST		Estimar o custo requerido para o desenvolvimento do software.	Usar o tamanho funcional de um projeto como uma entrada para o processo de estimativa a fim de determinar o custo para desenvolver a primeira versão de um software.
PROD		Estimar a produtividade requerida para o desenvolvimento do software.	Usar o tamanho funcional de um projeto como uma entrada para o processo de estimativa a fim de determinar a produtividade para desenvolver a primeira versão de um software.
PARA		Comparar softwares distintos para apoiar o processo de decisão.	Usar o tamanho funcional calculado para dois pacotes de softwares fechados distintos, para permitir a comparação das funcionalidades oferecidas por cada um para balizar a escolha de qual será implantando na organização.

Há sete marcadores de raiz possíveis: (i) os primeiros quatro marcadores de raiz, exibidos no quadro 1, estão relacionados ao objetivo para realizar a contagem de pontos de função em uma organização, conforme definido em APF (ver seção 2.2.2); e (ii) os últimos três marcadores de raiz, exibidos no quadro 2, estão relacionados às características do software modelado no momento da contagem. Para o primeiro caso, um ou mais dos quatro marcadores podem ser usados, mas para o segundo caso apenas um dos três marcadores pode ser usado.

Quadro 2 – Descrição dos marcadores de raiz usados para representar características do software

Nome	Ícone	Definição	Exemplo
NOVO		Representar um novo software.	Deseja-se obter o tamanho funcional para um novo software de gestão de recursos humanos que deve ser desenvolvido em uma organização.
EVOL		Representar um software sendo evoluído .	Deseja-se obter o tamanho funcional dos novos requisitos funcionais do usuário para um software de gestão de recursos humanos existente que deve ser evoluído.
EXIS		Representar um software existente que não está sofrendo alterações.	Deseja-se obter o tamanho funcional total do software após sua implantação ou evolução(ões).

Fonte: Marcos Freitas Junior, 2015

5.2.3 Nós folha

Esta seção apresenta os nós folha, uma representação gráfica que indica as funções elementares do software. Para cada nó folha, é necessário informar os marcadores de nó, os conectores de inclusão (quando aplicável) e os atributos de nó, obrigatoriamente. Esses três elementos são descritos nas próximas seções.

5.2.3.1 Marcador de nó

Esta seção apresenta os marcadores de nó a serem associados aos nós folha. Eles indicam a presença de características relacionadas ao comportamento das funções elementares durante sua execução. Sempre que uma função elementar do software contiver uma ou mais

das características representadas por marcadores de nó, esses devem ser obrigatoriamente incorporados ao respectivo nó folha. Há 15 marcadores de nó possíveis, conforme apresentados nos quadros 3 e 4.

Quadro 3 – Descrição dos marcadores de nó usados para representar características relacionadas ao comportamento das funções elementares equivalentes as lógicas de processamento de APF

Nome	Ícone	Definição	Exemplo	LP APF
VALI		Realizar validações de dados solicitadas por regras de negócio.	Validar se um CPF é válido de acordo com as regras de validação emitidas pela Receita Federal.	01
CALC		Realizar ao menos um cálculo que atenda a uma regra de negócio.	Cálculo de juros pelo atraso de pagamento de boleto.	02
CONV		Realizar conversão de formato para um atributo, mantendo a mesma informação em formato equivalente.	Conversão do atributo “temperatura” da unidade Graus Celsius para Kelvin.	03
FILT		Realizar filtros de acordo com critérios específicos sobre um conjunto de dados.	Listagem dos empregados da organização que tenham “dois filhos e mais de 40 anos de idade”.	04
COND		Depender de condições que, quando analisadas, podem levar a resultados diferentes.	Exibição de diferentes atributos em tela dependendo se o empregado é horista ou assalariado.	05
MANT		Manter dados de uma ou mais entidades.	Cadastrar um novo empregado na organização.	06
RECP		Recuperar ou referenciar dados de uma ou mais entidades.	Recuperar as informações da taxa de conversão do dólar para determinar o valor de venda um item.	07, 08
DERV		Derivar (gerar) dados que caracterizem uma nova informação em função de outros dados já existentes.	Geração de número de matrícula combinando parte do nome do aluno, ano corrente e dia de matrícula.	09
COMP		Possuir atributos que, se alterados, alteram o comportamento do processo de negócio da organização.	Alteração de data do pagamento dos funcionários provoca alterações em processos de negócio da	10

APRE		Apresentar informações ao usuário.	Exibição da lista de empregados de uma organização.	11
RECB		Capacidade de receber informações advindas dos usuários.	Inserção de um conjunto de dados para cadastrar um novo empregado na organização.	12

Fonte: Marcos Freitas Junior, 2015

Quadro 4 – Descrição dos marcadores de nó usados para representar características adicionais relacionadas ao comportamento das funções elementares sem equivalência com as lógicas de processamento de APF

Nome	Ícone	Definição	Exemplo
LOTE		Ser executada em lote (<i>batch</i>), previamente agendada pelo usuário.	Geração de relatório sobre lucros e perdas da empresa no ano.
MENS		Exibir mensagens aos usuários.	Exibição da mensagem “Operação não permitida” caso o usuário tente incluir um empregado já existente.
CONS		Cadastrar ou exibir dados estáticos, constantes ou de domínio.	Exibição da lista com estados de um país.
PRIM		Ter a intenção primária de manter dados de uma ou mais entidades; possuir atributos que, se alterados, alteram o comportamento do processo de negócio da organização ou apresentar informações ao usuário.	Exibição da lista de empregados de uma organização, cuja intenção primária é apresentar informações ao usuário.

Fonte: Marcos Freitas Junior, 2015

O último marcador de nó representa qual marcador (dentre as opções MANT, COMP e APRE) representa o principal objetivo das funções elementares. Assim, ele é obrigatório para um determinado nó folha sempre que pelo menos dois dos três marcadores MANT, COMP ou APRE tiverem sido selecionados para esse nó folha; caso contrário, ele não é aplicável. Por exemplo (ver figura 7), a função Cadastrar Empregado contém os marcadores VALI, APRE, MENS, RECP, COND, MANT e PRIM, de modo que o marcador PRIM foi incluído porque haviam sido incluídos os marcadores APRE e MANT. Assim, o analista de requisitos teve que informar se o principal objetivo dessa função elementar era: (i) Manter dados de uma ou mais

entidades (MANT), ou (ii) Apresentar informações ao usuário (APRE). Especificamente nesse exemplo, o valor de PRIM é MANT.

5.2.3.2 Conector de inclusão

Esta seção apresenta o conector de inclusão para os nós folha. Trata-se de uma representação gráfica que evidencia a existência de possíveis relacionamentos, e seu tipo, entre duas funções elementares.

O quadro 5 apresenta as definições de quando um conector de inclusão deve ser usado na Árvore de pontos de função. Sempre que o relacionamento entre funções elementares se encaixar nas definições apresentadas no quadro 5, então o uso do conector de inclusão é obrigatório. Um mesmo nó folha pode estar relacionado a diferentes nós folha, por meio de diferentes conectores de inclusão.

Quadro 5 – Descrição do conector de inclusão

Representação	
Descrição	Deve ser usado quando há uma relação entre duas funções elementares em que: a função elementar posterior (depois da seta), quando executada sozinha, não atende aos requisitos funcionais do usuário se a função elementar anterior (contrária à seta) não for executada primeiro. Deve ser usada apenas para conectar nós folha entre si.
Exemplo	Considere o requisito funcional um do SISRH (ver RF1 do apêndice A). Se o requisito funcional do usuário exige que, ao cadastrar um empregado, também sejam cadastrados dados de empregados assalariados ou horistas e dependentes; então cadastrar dependente, assalariado ou horista sem que se tenha cadastrado um empregado não atende ao requisito funcional do usuário. Igualmente, cadastrar um empregado sem que também sejam cadastrados dados de empregados assalariados ou horistas e dependentes não atende ao requisito funcional do usuário. Portanto, os nós folha "Cadastrar Empregado", "Cadastrar Horista", "Cadastrar Assalariado" e "Cadastrar Dependentes" devem ser conectadas por um conector de inclusão (ver figura 7).

Fonte: Marcos Freitas Junior, 2015

5.2.3.3 Atributo de nó

Esta seção apresenta os atributos de nó folha. Trata-se de descrições textuais das informações

exibidas e/ou informadas por um usuário do software durante a execução das funções elementares. Esse detalhamento deve ser realizado para todas as funções elementares modeladas na Árvore de pontos de função. Como exemplo, considere as abas visíveis para o protótipo de tela da figura 8, que representa a função elementar “Cadastrar Empregado”, exibida na figura 7. Nesse caso, os seguintes atributos de nó devem ser informados: código; nome; CPF; sexo; estado civil; RG; nascimento; CEP; endereço; número; complemento; bairro; cidade; estado; telefones; e-mail; categoria; status (ativo/bloqueado); nome dependente; CPF dependente; RG dependente; telefone dependente, foto e idade. Além disso, as seguintes informações adicionais devem ser incluídas para cada atributo de nó informado:

- Se o atributo de nó é exibido exclusivamente no cabeçalho ou rodapé da tela.
- Se o atributo de nó está disponível apenas para leitura de dados pelo usuário.
- Se o atributo de nó está disponível para entrada de dados pelo usuário.
- Se o atributo de nó informado isoladamente tem ou não sentido para um usuário de processo de negócio.

Figura 8 – Exemplo de tela para cadastro de empregado que ilustra atributos de nó da Árvore de pontos de função

The screenshot shows a web-based form titled "Cadastro" for registering an employee. The form is organized into several sections:

- Personal Information:** Fields for Código (2), Nome (JOSÉ CARLOS MENEZES), CPF (000.000.000-00), Sexo (MASCULINO), Est. Civil (CASADO(A)), RG (14.654.770), Nascimento (02/04/1983), and Idade (29A 2M 28D).
- Contact Information:** Fields for CEP (37701-010), Endereço (RUA PREFEITO CHAGAS), Nº (459), Compl (501), Bairro (CENTRO), Cidade (POÇOS DE CALDAS), and Estado.
- Professional Details:** Fields for E-mail (josemenezes@jmenezes.com.br) and Categoria (VIP). There are checkboxes for "Ativo" (checked) and "Bloqueado" (unchecked).
- Dependents:** A table with columns for Name, CPF, RG, and Telephone. One dependent is listed: LUCIANA MENEZES with CPF 000.000.000-00 and Telephone (35)9999-0000.
- Navigation and Actions:** A sidebar on the right contains icons for "Pesquisar", "Dependentes", "Localizações", "Reservas", and "Contas". At the bottom, there is a row of buttons: Adicionar, Alterar, Excluir, Salvar, Desfazer, Início, Anterior, Próximo, Último, and Fechar.

5.2.4 Relacionamentos entre níveis

Esta seção apresenta as formas de relacionar nós de diferentes níveis entre si, ou seja, nós intermediários com nós folha. Quando nós estão relacionados entre si, é necessário informar obrigatoriamente: rótulos de entidade de dados; atributos de rótulo de entidade; e, marcadores de rótulo de entidade. Os três elementos são descritos nas próximas seções.

5.2.4.1 Rótulo de entidade de dados

Esta seção apresenta o rótulo de entidade de dados. Trata-se de uma representação gráfica para representar cada entidade de dados do software. Para cada rótulo de entidade de dados, é necessário informar todos os marcadores de rótulos de entidade, os conectores de exclusão (quando aplicável) e os atributos de rótulo de entidade, obrigatoriamente. Esses três elementos são descritos nas próximas seções.

5.2.4.2 Marcador de rótulo de entidade

Esta seção apresenta os marcadores de rótulo de entidade a serem associados ao rótulo de entidade de dados. Eles indicam a presença de determinadas características para as entidades do software. Sempre que um rótulo de entidade de dados apresentar uma ou mais das características indicadas pelos marcadores de rótulos de entidade, esses devem ser incorporados ao respectivo rótulo de entidade de dados, obrigatoriamente. Há cinco marcadores de rótulos de entidade possíveis, apresentados no quadro 6.

5.2.4.3 Conector de exclusão

Esta seção apresenta o conector de exclusão para o rótulo de entidade de dados. Trata-se de uma representação gráfica que evidencia a existência de possíveis relacionamentos, e seu tipo, entre dois rótulos de entidade de dados.

O quadro 7 apresenta as definições de quando um conector de exclusão deve ser usado na Árvore de pontos de função. Sempre que o relacionamento entre rótulo de entidade de dados se encaixar nas definições apresentadas no quadro 7, então o uso do conector de

exclusão é obrigatório. Um mesmo rótulo de entidade de dados pode estar relacionado a diferentes rótulos de entidade de dados, por meio de diferentes conectores de exclusão.

Quadro 6 – Descrição dos marcadores de rótulos de entidade de dados

Nome	Ícone	Definição	Exemplo
EXTR		Ser mantida por um software externo , e apenas referenciada pelo software em questão.	Classificador de rótulo de entidade de dados TAXA, mantida por outro software e apenas referenciado pelo SISRH (ver figura 7) para calcular os descontos a serem realizados na folha salarial (ver apêndice A).
TIPO		Representar um tipo de entidade que armazenam dados de fundamental relevância a um usuário.	Entidade “Empregado” (ver figura 7).
SUBT		Representar uma subdivisão de um tipo de entidade. Um subtipo herda todos os atributos e relacionamentos de seus tipos de entidade pai, e pode ter atributos adicionais e relacionamentos próprios (IFPUG, 2010).	Entidade “Horista” (ver figura 7).
ATRI		Representar um tipo de entidade de dados (atributiva) que descreve complementarmente uma ou mais características de outra entidade.	Entidade “Dependente” (ver figura 7).
ASSO		Representar um tipo de entidade (associativa) que contém atributos e que descreve um relacionamento muitos-para-muitos entre dois outros tipos de entidade.	Entidade “Cargo” (ver figura 7).

Fonte: Marcos Freitas Junior, 2015

5.2.4.4 Atributo de rótulo de entidade

Esta seção apresenta os atributos de rótulo de entidade. Trata-se de descrições das informações armazenadas nas entidades de dados do software, representadas pelos rótulos, e usadas pelas funções elementares do software.

Para incluir os atributos de rótulo de entidade, devem ser considerados todos os dados armazenados pelo software e necessários para executar as funções elementares associadas. Por exemplo, para a entidade de dados “Cargo”, modelada como um rótulo de entidade de

dados (veja figura 7), deve ser armazenado pelo software os seguintes atributos, a serem usados posteriormente pelas funções elementares: “CBO”, “Cargo”, “Salário” e “Descrição”. Adicionalmente, é necessário informar, para cada atributo de rótulo de entidade, se ele é reconhecido por um usuário de processo de negócio, isoladamente.

Quadro 7 – Descrição do conector de exclusão

Representação	
Descrição	Deve ser usado para conectar um rótulo de entidade de dados com outros que o compõe, de modo que, a entidade posterior (depois da seta), é parte constituinte da anterior (contrária à seta). Se o rótulo contrário a seta for removido, os rótulos conectados por meios de conectores de exclusão serão removidos juntamente já que fazem parte do outro e não existem independentemente sem estarem vinculados ao rótulo de entidade de dados anterior. Deve ser usada apenas para conectar rótulo de entidade de dados entre si.
Exemplo	Considere o requisito funcional três do SISRH (ver RF3 do apêndice A). Se o requisito funcional do usuário exige que, ao excluir um empregado, também sejam excluídos os dependentes do empregado e os dados de assalariado ou horista; então os rótulos de entidades de dados dependente, assalariado e horista não tem significado por si só sem a presença de um empregado vinculado. Neste caso, "Empregado", "Horista", "Assalariado" e "Dependente" devem ser conectados via conector de exclusão (ver figura 7).

Fonte: Marcos Freitas Junior, 2015

5.3 Uso da Árvore de pontos de função na contagem de pontos de função

Os dados contidos na Árvore de pontos de função devem ser usados para a contagem dos pontos de função de acordo com as regras de APF. Para isso, regras de mapeamento devem ser usadas para que os dados modelados na árvore sejam usados como entrada para APF. Esta seção descreve tal mapeamento, seguindo os passos da contagem de pontos de função, conforme procedimento descrito na seção 2.2.

5.3.1 Obter documentação disponível

O processo de contagem de pontos de função inicia-se com a obtenção da documentação do software a ser medido. Em APF-APF, o único artefato de entrada necessário

para a contagem é a *Árvore de pontos de função*, conforme ilustrado no fluxograma 3. As informações contidas nesse artefato devem ser usadas para embasar a execução dos demais passos para a contagem de pontos de função, de acordo com as regras do CPM 4.3.1, conforme as seções seguintes.

5.3.2 Determinar escopo e fronteira da contagem

O propósito da contagem é representado em APF-APF pelos marcadores de raiz ESFO, CUST, PROD e AÇÃO. Para que o propósito da contagem seja identificado, é preciso verificar quais desses marcadores de raiz estão representados na raiz da *Árvore de pontos de função*. O tipo da contagem é representado em APF-APF pelos marcadores de raiz NOVO, MANT ou APLIC. Para que o tipo da contagem seja identificado, deve-se verificar qual desses marcadores de raiz está representado na raiz da árvore. O escopo da contagem deve ser identificado a partir dos nós folha e rótulos de entidade de dados. Todos os nós folha e rótulos de entidade de dados modelados devem ser considerados como pertencentes ao escopo da contagem. A fronteira da contagem é identificada a partir da raiz da *Árvore*, de modo que, cada raiz representa uma fronteira.

5.3.3 Medir funções de dados

Esta seção descreve como medir funções de dados, com base nas informações da *Árvore de pontos de função*.

5.3.3.1 Identificar as funções de dados

Todos os rótulos de entidade de dados da *Árvore de pontos de função* podem ser identificados como funções de dados em APF. Para isso, é necessário que:

1. O rótulo de entidade de dados não seja do tipo dados de código. Para isso, desconsideram-se da contagem rótulos de entidade de dados que tiverem associados a si nós folha com marcadores de nó CONS.
2. O rótulo de entidade de dados armazene ao menos dois atributos reconhecidos isoladamente por um usuário de processo de negócio. Para isso, desconsideram-se

da contagem rótulos de entidade de dados que contenham menos de dois atributos de rótulo de entidade reconhecidos por usuários de processos de negócio.

3. O rótulo de entidade de dados com marcador de rótulo de entidade “ASSO” não contenha apenas chaves estrangeiras. Assim, desconsideram-se da contagem rótulos de entidade de dados que possuam o marcador de nó “ASSO” e menos de três atributos de rótulos de entidade reconhecidos por um usuário de processo de negócio.
4. O rótulo de entidade de dados seja independente logicamente dos demais rótulos de entidade de dados. Para isso, deve-se Agrupar rótulos de entidade de dados ligados via conectores de exclusão e que não contenham a ele associados nós folhas com marcadores de nó CONS; menos de dois atributos de rótulo de entidade de dados reconhecidos por usuários de processos de negócio e marcadores de rótulo de entidade ASSO com menos de três atributos de rótulo de entidade. Os rótulos de entidade de dados não ligados via conectores de exclusão devem ser considerados como funções de dados.

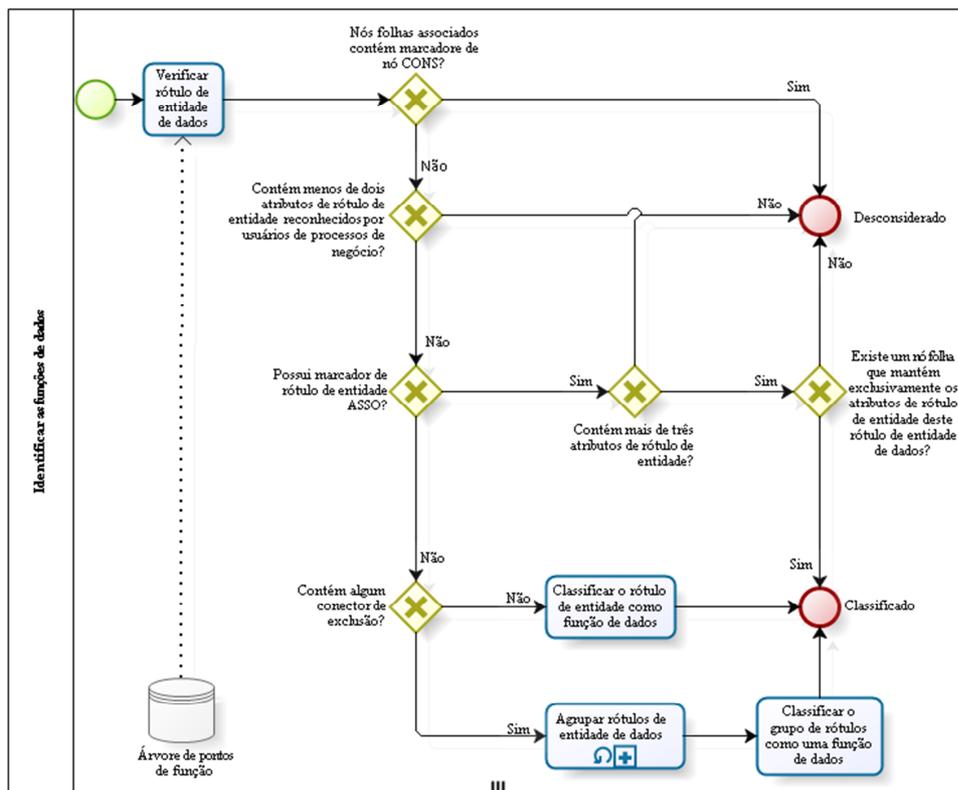
O fluxograma 4 ilustra o procedimento para identificar as funções de dados, seguindo as regras de APF com base nas informações da árvore e o fluxograma 5 apresenta detalhadamente o subprocesso para Agrupar rótulos de entidade de dados ligados via conectores de exclusão.

5.3.3.2 Classificar cada função de dados

Na sequência, cada função de dados deve ser classificada como ALI ou AIE. A classificação é realizada com base nas funções de transação que manipulam seus dados. Para isso, observam-se as seguintes regras:

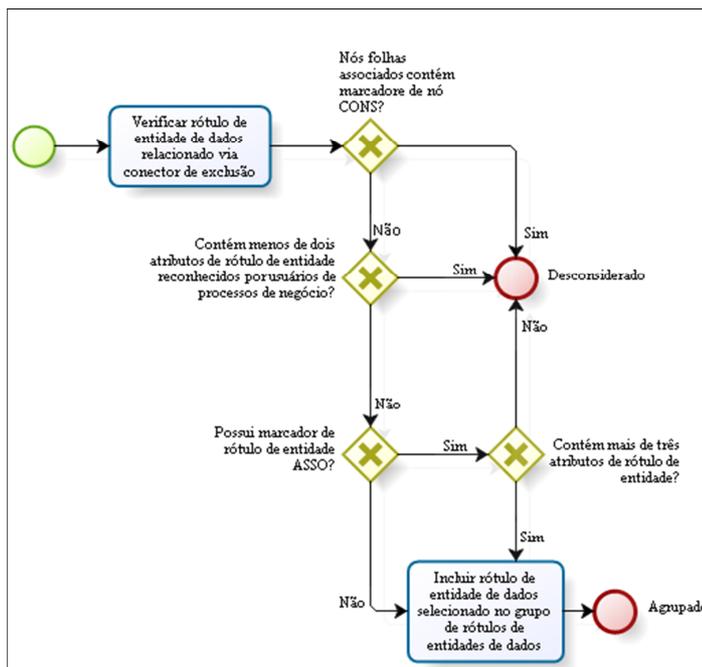
- Funções de dados, cujos rótulos de entidade de dados constituintes possuam – individualmente ou em conjunto a eles associados – ao menos uma função de transação que contenha os marcadores de nó MANT, devem ser classificadas como ALI.
- Funções de dados, cujos rótulos de entidade de dados constituintes possuam – individualmente ou em conjunto a eles associados – ao menos uma função de transação que contenha os marcadores de nó EXTR, devem ser classificadas como AIE.

Fluxograma 4 – Procedimento para identificar as funções de dados



Fonte: Marcos Freitas Junior, 2015

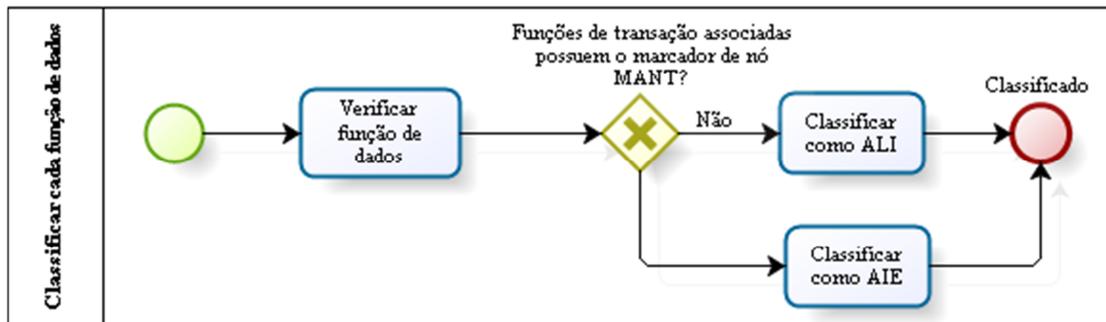
Fluxograma 5 – Detalhamento de subprocesso para Agrupar rótulos de entidade de dados



Fonte: Marcos Freitas Junior, 2015

O fluxograma 6 ilustra os procedimentos necessários para classificar funções de dados.

Fluxograma 6 – Procedimento para classificar cada função de dados



Fonte: Marcos Freitas Junior, 2015

5.3.3.3 Determinar complexidade funcional

Na sequência, a complexidade funcional deve ser determinada, com base no número de DERs, RLRs e usando as tabelas originais de APF (ver Tabelas 3, 4, 5 e 6).

Para calcular o número de RLRs de uma função de dados, observa-se a classificação dos rótulos de entidade de dados realizada pelo usuário. Para isso, observam-se as seguintes regras:

- Cada rótulo de entidade de dados classificado como “tipo” deve ser considerado um RLR da função de dados, desde que não tenha em seu grupo lógico de dados uma entidade subtipo. Nesse caso, o rótulo de entidade de dados classificado como “tipo” não deve ser considerado um RLR da função de dados.
- Cada rótulo de entidade de dados classificado como entidade “subtipo”, “atributiva” ou “associativa” deve ser considerado um RLR da função de dados.

Além disso, para calcular o número de DERs de uma função de dados, são contados os atributos de rótulo de entidade que integram a função de dados e que são reconhecidos isoladamente por um usuário de processo de negócio.

5.3.3.4 Determinar a contribuição

Na sequência, a partir do número de DERs e RLRs calculados nos passos anteriores, determina-se a complexidade funcional e a contribuição em termos e tamanho funcional de

cada função de dados, usando as tabelas originais de APF (ver Tabelas 3, 4, 5 e 6).

5.3.4 Medir funções de transação

Esta seção descreve como medir funções de transação, com base nas informações da Árvore de pontos de função.

5.3.4.1 Identificar os processos elementares do software

Todos os nós folha da Árvore de pontos de função, chamados de funções elementares, podem ser identificados como processos elementares em APF. Para isso, é necessário que:

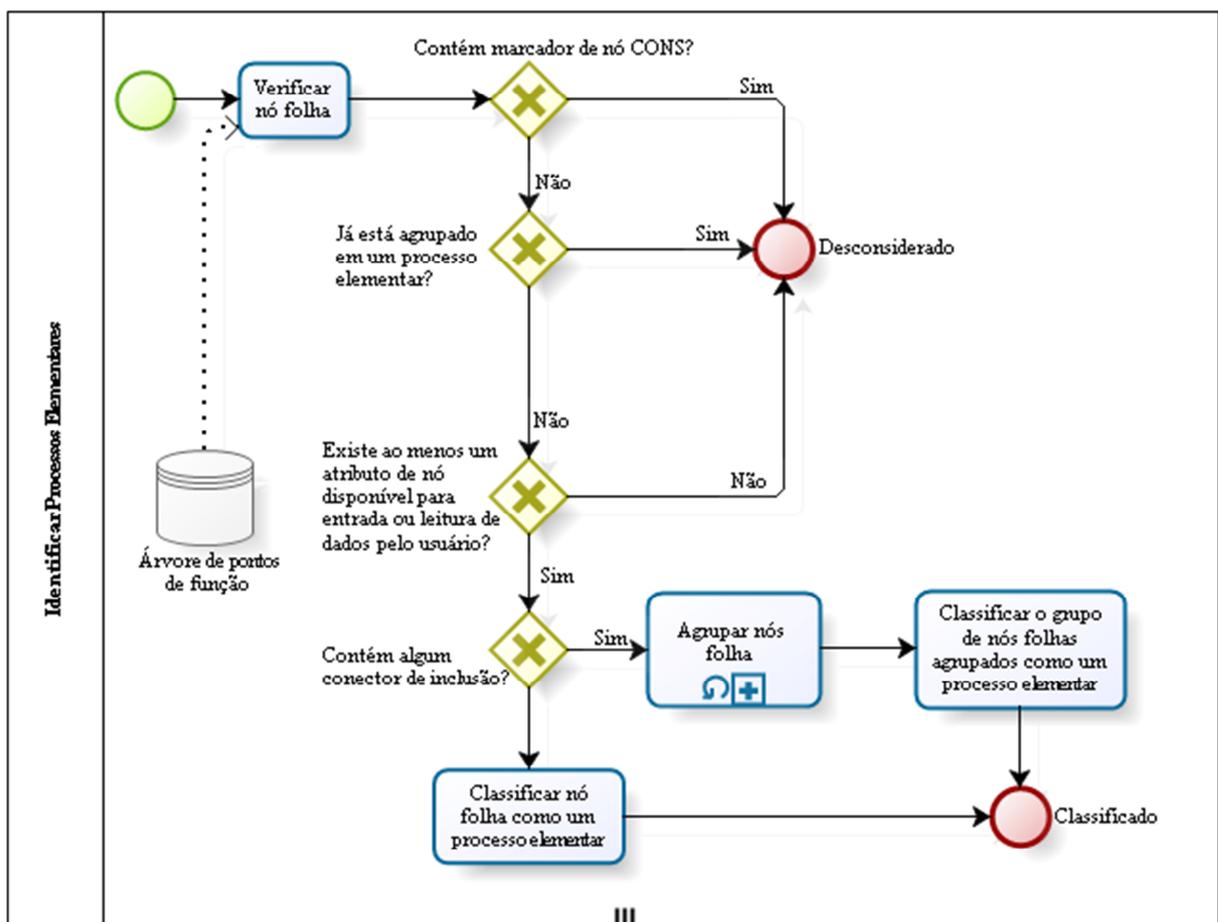
1. A função elementar, representada pelo nó folha, não mantenha ou consulte dados de código (conforme descrito na seção 2.2.3). Para isso, desconsideram-se da contagem todos os nós folha que possuam o marcador de nó CONS.
2. A função elementar representada pelo nó folha processe dados ou informações de controle recebidos de fora da fronteira do software ou envie dados ou informações de controle para fora da fronteira do software. Para isso, desconsideram-se da contagem todos os nós folha que não possuam ao menos um atributo de nó folha disponível para entrada ou leitura de dados pelo usuário do software.
3. A função elementar, representada por um nó folha ou um conjunto de nós folha, seja logicamente independente das demais funções elementares, representada por outros nós folha ou um conjunto de nós folha. Isso significa que a função elementar deve deixar o software em um estado consistente com sua execução, constituir uma transação completa, ser autocontida e ser a menor unidade de atividade com significado para o usuário (conforme descrito na seção 2.2.4). Para isso, deve-se Agrupar nós folha ligados via conectores de inclusão. Os nós folha não ligados via conectores de inclusão devem ser considerados como funções de transação.

Como exemplo, considere que as funcionalidades “Cadastrar Empregado”, “Cadastrar Horista”, “Cadastrar Assalariado” e “Cadastrar Dependente” foram modeladas como quatro funções elementares nos nós folha da árvore (ver figura 7). Os quatro nós folha estão relacionados por um conector de inclusão, já que o requisito funcional exige que, ao se cadastrar um empregado, também sejam cadastrados os dados dos empregados horistas ou

assalariados, assim como os dados de seus dependentes. Considerando a dependência lógica entre as funções elementares, elas devem ser classificadas como um único processo elementar, chamado de “Cadastrar Empregado\Cadastrar Horista\Cadastrar Dependente”.

O fluxograma 7 ilustra o procedimento para identificar os processos elementares do software, seguindo as regras de APF com base nas informações da árvore. O fluxograma 8 ilustra o procedimento para agrupar nós folha ligados via conectores de inclusão.

Fluxograma 7 – Procedimento para identificar os processos elementares do software



Fonte: Marcos Freitas Junior, 2015

5.3.4.2 Determinar os processos elementares únicos

Após identificar os processos elementares, deve-se identificar os processos elementares únicos (ver seção 2.2.4). Assim, para cada par de processo elementar, com base nas funções elementares da Árvore de pontos de função, verifica-se se possuem os mesmos:

Fluxograma 8 – Detalhamento de subprocesso para Agrupar nós folha

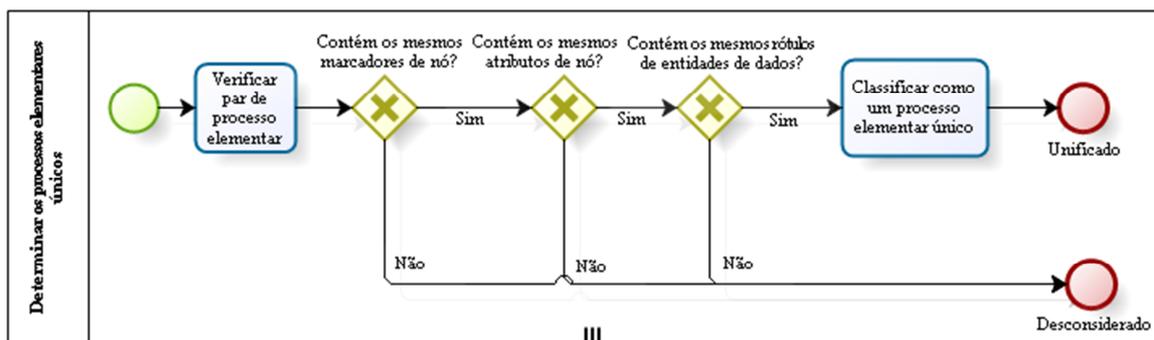


Fonte: Marcos Freitas Junior, 2015

1. Marcadores de nó, que representam as lógicas de processamento do processo elementar.
2. Atributos de nó, que representam os DERs do processo elementar.
3. Rótulo de entidades de dados, que representam os ALRs do processo elementar.

Caso um par de processo elementar tenha os mesmos atributos de nó, marcadores de nó e rótulo de entidade de dados, ambos devem ser classificados como um processo elementar único. O fluxograma 9 ilustra o procedimento para determinar os processos elementares únicos, seguindo as regras de APF com base nas informações da árvore.

Fluxograma 9 – Procedimento para determinar os processos elementares únicos



Fonte: Marcos Freitas Junior, 2015

Ao final dessa etapa, cada processo elementar único é considerado uma função de transação.

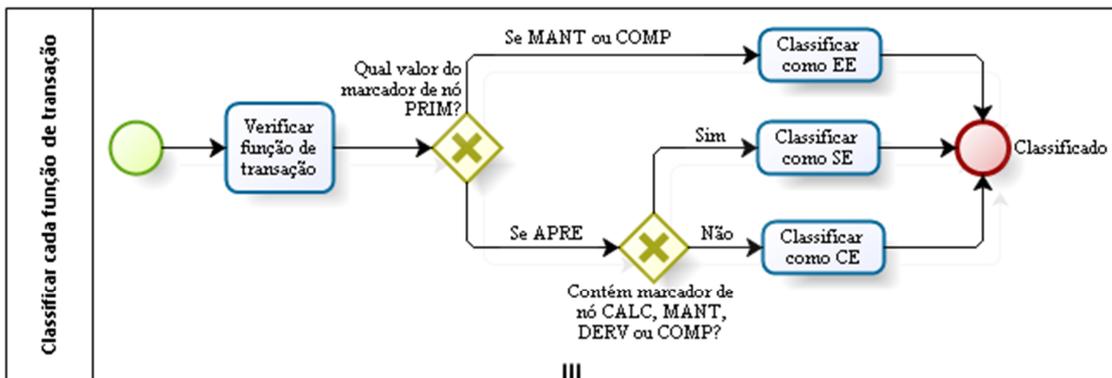
5.3.4.3 Classificar cada função de transação

Na sequência, classifica-se cada função de transação como EE, SE ou CE, baseada em sua intenção primária (marcador de nó PRIM). Para isso, as seguintes regras devem ser usadas, com base nas funções de transação identificadas a partir da Árvore de pontos de função:

- **EE**: funções de transação cujas funções elementares constituintes possuam, individualmente ou em conjunto, o marcador PRIM com valor MANT ou COMP.
- **SE**: funções de transação cujas funções elementares constituintes possuam, individualmente ou em conjunto, os marcadores de nó CALC, MANT, COMP ou DERV, e atributos de nó, estando ao menos um disponível para leitura pelo usuário.
- **CE**: funções de transação cujas funções elementares constituintes não possuam, individualmente ou em conjunto, os marcadores de nó CALC, MANT, COMP ou DERV, mas que possuam atributos de nó, estando ao menos um deles disponível para leitura pelo usuário.

O fluxograma 10 ilustra o procedimento para determinar os processos elementares únicos, seguindo as regras de APF com base das informações da árvore.

Fluxograma 10 – Procedimento para classificar cada função de transação



Fonte: Marcos Freitas Junior, 2015

5.3.4.4 Determinar a contribuição

Na sequência, determinam-se a complexidade funcional e a contribuição em termos de tamanho funcional de cada função de transação, com base no número de DERs e de ALRs e usando as tabelas originais de APF (ver tabelas 3, 4, 5 e 6).

Para calcular o número de DERs de uma função de transação, contam-se os atributos de nó, reconhecidos isoladamente por um usuário de processo de negócio, associados às funções elementares que integram a função de transação.

Para calcular o número de ALRs de uma função de transação, observam-se os rótulos de entidades de dados e as entidades externas relacionadas às funções elementares que constituem a função de transação. Cada rótulo de entidade relacionado deve ser considerado um ALR desde que ele tenha sido identificado como uma função de dados (ver passo 4.3.6.1). Além disso, cada entidade externa informada pelo usuário deve ser considerada um ALR desde que ele tenha sido identificado como uma função de dados durante a modelagem do respectivo software (ver passo 4.3.6.1).

5.3.5 Calcular tamanho funcional, e documentar e relatar

O cálculo do tamanho deve ser realizado por meio da aplicação das fórmulas adequadas ao tipo de contagem informado (ver seção 2.2.5). Por fim, a documentação dos dados que sustentam a contagem realizada deve ser realizada de acordo com os procedimentos descritos anteriormente para documentação do propósito, escopo, tipo de contagem, nome de funções de transação e de dados, lógicas de processamento executadas pelas funcionalidades, etc.

5.4 Considerações finais

Este capítulo foi destinado a apresentar a abordagem APF-APF para a realização da contagem de pontos de função com base na Árvore de pontos de função. O objetivo de APF-APF é realizar a contagem de modo mais sistemático, garantido a confiabilidade, a validade e a eficiência das medições realizadas. APF-APF foi desenvolvida visando mitigar os problemas relatados para APF e também aqueles observados nas abordagens que tentaram anteriormente resolver esse problema. Para isso, foi apresentado como o tamanho funcional do software deve ser obtido a partir da modelagem da Árvore de pontos de função. No capítulo seguinte, o protótipo de ferramenta desenvolvido para automatizar os procedimentos descritos neste capítulo é apresentado bem como os resultados da avaliação desta abordagem.

6. Prototipação e avaliação da abordagem APF-APF

Visando atender um dos objetivos buscados com este trabalho, um protótipo de ferramenta foi desenvolvido para apoiar a execução de APF-APF, apresentada no capítulo 5. Esse protótipo de ferramenta possui apenas o objetivo de validar a viabilidade de automação de APF-APF e servir para apoiar a realização de sua experimentação. Assim, não há o objetivo de desenvolver uma ferramenta com fins comerciais, incluindo preocupações com usabilidade, desempenho, entre outros. Por esse motivo, simplificações na apresentação das informações da Árvore de pontos de função e na inclusão dos marcadores de raiz, nós e rótulos de dados foram realizadas para facilitar o desenvolvimento do protótipo da ferramenta.

A avaliação de APF-APF, por meio do uso do protótipo de ferramenta desenvolvido, será realizada com a especificação funcional de um software de Recursos Humanos (RH), com 125 pontos de função calculados oficialmente pelo IFPUG. A especificação funcional será analisada por 11 analistas de sistemas que elaborarão a Árvore de pontos de função para que posteriormente seja usada em APF-APF para o cálculo do tamanho funcional do software de RH. A seguir, é apresentada uma visão geral do protótipo desenvolvido e da avaliação da abordagem de APF-APF realizada.

6.1 Protótipo de ferramenta

Esta seção apresentará os detalhes do protótipo de ferramenta desenvolvido para validar APF-APF. Os requisitos do protótipo, as premissas para o seu desenvolvimento, a arquitetura usada e a descrição das principais funcionalidades são descritos a seguir.

6.1.1 Visão geral do protótipo de ferramenta

Um protótipo de ferramenta foi desenvolvido com o objetivo de aumentar a eficiência da contagem de pontos de função, sem que a confiabilidade ou a validade em relação à APF original sejam prejudicados. O protótipo de ferramenta possui dois módulos, responsáveis por apoiar, respectivamente a modelagem da Árvore de pontos de função e o cálculo de pontos de função com base na árvore, conforme descritos a seguir:

1. **Modelagem da Árvore de pontos de função:** responsável por apoiar o usuário na elaboração da árvore, incluindo: (i) a raiz e os marcadores de raiz aplicáveis; (ii)

nós intermediários; *(iii)* nós folha com marcadores de nó, conectores de inclusão e atributos de nó; e *(iv)* rótulos de entidade de dados com atributos de rótulo de entidade, marcadores de rótulo de entidade de dados e conectores de exclusão.

2. **Cálculo de pontos de função com base na Árvore de pontos de função:** responsável pelo cálculo automatizado do tamanho funcional do software, de acordo com as regras de mapeamento apresentadas na seção 5.3, apresentando ao final:
 - a. Os nomes das funções de dados e das funções de transação medidas;
 - b. O número de DERs e RLRs, medidos para cada função de dados, e o número de DERs e de ALRs, medidos para cada função de transação.
 - c. A lista de DERs e RLRs para cada função de dados e a lista de DERs e ALRs para cada função de transação.
 - d. A complexidade e o tamanho funcional de cada função de dados e de transação.
 - e. A classificação de cada funcionalidade como ALI, AIE, EE, SE ou CE.
 - f. O resultado final da contagem em pontos de função, considerando o tipo de projeto em questão.

Uma vez que a principal contribuição deste trabalho é a definição de uma abordagem não necessariamente automatizada, o protótipo de ferramenta elaborado representa simplificada a abordagem, apresentada no capítulo 5. Foram realizadas simplificações na apresentação dos marcadores de raiz e marcadores de nó, representados no protótipo de ferramenta por descrições textuais em vez de ícones, conforme exibido anteriormente na figura 7. Embora as simplificações realizadas possam comprometer a usabilidade do protótipo de ferramenta, elas não comprometem a abordagem proposta.

6.1.2 Requisitos do protótipo de ferramenta

Após a identificação de lacunas por meio da análise de trabalhos relacionados, os requisitos funcionais para o protótipo de ferramenta foram definidos a partir das regras estabelecidas para a abordagem APF-APF (ver seção 5.3). A partir dos requisitos funcionais elicitados, elaborou-se um Diagrama de casos de uso, considerando o único papel que pode ser exercido por um usuário do protótipo de ferramenta: o analista de requisitos. O Diagrama de casos de uso do protótipo de ferramenta é apresentado na figura 9.

software do software em si (MILLER, 2008). Além dos componentes MVC padrão, optou-se pelo desenvolvimento de um componente adicional, denominado “*persistência*”, específico e dedicado à comunicação do software com o banco de dados, visando maximizar as vantagens obtidas pela separação lógica das camadas, previsto pelo MVC. As seções a seguir descrevem os quatro componentes citados.

6.1.4.1 Componente modelo

O componente modelo é composto por 17 classes, apresentadas no apêndice E desta dissertação. Esse componente centraliza as regras para a modelagem da Árvore de pontos de função, garantindo a correta execução de suas funcionalidades, e as regras de APF original. Ele é responsável por: (i) validar a consistência dos dados informados, impedindo, por exemplo, que um rótulo de entidade de dados com marcador de rótulo de entidade EXTR seja associado pelo usuário a um nó folha com marcador de nó MANT; (ii) validar as operações executadas, impedindo, por exemplo, que se crie mais de um nó raiz para a árvore ou que se conecte uma raiz diretamente a um nó folha; e (iii) aplicar as regras de APF original aos dados informados pelo usuário durante a modelagem, de modo que se identifique propósito, escopo, tipo de contagem, fronteira, funções de dados, funções de transação, DERs, ALRs, RLRs, complexidades e suas contribuições e o tamanho funcional do software medido.

6.1.4.2 Componente visão

O componente visão é composto por seis classes, descritas no apêndice E desta dissertação. Das seis classes, cinco são responsáveis por coletar dados como insumo para a realização da contagem e uma é responsável pela apresentação dos resultados da contagem após a aplicação das regras previstas pela abordagem APF-APF. Este componente é responsável por: (i) receber e enviar ao componente controle dados de: nó raiz, intermediário e folha; marcador de raiz, folha e rótulo de entidade; atributos de nó folha e de rótulo de entidade de dados; conectores de inclusão e os relacionamentos entre níveis; e dependência lógica entre as entidades modeladas no protótipo de ferramenta; e (ii) apresentar graficamente ao usuário informações recebidas do componente controle ao usuário do software.

6.1.4.3 Componente controle

O componente controle é composto por duas classes, descritas no apêndice E desta dissertação. Este componente é responsável pela intermediação da comunicação entre o componente modelo e o componente visão, sendo responsável por: (i) receber dados do componente visão e enviá-los ao componente modelo; e (ii) receber dados do componente modelo e enviá-los ao componente visão.

6.1.4.4 Componente persistência

O componente persistência é composto por três classes, descritas no apêndice E desta dissertação. O propósito desse componente é possibilitar que o componente modelo seja focado exclusivamente nas regras de negócio do software sem interferência de detalhes técnicos a respeito da conexão com o banco de dados, maximizando as vantagens oferecidas pelo padrão de projetos MVC. Ele é responsável por realizar a comunicação direta com o banco de dados executando operações, tais como: inclusões, alterações, exclusões e consultas dos dados modelados pelo usuário no componente de apresentação.

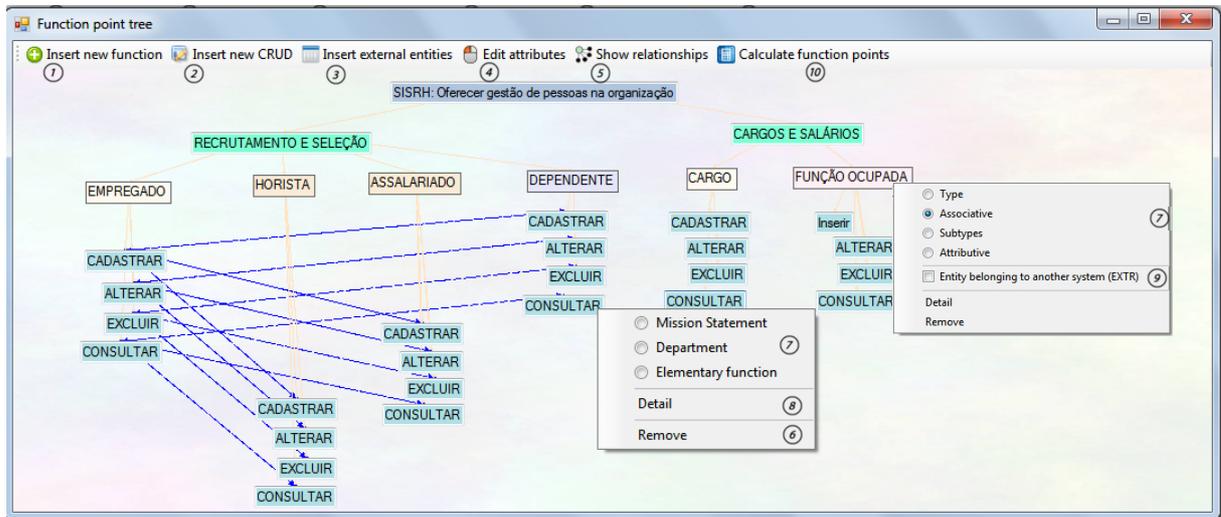
6.1.5 O protótipo de ferramenta desenvolvido

Esta seção apresenta o protótipo de ferramenta desenvolvido para este projeto. São apresentadas as principais interfaces destinadas aos Analistas de requisitos, destacando-se seus aspectos fundamentais. A interface com o Analista de requisitos é composta de seis telas, sendo quatro delas responsáveis por coletar dados do software e as outras duas responsáveis pela exibição dos resultados de contagem. A figura 10 apresenta a interface da tela inicial do software com a modelagem da Árvore de pontos de função finalizada para um software de gestão de RH, como exemplo ilustrativo, em que foram adicionadas numerações para identificar áreas e funcionalidades.

A seguir, estão descritas as funções associadas às áreas numeradas na figura 10:

1. Usada para inserir um novo nó do tipo raiz, intermediário ou folha.
2. Usada para adicionar, como sugestão inicial, quatro novos nós folha padrão denominados, respectivamente, “Cadastrar”, “Alterar”, “Excluir” e “Consultar” com o objetivo de tornar a modelagem mais produtiva.

Figura 10 – Interface principal do protótipo de ferramenta com a modelagem de um software de gestão de RH



Fonte: Marcos Freitas Junior, 2015

3. Usada para inserir um rótulo de entidade de dados que representa uma entidade de outro software, cujos dados são recuperados e usados pelo software sendo modelado. Em APF-APF, as entidades de outros softwares são representadas pelo marcador de rótulos de entidade de dados “EXTR”.
4. Usada para consultar ou remover atributos de nó folha ou de rótulo de dados.
5. Usada para consultar relacionamentos entre nós folha.
6. Usada para remover um nó.
7. Usada para informar o tipo de nó folha ou de rótulo de entidade de dados. Um nó folha pode ser classificado como “Missão do sistema”, que representa a raiz da árvore; “Departamento”, que representa um nó intermediário da árvore; ou uma “Função elementar”, que representa um nó folha da árvore. Esses três tipos são representados pelas cores lilás, verde e azul, respectivamente. E um rótulo de entidade de dados pode ser classificado como “Tipo”, “Associativa”, “Subtipo” ou “Atributiva”, representados pelas cores bege, rosa claro, roxo e rosa escuro, respectivamente.
8. Usada para apresentar os marcadores de nó folha e os atributos de nó folha (ver figura 11) ou os marcadores de rótulo de entidade de dados e os atributos de entidade de dados. Essa apresentação é realizada em outra tela e depende da classificação do nó selecionado. Diferentemente da proposta original de Árvore de pontos de função, a apresentação e inserção dos marcadores de nó folha, atributos

de nó folha, marcadores de rótulo de entidade de dados e atributos de entidade de dados foram alterados no protótipo de ferramenta, desenvolvido de modo que foram implementados apenas textualmente.

9. Usada para informar se a entidade modelada é externa ao software e apenas referenciada por ele. Essa funcionalidade é aplicável apenas a rótulo de entidade de dados.
10. Usada para realizar o cálculo de pontos de função do software quando a árvore estiver completamente preenchida.

A seguir, estão descritas as funções associadas às áreas numeradas na figura 11:

Figura 11 – Interface para incluir informações de uma funcionalidade modelada

Detail elementary function

Name of the function:

Leaf node markers (1)

References business data in related entities?
 Maintain business data in related entities?
 Retrieves data in related entities?
 Receives data informed by the user?
 Display data to users?
 To display data to the user the function performs some filter?
 Performs the conversion of data to meet a business rule?
 Performs validations to meet a functional requirement?
 Performs calculations to meet a business rule?
 Analyzes and verifies conditions which apply?

Is executed in batch (batch processing) to meet a functional requirement of the user?
 Displays message to the user?
 The amounts received by this function may influence their behavior?
 Generates additional data transforming existing data?
 Aims to maintain or display (exclusively) domain data or constants?

What is the main purpose of this feature?
 Maintain data in the software Change the behavior of the software Display data to users

Function attributes (2)

Attributes	Enabled for inclusion?	Displayed to the user?	This attribute alone makes sense for business?	Displayed in the header or footer?	It is important for the business user?
▶ CBO	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
CARGO	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
SALÁRIO	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
DESCRIÇÃO	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
*	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Done Remove Cancel

Fonte: Marcos Freitas Junior, 2015

1. Usada para informar a presença e/ou a ausência de marcadores de nó em relação a uma funcionalidade específica. Diferentemente da proposta original de Árvore de pontos de função (ver seção 5.2.3.1), em que os marcadores de nó são informados

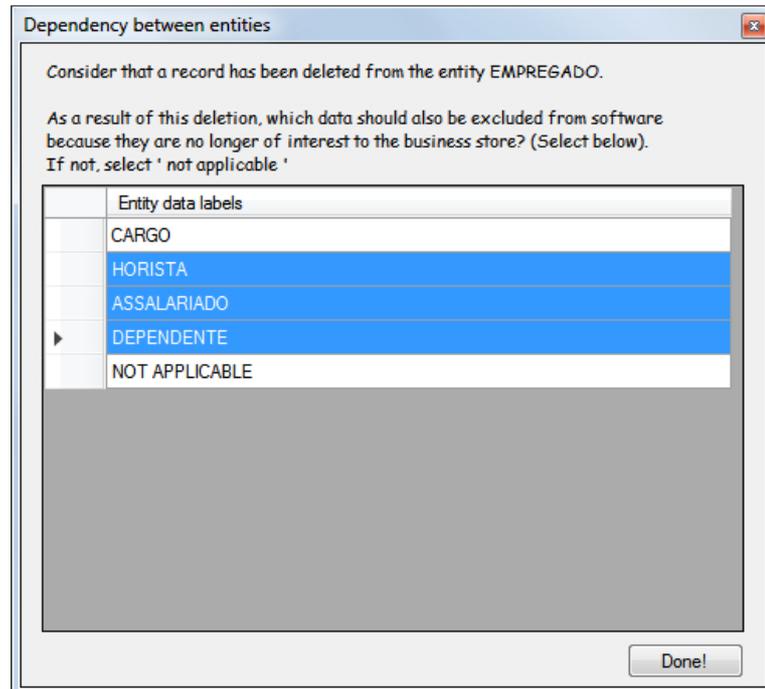
de forma gráfica por meio de ícones no protótipo da ferramenta esse marcadores, foram implementados apenas textualmente.

2. Usada para informar os atributos de nó da função elementar, bem como as características de cada atributo em relação a uma funcionalidade específica.
3. Usada para finalizar a inclusão das informações e possibilitar que a ferramenta armazene as informações em seu banco de dados.
4. Usada para remover um determinado atributo.

De modo semelhante, o usuário deve detalhar os rótulos de entidades de dados do software. A descrição da interface dessa parte do protótipo pode ser consultada no apêndice E.

Após a elaboração completa da árvore, a funcionalidade “Calcular pontos de função” (representada pela opção 6 da tela principal, ver figura 10) deve ser executada. Se todas as informações necessárias tiverem sido preenchidas corretamente, a ferramenta solicitará que o usuário responda a um questionário para identificar a dependência entre os rótulos de dados modelados na árvore, conforme exemplo ilustrativo apresentado na figura 12.

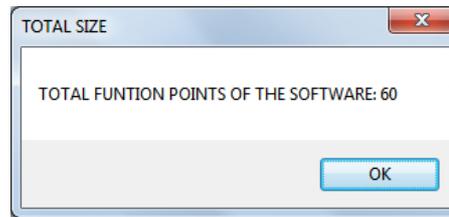
Figura 12 – Interface para seleção dos rótulos de entidades de dados dependentes entre si



Fonte: Marcos Freitas Junior, 2015

Por fim, a ferramenta apresenta o tamanho do software em pontos de função (ver figura 13) e o detalhamento das funções de dados e de transação identificadas (ver figura 14).

Figura 13 – *Pop-up* apresentando o tamanho total do software calculado pela ferramenta



Fonte: Marcos Freitas Junior, 2015

A seguir, estão descritas as funções associadas às áreas numeradas na figura 14:

1. Usada para apresentar o detalhamento da contagem realizada.
2. Usada para recuperar e apresentar a lista dos atributos que foram considerados DERs para uma determinada função de dados ou de transação selecionada.

Figura 14 – Interface para apresentar o detalhe do tamanho funcional do software calculado pela ferramenta

Name of the transactional function	Type	ALRs/RLRs	DERs	Complexity	Size
FUNÇÃO OCUPADA (FUNÇÃO OCUPADA)	ALI	1	6	Baixa	7
EMPREGADO (DEPENDENTE / ASSALARIADO / HORISTA)	ALI	3	14	Baixa	7
CARGO (CARGO)	ALI	1	5	Baixa	7
▶ CADASTRAR (CARGO)	EE	1	6	Baixa	3
ALTERAR (CARGO)	EE	1	6	Baixa	3
EXCLUIR (CARGO)	EE	1	3	Baixa	3
CONSULTAR (CARGO)	CE	1	6	Baixa	3
INSERIR (FUNÇÃO OCUPADA)	EE	1	8	Baixa	3
ALTERAR (FUNÇÃO OCUPADA)	EE	1	7	Baixa	3
EXCLUIR (FUNÇÃO OCUPADA)	EE	1	4	Baixa	3
CONSULTAR (FUNÇÃO OCUPADA)	CE	1	8	Baixa	3
CADASTRAR (EMPREGADO)	EE	1	16	Média	4
ALTERAR (EMPREGADO)	EE	1	21	Média	4
EXCLUIR (EMPREGADO)	EE	1	4	Baixa	3
CONSULTAR (EMPREGADO)	EE	1	16	Média	4

Fonte: Marcos Freitas Junior, 2015

6.2 Avaliação da abordagem APF-APF

Avaliações experimentais foram realizadas para analisar APF-APF em relação aos três objetivos específicos pretendidos para este trabalho: confiabilidade, validade e eficiência. Nas próximas seções, são apresentados os procedimentos adotados para o experimento bem como os resultados obtidos após sua execução.

6.2.1 Procedimento do experimento

Como procedimento, foram planejados e executados os seguintes passos:

1. Escolha aleatória de uma especificação de software elaborada e medida pelo Comitê de Práticas de Contagem do IFPUG ou por seus integrantes. A especificação continha a descrição de um software de RH com 125 pontos de função. O software continha funcionalidades para: *(i)* incluir, alterar, excluir, listar e consultar um empregado da organização (horista ou assalariado) e seus dependentes; *(ii)* incluir, alterar, excluir, listar e consultar as funções da organização; *(iii)* incluir, alterar, excluir, listar e consultar funções ocupadas por um empregado da organização; *(iv)* migrar os dados dos empregados do software de RH antigo, exibindo um relatório de todos os empregados migrados; e *(v)* gerar relatórios para consultar o total de empregados por país e/ou cidade.

2. Escolha aleatória de 11 analistas de sistemas/requisitos atuantes no mercado, com experiência entre um a oito anos, selecionados em duas organizações: *(i)* uma com sede no Rio de Janeiro, São Paulo e Brasília, atuante na área de Engenharia de software; e *(ii)* outra com sede em São Paulo, atuante na área de Desenvolvimento de sistemas. A escolha dos participantes de ambas as organizações se deu por conveniência, ou seja, não foram estabelecidas variáveis como idade, sexo ou cargo. As únicas condições estabelecidas para a participação foram: *(i)* formação em área de conhecimento relacionada a sistemas de informação e *(ii)* atuação profissional como analista de sistemas ou de requisitos.

3. Modelagem da Árvore de pontos de função para o software selecionado usando como insumo exclusivamente a especificação de software fornecida. Para todos os participantes, a especificação, as instruções em relação às regras da abordagem APF-APF e o protótipo de ferramenta desenvolvido foram enviados por e-mail (ver apêndice G). Para os profissionais da primeira organização, a modelagem da Árvore de pontos de função ocorreu durante horário comercial visando simular, do modo mais real, o uso de APF-APF no ambiente corporativo. Para a segunda organização, a modelagem da Árvore de pontos de função foi realizada fora de horário comercial, por indisponibilidade dessa atividade ser executada durante esse horário.

4. Aplicação da abordagem APF-APF sobre a mesma especificação de software por dois grupos de analistas. O primeiro grupo, composto de oito pessoas selecionadas na primeira organização, usando o protótipo de ferramenta desenvolvido e o segundo grupo,

composto de três pessoas atuantes na segunda organização, trabalhando de modo manual¹¹. Cada analista deveria registrar o tempo usado para a aplicação de APF-APF.

Para análise dos resultados obtidos, os seguintes critérios foram definidos:

- **Confiabilidade:** os tamanhos funcionais do software em pontos de função obtidos por meio da aplicação de APF-APF, por cada um dos analistas, de ambos os grupos (independentemente se realizaram a tarefa de forma manual ou com o apoio do protótipo de ferramenta), foram comparados entre si. Esperava-se obter um coeficiente de variação entre esses valores, em relação a sua média, menor que 5%, já que para APF original o desvio padrão costuma ser de até 10% (IFPUG, 2010).
- **Validade:** os tamanhos funcionais do software em pontos de função obtidos por meio da aplicação de APF-APF, por cada um dos analistas, de ambos os grupos (independentemente se realizaram a tarefa de forma manual ou com o apoio do protótipo de ferramenta), foram comparados com o resultado oficial divulgado pelo IFPUG. Esperava-se obter um coeficiente de variação, em relação ao resultado oficial, menor que 2%. Esse valor foi estabelecido tendo como referência o menor coeficiente de variação observado para os trabalhos relacionados que testaram as abordagens propostas (ADEM; KASIRUN, 2010; MIYAWAKI; IJIMA; HO, 2008; UEMURA; KUSUMOTO; INOUE, 2001; ABRAHÃO; INSFRÁN, 2008; FRATERNALI; BONGIO, 2006; EDAGAWA et al., 2011 e LAMMA; MELLO; RIGUZZI, 2004).
- **Eficiência:** os tempos gastos para a obtenção dos tamanhos funcionais do software em pontos de função obtidos por meio da aplicação de APF-APF por cada um dos analistas do grupo que realizou a tarefa de forma manual foi comparada com dos analistas do grupo que realizou a tarefa com o apoio do protótipo de ferramenta. Esperava-se obter um tempo médio gasto com apoio do protótipo da ferramenta 25% inferior ao tempo médio gasto de forma manual. Embora seja propósito dessa abordagem tornar a medição de pontos de função mais eficaz, comparada com a APF original, a ausência de dados em relação ao tempo de contagem do estudo de caso impossibilitou que esses valores fossem comparados. Considerou-se, portanto, que se

¹¹ Deve-se destacar que, o planejamento inicial consistia em separar homogeneamente ambos os grupos, de modo que, houvesse um equilíbrio entre: a quantidade de respondentes selecionados para ambas as organizações; a quantidade de respondentes utilizando a abordagem de manual e automatizada e uma melhor distribuição entre o perfil profissional dos respondentes. Entretanto, devido à indisponibilidade relatada para cinco respondentes da segunda organização, após a realização de experimento na primeira organização, não foi possível distribuir novamente os respondentes. Como consequência, algumas ameaças à validade deste experimento foram relatadas na seção 6.2.2.

a abordagem automatizada apresenta maior eficiência do que a abordagem manual, isso também ocorrerá para APF original.

6.2.2 Ameaças à validade do experimento

A seguir, as ameaças à validade interna e externa do experimento são detalhadas. As seguintes ameaças foram consideradas:

- Foram identificadas as seguintes ameaças à validade interna desse experimento que limitam o grau de confiança nos resultados obtidos com a amostra:
 - a. A seleção de respondentes dos grupos da organização um e dois não foi realizada de modo igualitário quantitativamente e qualitativamente. Essa ameaça ocorreu devido à evasão de respondentes da segunda organização com características específicas e relevantes ao experimento.
 - b. Embora o experimento tenha sido realizado em dois locais distintos (as respectivas organizações) membros de uma mesma organização podem ter influenciado outros respondentes que se encontravam no mesmo ambiente. Deste modo, há possibilidade de ter ocorrido troca de informações e discussões internas entre os respondentes, que homogeneizaram as modelagens de Árvore de pontos de função entregues e, conseqüentemente, os problemas observados.
 - c. Por ter sido realizado em ambiente corporativo durante horário comercial, os respondentes da primeira organização podem ter realizado o preenchimento da Árvore de pontos de função esperando alcançar um determinado resultado específico – positivo ou negativo – pelo fato de saberem que estavam sendo observados. Além disso, por se tratar de um ambiente propício, podem ter ocorrido interrupções para realização de atividades profissionais.
 - d. A modelagem da Árvore de pontos de função pode ter sido influenciada por conhecimentos prévios em APF por parte dos respondentes, que acabaram por realizar determinadas conexões ou inclusões de informações nesse artefato, sem que fossem levados em conta exclusivamente os requisitos funcionais do usuário descritos na especificação de software fornecida.
 - e. A implementação parcial do protótipo de ferramenta pode ter influenciado os resultados observados para o conjunto de respondentes que utilizaram o protótipo de ferramenta para modelagem da Árvore de pontos de função.

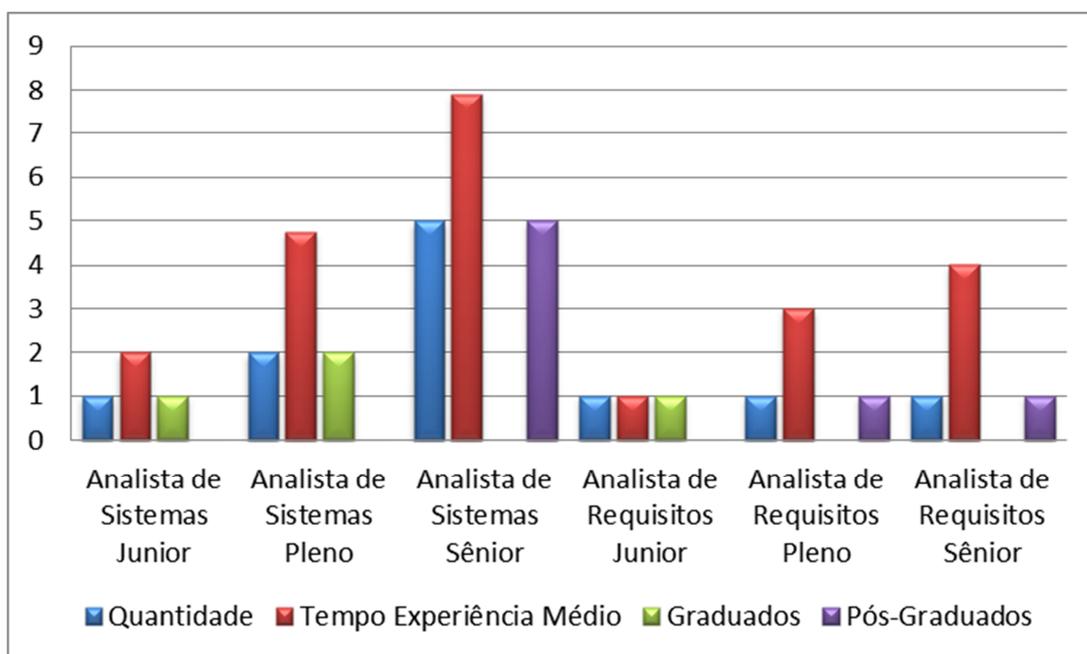
- f. As instruções em relação à abordagem APF-APF e em relação à utilização do protótipo de ferramenta podem não ter sido claras o suficiente para entendimento completo em relação às regras e componentes de APF-APF, bem como em relação à correta utilização do protótipo de ferramenta desenvolvido.
- Foram identificadas as seguintes ameaças a validade externa que limitam a capacidade de generalizar os resultados obtidos:
- g. Considera-se que a amostra de respondentes pode não ser representativa, uma vez que o experimento foi realizado com um baixo número de profissionais, de uma única região e que podem não ser representativos para projetos de software em geral.
- h. Foi selecionada apenas uma especificação de software de um sistema de RH que pode não representar a gama de projetos existentes dentro de um processo de desenvolvimento de software e a complexidade dos requisitos funcionais dentro de uma organização.
- i. Pode ter havido imposições de superiores, não declaradas explicitamente, em relação ao tempo a ser utilizado por cada respondente da organização um para elaboração da Árvore de pontos de função. A restrição de tempo possibilitaria que cada respondente retornasse o mais breve possível as suas atividades diárias.
- j. O experimento poderia ter sido diversificado para detectar influências específicas de variáveis na assertividade dos resultados gerados.

6.2.3 Resultados do experimento

O gráfico 2 apresenta o perfil profissional e a formação acadêmica dos respondentes que participaram do experimento. Dentre os onze participantes, a maioria (55%) era composta por analista de sistemas sênior, pós-graduados, com tempo médio de atuação nessa função de 7,9 anos. Apenas 27% dos participantes, que atuavam em sua totalidade na organização 2, representavam analistas de requisitos, função considerada mais adequada para elaboração da Árvore de pontos de função.

A tabela 8 apresenta o detalhamento dos perfis de cada respondente bem como o tempo usado por cada um para executarem os procedimentos previstos na abordagem APF-APF e o total de pontos de função calculados.

Gráfico 2 – Perfil dos respondentes selecionados para participar do experimento



Fonte: Marcos Freitas Junior, 2015

Dentre as modelagens da Árvore de pontos de função elaboradas, duas foram descartadas por não conter o detalhamento suficiente para que as regras de APF-APF fossem aplicadas: uma foi entregue via protótipo de ferramenta com o artefato Árvore de pontos de função sem nenhuma informação (em branco) e outra foi entregue em papel apenas com a estrutura da Árvore de pontos de função sem que, no entanto, fossem documentados os marcadores e atributo de nós folha e marcadores e atributos de rótulos de entidade de dados.

O desvio padrão e coeficientes de variação para os tamanhos calculados pela abordagem APF-APF em relação à média e ao resultado oficial; bem como a média do tempo utilizado na abordagem APF-APF são apresentados na tabela 9. Os dados foram calculados considerando visões distintas (agrupamentos de respondentes baseados em uma determinada característica).

6.2.4 Análise qualitativa de APF-APF considerando os resultados obtidos

A seguir, são apresentadas as análises dos resultados obtidos conforme descritos na seção anterior. Os resultados são analisados de acordo com os objetivos específicos pretendidos pela abordagem APF-APF.

Tabela 8 – Resultados obtidos após a execução dos experimentos¹²

	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	Média
Formação	B	B	PG	PG	PG	PG	B	PG	B	PG	PG	-
Cargo	ASJ	ASP	ASS	ASS	ASS	ASS	ASS	ASS	ARS	ARJ	ARP	-
Experiência (anos)	2	3,5	7	9,5	8	7	6	8	4	1	3	5,4
	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	Média
Tipo de execução	A	A	A	A	A	A	A	M	M	M	M	-
Tamanho (PF)	104	109	88	110	111	102	D	D	124	122	124	110
Tempo (horas)	1,7	2,5	2,6	1,1	2,3	2,1	-	-	3,5	4,2	3,8	2,64

Fonte: Marcos Freitas Junior, 2015

Tabela 9 – Resultados consolidados por visões após a execução dos experimentos¹³

Visão dos dados	Média de PF	DP baseado na média	CV (%)	DP baseado no resultado oficial	CV (%)	Média de tempo (horas)
01 Todos respondentes	110,44	11,84	10,72	19,45	17,61	2,64
02 Todos respondentes com exceção de R3	113,25	8,89	7,85	15,39	13,59	2,65
03 Respondentes abordagem automatizada	104,00	8,60	8,27	24,56	23,62	2,05
04 Respondentes abordagem automatizada sem R3	107,20	3,96	3,69	18,15	16,93	1,94
05 Respondentes abordagem manual	123,33	1,15	0,93	2,35	1,91	3,83
06 Respondentes com perfil de Analista de Sistemas	104,00	8,60	8,27	24,56	23,62	2,05
07 Respondentes com perfil de Analista de Requisitos	123,33	1,15	0,93	2,35	1,91	3,83

Fonte: Marcos Freitas Junior, 2015

¹² Legenda: P – Participante; B – Bacharel; PG – Pós-Graduado; ASJ – Analistas de Sistemas Júnior; ASP – Analista de Sistemas Pleno; ASS – Analista de Sistemas Sênior; ARJ – Analista de Requisitos Júnior; ARP – Analista de Requisitos Pleno; ARS – Analistas de Requisitos Sênior; A – Automatizada (via protótipo de ferramenta); M – Manual; D – Amostra descartada.

¹³ Legenda: DP – Desvio padrão; CV – Coeficiente de Variação; R – Respondente .

- **Confiabilidade:** o coeficiente de variação observado para os resultados de todos os respondentes, em relação a sua média (ver visão 1 da tabela 9), é consideravelmente superior ao pretendido (de até 5%) com a abordagem APF-APF em relação a sua confiabilidade. Quando os resultados são analisados sem o respondente 3 (que apresentou um tamanho funcional com grande afastamento dos demais) observa-se que o coeficiente de variação em relação à utilização da abordagem APF-APF pelos demais respondentes (ver visão 2 da tabela 9) apresenta melhores resultados quando comparados ao anterior. O mesmo ocorre quando o coeficiente de variação para a abordagem automatizada (ver visão 3 da tabela 9) é comparado com o mesmo conjunto de dados sem o respondente 3 (ver visão 4 da tabela 9). Observa-se, portanto, que os valores obtidos para esse respondente são desproporcionais em relação ao comportamento da amostra, influenciando negativamente os resultados observados. Em geral, nota-se que a dispersão relativa (coeficiente de variação) em relação à média demonstra homogeneidade dos resultados obtidos, já que apresentam um coeficiente de variação inferior a 20% (ZAMBERLAN, 2015). Embora os resultados observados não sejam, em geral, positivos, eles indicam, quando analisados de modo específico (ver motivos da variação ainda nessa seção) uma tendência para melhoria futura desses resultados em virtude de grande parte das Árvores de pontos de função terem sido realizadas sem todas as informações requeridas.
- **Validade:** o coeficiente de variação observado para os resultados de todos os respondentes (ver visão 1 da tabela 9) é superior ao pretendido (de até 2%) com a abordagem APF-APF em relação a sua validade. Quando os resultados são analisados sem o respondente 3 (que apresentou um tamanho funcional com grande afastamento dos demais) observa-se que o coeficiente de variação em relação à utilização da abordagem APF-APF pelos demais respondentes (ver visão 2 da tabela 9) apresenta melhores resultados quando comparados ao anterior. O mesmo ocorre quando o coeficiente de variação para a abordagem automatizada (ver visão 3 da tabela 9) é comparado com o mesmo conjunto de dados sem o respondente 3 (ver visão 4 da tabela 9). Observa-se, portanto que os valores obtidos para esse respondente são desproporcionais em relação ao comportamento da amostra, influenciando negativamente os resultados observados. Em geral, nota-se que embora o desvio padrão tenha sido alto em todas as visões, a dispersão relativa (coeficiente de variação) em relação a média demonstra que há homogeneidade dos resultados obtidos, já que apresentam um coeficiente de variação inferior a 20% (ZAMBERLAN,

2015). Embora os resultados observados não sejam, em geral, positivos, eles indicam, quando analisados de modo específico (ver motivos da variação ainda nessa seção), uma tendência para melhoria futura desses resultados, em virtude de grande parte das Árvores de pontos de função terem sido realizadas sem todas as informações requeridas.

- **Eficiência:** quando o tempo usado, entre os respondentes que usaram a abordagem manual (ver visão 3 da tabela 9) com os que usaram a abordagem automatizada (ver visão 5 da tabela 9), é comparado nota-se que quando é usado o protótipo de ferramenta desenvolvido há um ganho na eficiência da execução dos procedimentos previstos em APF-APF de aproximadamente 47% comparado à modelagem manual. Portanto, para esse quesito, APF-APF mostrou resultados consideravelmente superiores ao pretendido. Entretanto, quando analisados em contexto específico (ver motivos da variação ainda nessa seção) verifica-se que há uma tendência de menor eficiência para a abordagem APF-APF via protótipo de ferramenta em virtude de grande parte das Árvores de pontos de função terem sido realizadas sem todas as informações requeridas.

Nota-se que os coeficientes de variação observados para validade e confiabilidade quando a Árvore de pontos de função é elaborada por Analistas de sistemas apresenta resultados consistentes e aderentes ao pretendido com APF-APF. Entretanto, a distribuição dos perfis e do tipo de execução impossibilita identificar o que contribui positivamente para que os resultados calculados por Analistas de Requisitos via abordagem manual sejam consideravelmente melhores do que os observados para Analistas de Sistemas via protótipo de ferramenta. É possível supor que, há melhor entendimento dos requisitos do software e/ou dos conceitos de APF-APF quando Analistas de requisitos são responsáveis pela elaboração da Árvore de pontos de função. Entretanto, a utilização exclusiva de Analistas de Requisitos com a abordagem manual e de Analistas de sistemas com abordagem via protótipo de ferramenta impossibilita concluir que os melhores resultados sejam consequência do perfil dos profissionais e não do tipo de execução. Além disso, a quantidade de Analistas de requisitos usadas no experimento é baixa, não permitindo que os resultados sejam generalizados (ver seção 6.2.2).

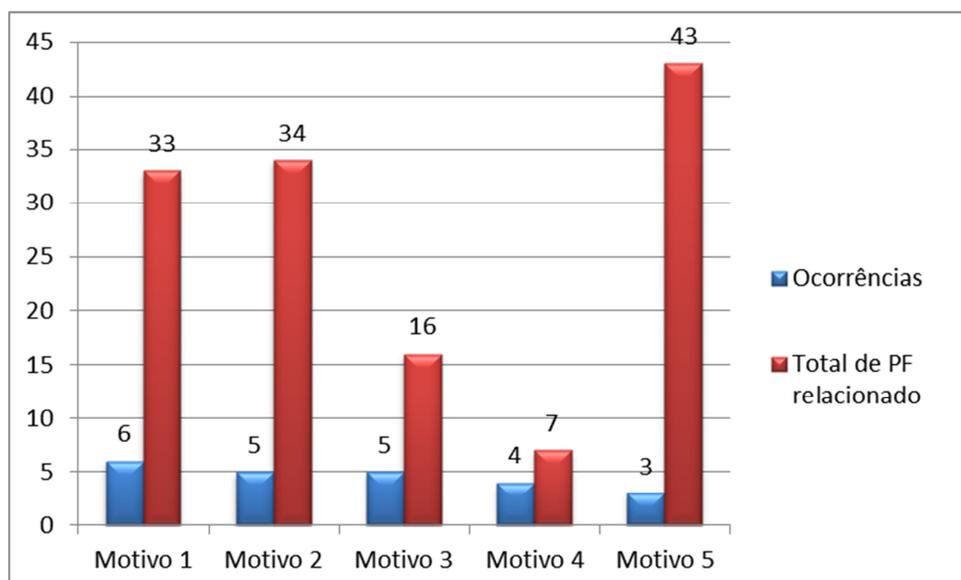
Embora quantitativamente a abordagem APF-APF tenha apresentado problemas em relação à confiabilidade e à validade dos resultados de medição obtidos no experimento realizado, verifica-se que, quando analisadas individualmente as modelagens realizadas por cada um dos participantes (ver apêndice F), a variação para os tamanhos calculados por APF-

APF é ocasionada por cinco motivos. Esses motivos, ordenados pela frequência em que eles foram observados nas Árvores de pontos de função elaboradas, são os seguintes:

1. Ausência de relacionamento entre nó folha e rótulos de entidade de dados usados durante sua execução para gravação ou recuperação de dados.
2. Ausência de funções elementares, descritas no estudo de caso, mapeadas como nós folha da Árvore de pontos de função.
3. Ausência do marcador de nó CALC em funções elementares que realizam cálculos funcionais.
4. Ausência de entidades, descritas no estudo de caso, mapeadas como rótulo de entidade de dados da Árvore de pontos de função.
5. Uso indevido do conector de inclusão para conectar funções elementares.

O número de ocorrências observadas para cada um dos motivos descritos acima bem como o total de pontos de função não calculado por APF-APF em virtude desses motivos é apresentado no gráfico 3. Na sequência, cada motivo é analisado individualmente.

Gráfico 3 – Motivos para a variação do coeficiente de confiabilidade e validade dos resultados de medição obtidos por APF-APF quando comparados com os coeficientes pretendidos



Fonte: Marcos Freitas Junior, 2015

1. **Relacionamento entre nós folha e rótulos de entidade de dados não modelado:** a ausência de relacionamento entre nós folha e todos os rótulos de entidade de dados usados durante sua execução para gravação ou recuperação de dados foi observada nas

Árvores de pontos de função elaboradas por todos os respondentes. O respondente 4, por exemplo, não relacionou o nó folha “Excluir Empregado” ao rótulo de entidade de dados “Função Ocupada” (ver figura 25 do apêndice F), de modo que, ao aplicar as regras de mapeamento para a contagem de pontos de função de APF-APF, esse rótulo de entidade de dados não foi considerado como um ALR da função de transação “Excluir Empregado”. Por definição da abordagem APF-APF, nós folhas devem possuir, quando aplicável, rótulos associados que descrevam qual entidade de dados do software é mantida ou referenciada pelas funções elementares relacionadas. Sendo assim, cada nó folha deveria ser relacionado com todos os rótulos de entidades de dados usados durante sua execução para gravação ou recuperação de dados.

2. **Marcador de nó CALC não modelado:** a ausência de informações em relação especificamente ao marcador de nó CALC foi observado para os respondentes 1, 2, 3 e 4. O respondente 2, por exemplo, ao detalhar os marcadores de nós para a função elementar “Listar Funções” não vinculou o marcador de nó CALC a essa funcionalidade (ver figura 21 do apêndice F), de modo que, ao aplicar as regras de mapeamento para a contagem de pontos de função de APF-APF, essa função elementar não foi classificada com o tipo “SE”. Por definição da abordagem APF-APF, o marcador de nó CALC deve ser usado sempre que a função elementar relacionada realizar ao menos um cálculo que atenda a uma regra de negócio. Na especificação de software fornecida, era descrito o seguinte requisito funcional do usuário: “*Permitir visualizar todas as funções e o total de funções da empresa.*”.

3. **Entidades não modeladas:** a ausência de informações em relação às entidades que compunham o software descrito na especificação de software fornecida foi observada para os respondentes 2, 3, 4, 5 e 6. O respondente 6, por exemplo, não modelou como rótulo de entidade de dados a entidade “Função Ocupada” (ver figura 29 do apêndice F), de modo que, ao aplicar as regras de mapeamento para a contagem de pontos de função de APF-APF, essa entidade não foi classificada como um ALI. Por definição da abordagem APF-APF, os rótulos de entidade de dados representam graficamente as entidades de dados do software sendo modelado. Sendo assim, cada entidade de dados do software deveria ser representada por um rótulo de entidade de dados.

4. **Conectores de inclusão usados indevidamente:** embora os motivos 1 e 2 tenham aparecido com maior frequência dentre as causas da variação do resultado de medição calculado por APF-APF, o motivo 5 é o que apresentou maior o número de pontos de função relacionados desconsiderados e, portanto, é o motivo com maior potencial para ocasionar

variação nas contagens realizadas por APF-APF. O uso indevido do conector de inclusão foi observado para os respondentes 3, 4 e 5. O respondente 3 conectou via conector de inclusão, por exemplo, as funções elementares “Listar Local” e “Alterar Empregado”; e “Listar Local” e “Inserir Empregado” (ver figura 23 do apêndice F). A função “Listar Local” representava uma lista *dropdown* exibida durante as funções “Incluir Empregado” e “Alterar Empregado”. Na especificação de software fornecida, era descrito o requisito funcional do usuário: “*O sistema deve apresentar listas com todos os locais... cadastrados*”. Por definição da abordagem APF-APF, o conector de inclusão deveria ser usado entre tais funções elementares se: a função “Listar Local” (posterior), quando executada sozinha, não atendesse aos requisitos funcionais do usuário se a função “Incluir Empregado” (anterior) não fosse executada primeiro”. Nesse contexto, apenas a exibição da lista de empregados era suficiente para atender aos requisitos funcionais do usuário e, portanto, não deveria ser conectada a quaisquer outras funções elementares via conector de inclusão. Desse modo, supõe-se que a interface do software de Recursos Humanos descritos no estudo de caso tenha confundido os participantes que, ao observarem que não era possível acessar a lista de locais sem acessar a funcionalidade Incluir/Alterar Empregado conectaram indevidamente ambas as funcionalidades.

Portanto, quando se leva em consideração os motivos pelos quais os coeficientes de variação de confiabilidade e validade obtidos após o experimento são piores do que os coeficientes pretendidos, verifica-se que tal variação foi ocasionada pela ausência de informações do software modeladas na Árvore de pontos de função. Considerando que esse artefato tenha sido elaborado para concentrar todas as informações necessárias para a contagem, é esperado que a não inclusão de quaisquer informações tenha impacto nos resultados de medição gerados via APF-APF. A ausência de informações relevantes à APF-APF foi observada com menor frequência para o segundo grupo, que incluíram grande parte das informações requeridas para APF-APF, gerando resultados de medição com coeficiente de variação baixo, e conseqüentemente grau de validade e confiabilidade elevados.

A(s) razão(ões) para a ausência de informações na Árvore de pontos de função não foi(ram) identificada(s), podendo ter sido ocasionada pelas influências de variáveis durante o experimento, que ocasionaram ameaças à sua validade descritas na seção 6.2.2. Diante desse contexto, considera-se necessário adicionar um dispositivo, à abordagem APF-APF, para assegurar que todas as informações necessárias à contagem de pontos de função sejam informadas, de modo que seja possível assegurar que os resultados de medição gerados apresentem maior grau de confiabilidade e validade. Sugere-se que novos experimentos

devam ser realizados para confirmar (ou não) se a abordagem proposta é suficiente para aumentar a validade e confiabilidade, com maior eficiência, da contagem de pontos de função e para identificar os motivos pelos quais foram obtidos resultados aquém do esperado.

Quando o tempo utilizado pelos participantes é analisado, nota-se que o segundo grupo de respondentes levou, em média, 47% mais tempo do que o primeiro grupo para obter o tamanho em pontos de função do software de recursos humanos. Se por um lado há a possibilidade dessa eficiência ter sido ocasionada em razão do protótipo de ferramenta desenvolvido, por outro é possível considerar que a variação no tempo possa ter sido influenciada porque o segundo grupo usou mais tempo para incluir as informações requeridas para a Árvore de pontos de função. Deste modo, não é possível concluir que o ganho na eficiência tenha sido ocasionado exclusivamente pelo protótipo de ferramenta desenvolvido, mas também por uma maior quantidade de informações inseridas pelos respondentes que usaram a abordagem de APF-APF manual.

Quando os três objetivos específicos propostos nessa dissertação de mestrado são levados em consideração, baseados exclusivamente nos resultados obtidos em experimento, nota-se que há indícios de que a abordagem APF-APF é mais efetiva em assegurar maior produtividade na contagem de pontos de função com a utilização do protótipo de ferramenta do que assegurar maior confiabilidade e validade dos resultados de medição obtidos.

Quando os resultados obtidos com APF-APF são comparados com os reportados pelas abordagens propostas para mitigar o problema de regras que requerem interpretação individual (analisadas na seção 3.2), nota-se que, embora tenha apresentado resultados inferiores em relação à validade dos resultados de medição gerados, APF-APF apresenta cobertura de 100% do total de passos de APF necessários para a contagem de pontos de função. O fato de APF-APF apresentar total conformidade com os procedimentos previstos em APF e baixa validade (mediante resultados obtidos em experimento) reforça a necessidade de novos experimentos e/ou a inclusão de um dispositivo à abordagem que garanta que todas as informações requeridas para a Árvore de pontos de função sejam informadas. Os resultados comparativos entre as abordagens, podem ser consultados na tabela 10.

6.3 Considerações finais

Este capítulo destinou-se a apresentar o protótipo desenvolvido para auxiliar no processo de medição do software com APF-APF e os resultados obtidos após a execução dos experimentos para análise da abordagem proposta. Foram apresentadas as diferentes fases do

projeto de desenvolvimento do protótipo da ferramenta, incluindo o levantamento de requisitos, seu projeto arquitetural e a sua interface com a descrição das suas principais funcionalidades.

Tabela 10 – Comparação dos resultados obtidos com APF-APF em relação aos resultados reportados por abordagens similares¹⁴

	A	B	C	D	E	F	G	H
Cobertura das regras de APF (%)	16	30	55	14	32	55	42	18
Percentual de variação dos resultados gerados comparados ao tamanho oficial calculado por especialistas (%) ¹⁵	-	-	-	31,25	-	-	-	6,15
Quantidade de contagens realizadas com a abordagem proposta	-	-	-	01	-	-	-	01
	I	J	K	L	M	N	O	APF-APF
Cobertura das regras de APF (%)	18	36	41	14	5	52	41	100
Percentual de variação dos resultados gerados comparados ao tamanho oficial calculado por especialistas (%)	6,66	2,35	4,08	5,01	-	15,91	-	17,61
Quantidade de contagens realizadas com a abordagem proposta	01	01	5	7	-	2	-	11

Fonte: Marcos Freitas Junior, 2015

Em relação ao experimento de avaliação realizado, apesar dos resultados gerais obtidos indicarem coeficiente de variação de 10,72% em relação à confiabilidade e 17,61% em relação à validade, considera-se que a abordagem APF-APF mostrou potencial para que resultados positivos possam ser obtidos por meio da realização de um experimento com maior controle em relação aos respondentes e as variáveis externas. Quando analisados de modo mais específico, nota-se que os coeficientes de variação de 1,15% em relação à confiabilidade e 1,91% em relação à validade quando realizados por Analistas de requisitos.

Os cinco motivos para a variação do coeficiente de confiabilidade e validade dos resultados de medição obtidos por APF-APF observados relacionam-se com a ausência de informações requeridas para a Árvore de pontos de função. Supõe-se que sem a ausência das

¹⁴ A legenda das siglas designadas para as abordagens propostas pode ser consultada nos quadros 13 e 14 do apêndice B.

informações requeridas para a contagem de pontos de função, as variações observadas diminuíam significativamente em relação ao observado nesse experimento.

A ausência das informações na Árvore de pontos de função tem efeito semelhante ao observado para as abordagens propostas que simplificam os procedimentos necessários para a contagem de pontos de função. Isso ocorre porque, sem as informações necessárias para a contagem de pontos de função, os passos para o cálculo são simplificados já que não são analisadas todas as condições necessárias para a contagem devido a ausência de informações. Como consequência, se faz necessário o desenvolvimento de um dispositivo que assegure que todas as informações necessárias serão modeladas na Árvore de pontos de função.

Por fim, verificou-se que o uso de um protótipo de ferramenta aumenta em até 47% a eficiência na contagem de pontos de função quando comparado com APF-APF manual. Embora esse resultado possa ser consequência de modelagens da Árvore de pontos de função elaboradas com maior rapidez devido à ausência de informações requeridas, considera-se que o resultado é satisfatório e indica possibilidade de maior eficiência na contagem de pontos de função.

7. Conclusão

Sendo usada como referência para a derivação de medidas derivadas, APF tem desempenhado um papel importante em organizações privadas e públicas, sendo imprescindível a obtenção de resultados confiáveis e válidos com a aplicação dessa medida. Entretanto, são relatados por diferentes autores, problemas em relação à confiabilidade dos resultados de medição obtidos com APF.

Desse modo, esse trabalho teve como objetivo a definição de uma abordagem, denominada APF-APF, que prevê que a contagem de pontos de função seja realizada com maior confiabilidade e eficiência, mantendo-se a validade com APF original. Como consequência, é proposto que o artefato Árvore de pontos de função englobe todas as informações necessárias para a contagem. Também foi desenvolvido um protótipo de ferramenta para automatizar APF-APF.

Os resultados obtidos após os experimentos indicaram que, em média, APF-APF produz resultados com variação de 10,72% em relação à confiabilidade e 17,61% em relação à sua validade comparada com APF original. Verificou-se que o principal motivo para obtenção de resultados aquém do pretendido estava relacionada à ausência de informações, requeridas para a Árvore de pontos de função, não inseridas pelos respondentes durante modelagem da Árvore no experimento. Não foi identificado nenhum problema específico em relação às regras definidas para APF-APF, de modo que supõe-se que a ausência destas informações tenha sido ocasionada pelas condições em que o experimento foi realizado.

Também há indícios de o protótipo de ferramenta desenvolvido para APF-APF melhora em 47% a produtividade para contagem de pontos de função. Embora os resultados em relação à validação e confiabilidade da abordagem tenham sido negativos, considera-se que APF-APF apresenta potencial para que melhores resultados possam ser obtidos se avaliada em novos experimentos. Quando os resultados obtidos são comparados os resultados dos trabalhos relacionados, verifica-se que APF-APF apresenta uma cobertura de 100% em relação à APF original, embora os resultados em relação à validade dessa abordagem tenham sido inferiores.

Embora o presente estudo tenha realizado experimento com respondentes de duas organizações para verificar a validade das hipóteses estabelecidas, recomenda-se cautela na generalização dos resultados deste experimento, já que: (i) a quantidade de respondentes utilizada é pequena, considerando os profissionais de uma organização e (ii) o uso de uma

única especificação funcional de um software de RH é pequena em relação a população de softwares com natureza e características distintas como observado em uma organização.

As principais contribuições científicas e para o mercado, bem como as recomendações para trabalhos futuros são descritos a seguir.

7.1 Contribuição científica

Após revisão sistemática da literatura, identificaram-se 15 abordagens propostas para aumentar a confiabilidade da contagem de pontos de função. A maior parte das abordagens adotam como principal estratégia a definição de um mapeamento entre componentes dos artefatos produzidos no ciclo de vida de software e componentes de APF. Entretanto, como os artefatos foram desenvolvidos para armazenar informações específicas e necessárias a contagem de pontos de função, os mapeamentos realizados pelos pesquisadores são incompletos, o que compromete a validade das abordagens propostas na literatura quando comparados com APF original. Desse modo, a estratégia adotada pelos trabalhos relacionados acaba surtindo, em geral, efeito negativo já que são propostos distintos mapeamentos que produzem resultados não confiáveis entre si e válidos perante APF original.

Diante deste contexto, esse trabalho além de identificar e relatar os problemas em relação à estratégia adotada na literatura para assegurar maior confiabilidade nos resultados de medição gerados, também propõe uma nova estratégia para calcular o tamanho em pontos de função do software, por meio de um novo artefato – denominado Árvore de pontos de função - que unifica todas as informações necessárias à contagem de pontos de função.

A adoção de um único artefato como base para o estabelecimento de mapeamentos possibilitaria que abordagens criadas por trabalhos relacionados a partir deste artefato, possam gerar resultados com maior confiabilidade entre si, e possam ser comparados.

7.2 Contribuição para o mercado

APF-APF foi concebida para garantir maior confiabilidade e eficiência à contagem de pontos de função, mantendo a validade com APF original. Desse modo, a abordagem pode ser utilizada para apoiar a realização de contagem de pontos de função dos softwares, em organizações da esfera privada ou pública, servindo como apoio para a geração de medidas indiretas mais confiáveis. Além disso, APF-APF pode ser aplicada nas organizações com

finalidade distintas, da que inicialmente foram planejadas. Dentre as possíveis aplicações alternativas destaca-se a possibilidade de APF-APF ser utilizada:

→ Em treinamentos de APF para estudantes e/ou profissionais atuantes no mercado de trabalho que tenham interesse nessa medida. Por conter todas as regras de APF, essa abordagem pode auxiliar possíveis interessados em treinar, de modo simulatório, a contagem de pontos de função. Baseado em uma determinada especificação funcional, por exemplo, um profissional poderia efetuar a contagem manual de pontos de função e paralelamente modelar a Árvore de pontos de função para obter o tamanho funcional segundo APF-APF. A comparação de ambos os valores poderia ser um indicativo se o tamanho calculado manualmente é válido em relação à APF. Ainda nesse contexto, a Árvore de pontos de função poderia ser utilizada de modo didático para exemplificar as informações necessárias para a contagem de pontos de função e como essas são utilizadas para que se possa obter o tamanho funcional de um software. Também possibilitaria estimular com que os alunos e/ou profissionais, baseados nos conceitos de APF, aplicassem o conhecimento de modo distinto, corroborando possivelmente para um aprendizado mais efetivo.

→ Como apoio para elaboração de propostas de desenvolvimento de software por pessoas com pouca ou nenhuma experiência na contagem de pontos de função. Embora a abordagem APF-APF mantenha a validade com APF, e conseqüentemente, também requeira uma grande quantidade de informações para o cálculo de pontos de função de um software (ver seção 3.1), uma modelagem “inicial” da Árvore de pontos de função poderia ser realizada com “aproximação” de algumas informações não disponíveis no início de ciclo de vida do software. Desse modo, seria possível estimar o tamanho de pontos de função e, conseqüentemente, apoiar a derivação de medidas indiretas como custo e/ou esforço para desenvolvimento.

→ Como apoio ao processo de documentação da contagem de pontos de função. Especialistas poderiam usar a Árvore de pontos de função para documentar, de modo sistemático e padronizado, todas as informações (disponíveis, presumidas e/ou levantadas diretamente com os responsáveis) a respeito de um software sendo contado. Como consequência, seria possível manter a rastreabilidade entre o tamanho funcional calculado por um especialista e quais informações foram usadas para que se chegasse a um determinado tamanho funcional. Isso poderia auxiliar inclusive organizações que adotem um modelo de desenvolvimento que exija menor detalhamento a respeito das funcionalidades de um software (necessário à contagem de pontos de função) e que queira manter a rastreabilidade das informações específicas usadas para a contagem de pontos de função. Embora seja um dos

procedimentos da contagem de pontos de função documentar e relatar as informações usadas para a contagem de pontos de função, não há nenhuma definição do IFPUG como isso deva ser realizado ou quais informações devam ser reportadas. Sendo assim, a Árvore de pontos de função poderia ser usada como modelo de documentação das informações de um software que influenciaram a contagem de pontos de função.

7.3 Trabalhos futuros

Trabalhos futuros podem: *(i)* Incorporar um dispositivo que assegure que todas as informações necessárias serão modeladas na Árvore de pontos de função; *(ii)* Melhorar a escalabilidade do artefato Árvore de pontos de função, para que a visualização dos elementos da Árvore de pontos de função não seja prejudicada para softwares com grande quantidade de funcionalidades e entidades e *(iii)* Realizar novos experimentos com APF-APF para confirmar as hipóteses estabelecidas após coleta dos resultados de que: a variação dos valores de confiabilidade e validade dos resultados é ocasionada em virtude da ausência de informações requeridas em APF-APF.

Referências

- ABDULLAH, N. A. S. et al. A. Software security characteristics for function point analysis. 2009 IEEE International Conference on Industrial Engineering and Engineering Management. **Anais...** . p.394–397, Set. 2009. Hong Kong: IEEE. Disponível em: <<http://www.scopus.com/inward/record.url?eid=2-s2.0-77949511584&partnerID=tZOtx3y1>>. Acesso em: 29 Jul. 2014.
- ABRAHÃO, S.; INSFRÁN, E. A Metamodeling Approach to Estimate Software Size from Requirements Specifications. 2008 34th Euromicro Conference Software Engineering and Advanced Applications. **Anais...** . p.465–475, Set. 2008. Parma: IEEE. Disponível em: <<http://www.scopus.com/inward/record.url?eid=2-s2.0-60349116222&partnerID=tZOtx3y1>>. Acesso em: 29 Jul. 2014.
- ABRAHÃO, S.; GÓMEZ, J.; INSFRÁN, E. Validating a size measure for effort estimation in model-driven Web development. **Information Sciences**, v. 180, n. 20, p. 3932–3954, Out 2010. Disponível em: <<http://www.scopus.com/inward/record.url?eid=2-s2.0-77958152177&partnerID=tZOtx3y1>>. Acesso em: 16/7/2014.
- ABRAHÃO, S. et al. A model-driven measurement procedure for sizing web applications: Design, automation and validation. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). **Anais...** . v. 4735 LNCS, p.467–481, 2007. Disponível em: <<http://www.scopus.com/inward/record.url?eid=2-s2.0-38049008392&partnerID=tZOtx3y1>>. .
- ABUALKISHIK, A. Z. et al. An exploratory study on the accuracy of FPA to COSMIC measurement method conversion types. **Information and Software Technology**, v. 54, n. 11, p. 1250–1264, Nov. 2012. Disponível em: <<http://www.scopus.com/inward/record.url?eid=2-s2.0-84864773384&partnerID=tZOtx3y1>>. Acesso em: 29 Jul. 2014.
- ADAMO, D. A.; FABRIZI, S.; VERGATI, M. G. A light functional dimension estimation model for software maintenance. Proceedings – 2007 IEEE Conference on Exploring Quantifiable IT Yields, EQUITY 2007. **Anais...** . p.73–78, 2007. Amsterdam. Disponível em: <<http://www.scopus.com/inward/record.url?eid=2-s2.0-70450184543&partnerID=tZOtx3y1>>.
- ADEM, N. A. Z.; KASIRUN, Z. M. Automating Function Points analysis based on functional and non functional requirements text. 2010 The 2nd International Conference on Computer and Automation Engineering. **Anais...** . v. 5, p.664–669, 2010. Singapore: IEEE. Disponível em: <<http://www.scopus.com/inward/record.url?eid=2-s2.0-77952607858&partnerID=tZOtx3y1>>. Acesso em: 29 Jul. 2014.
- ALLEN, M. J.; YEN, W. M. **Introduction to Measurement Theory**. 1 ed. Illinois: Waveland Press Inc, 2001. 310 p.
- ARNUPHAPTRAIRONG, T. Early stage software effort estimation using function point analysis: Empirical evidence. **Lecture Notes in Engineering and Computer Science**, v. 2,

n. 3, p. 730–735, Mar. 2013. Hong Kong. Disponível em:
<<http://www.scopus.com/inward/record.url?eid=2-s2.0-84880079456&partnerID=tZOtx3y1>>.

ARUMUGAM, C.; BABU, C. Developmental Size Estimation For Object-Oriented Software Based On Analysis Model. **International Journal of Software Engineering and Knowledge Engineering**, [S.l.], v. 23, n. 3, p. 289-308, Abr. 2013. Disponível em:
<<http://www.scopus.com/inward/record.url?eid=2-s2.0-84880537856&partnerID=tZOtx3y1>>. Acesso em: 29 Jul. 2014.

BARCELLOS, M.P., 2009, *Uma estratégia para medição de software e avaliação de bases de medidas para controle estatístico de processos de software em organizações de alta maturidade*, Tese de D.Sc., Programa de Engenharia de Sistemas e Computação, COPPE/UFRJ, Rio de Janeiro, Brasil

BHARADWAJ, A. K.; NAIR, T. R. G. Mapping General System Characteristics to Non-Functional Requirements. 2009 IEEE International Advance Computing Conference. **Anais...** . p.1634–1638, 2009. Patiala: IEEE. Disponível em:
<<http://www.scopus.com/inward/record.url?eid=2-s2.0-66249108274&partnerID=tZOtx3y1>>. Acesso em: 29 Jul. 2014.

BRASIL. **Instrução Normativa MP/SLTI N° 4**, de 12 de novembro de 2010a. Dispõe sobre o processo de contratação de Soluções de Tecnologia da Informação pelos órgãos integrantes do Sistema de Administração dos Recursos de Informação e Informática (SISP) do Poder Executivo Federal. Diário Oficial da União, n. 218, 16 de novembro de 2010, Seção 1.

_____. **Portaria SLTI/MP N° 31**, de 29 de novembro de 2010b, Dispõe sobre recomendações técnicas para a utilização da métrica Análise de Ponto de Função no âmbito da Administração Pública Federal direta, autárquica e fundacional e dá outras providências. Disponível em < <http://www.comprasgovernamentais.gov.br/paginas/portarias/portaria-no-31-de-29-de-novembro-de-2010>>. Acesso em: 08 Dez. 2014.

_____. Ministério da Ciência e Tecnologia. Secretaria de Política de Informática. **Pesquisa de Qualidade no Setor de Software Brasileiro 2009**. Brasília, 2010c.

BIOLCHINI, J.; MIAN, P. G.; NATALI, A. C. C.; COPPE/UFRJ; Systematic Review in Software Engineering. RT-ES 679/05. Rio de Janeiro, RJ, Brasil, 2005.

CALAZANS, A. T.; DE OLIVEIRA, K. M.; DOS SANTOS, R. R. Adapting function point analysis to estimate data mart size. 10th International Symposium on Software Metrics, 2004. Proceedings. **Anais...** . p.300–311, 2004. [S.l.]: IEEE. Disponível em:
<<http://www.scopus.com/inward/record.url?eid=2-s2.0-14844323674&partnerID=tZOtx3y1>>. Acesso em: 29 Jul. 2014.

CANTONE, G.; PACE, D.; CALAVARO, G. Applying function point to unified modeling language: conversion model and pilot study. 10th International Symposium on Software Metrics, 2004. Proceedings. **Anais...** . p.280–291, 2004. IEEE. Disponível em:
<<http://www.scopus.com/inward/record.url?eid=2-s2.0->

14844307654&partnerID=tZOtx3y1>. Acesso em: 29/7/2014.

CHAMUNDESWARI, A.; BABU, C. An Extended Function Point Approach for Size Estimation of Object-Oriented Software. Proceedings of the 3rd International Conference on Software and Data Technologies. **Anais...** . p.139–145, 2010. India: IEEE. Disponível em: <<http://www.scopus.com/inward/record.url?eid=2-s2.0-77952607858&partnerID=tZOtx3y1>>. Acesso em: 29 Jul. 2014.

CHEN, Q. et al. Study of Function Point Analysis based on fuzzy-interpolation. **Journal of Computational Information Systems**, Hangzhou, v. 6, n. 5, p. 1369–1375, Mai. 2010. Disponível em: <<http://www.scopus.com/inward/record.url?eid=2-s2.0-77956945842&partnerID=tZOtx3y1>>. Acesso em: 29 Jul. 2014.

CHOI, S.; PARK, S.; SUGUMARAN, V. Function Point Extraction Method from Goal and Scenario Based Requirements Text. In: KOP, C.; FLIEDL, G.; MAYR, H.; MÉTAIS, E. (Eds.) Natural Language Processing and Information Systems Lecture Notes in Computer Science. **Anais...** . p.12–24, 2006. Springer Berlin Heidelberg. Disponível em: <http://link.springer.com/chapter/10.1007%2F11765448_2>. Acesso em: 29 Jul. 2014.

DE MARCO, L. et al. Functional versus design measures for model-driven Web applications: A case study in the context of Web effort estimation. 2012 3rd International Workshop on Emerging Trends in Software Metrics. **Anais...** . p.21–27, 2012. Zurich: IEEE. Disponível em: <<http://www.scopus.com/inward/record.url?eid=2-s2.0-84864151910&partnerID=tZOtx3y1>>. Acesso em: 29 Jul. 2014.

DEL BIANCO, V.; LAVAZZA, L. An Empirical Assessment of Function Point-Like Object-Oriented Metrics. 11th IEEE International Software Metrics Symposium. **Anais...** . v. 2005, p.40–40, 2005. Como: IEEE. Disponível em: <<http://www.scopus.com/inward/record.url?eid=2-s2.0-33749044596&partnerID=tZOtx3y1>>. Acesso em: 29 Jul. 2014.

DEL BIANCO, V.; LAVAZZA, L.; MORASCA, S. A Proposal for Simplified Model-Based Cost Estimation Models. Lecture Notes in Computer Science. **Anais...** , v. 7343, p.59–73, 2012. Berlin: Springer Berlin Heidelberg. Disponível em: <<http://www.scopus.com/inward/record.url?eid=2-s2.0-84862205307&partnerID=tZOtx3y1>>. Acesso em: 29 Jul. 2014.

EDAGAWA, T.; AKAIKE, T.; HIGO, Y.; et al. Function point measurement from Web application source code based on screen transitions and database accesses. **Journal of Systems and Software**, v. 84, n. 6, p. 976–984, 2011. Disponível em: <<http://www.scopus.com/inward/record.url?eid=2-s2.0-79953709101&partnerID=tZOtx3y1>>. Acesso em: 13/2/2015.

FENTON, N.; BIEMAN, J. **Software Metrics: A Rigorous & Practical Approach**. 3 ed. Boston: Pws Publishing, 2014. 656 p.

FERREIRA, W. G.; MARQUES-NETO, H. T. Estimating the size of data mart projects. Proceedings of the 28th Annual ACM Symposium on Applied Computing – SAC '13. **Anais...** . p.1147, 2013. Coimbra: ACM Press. Disponível em: <<http://www.scopus.com/inward/record.url?eid=2-s2.0->

84877944932&partnerID=tZOtx3y1>. Acesso em: 29 Jul. 2014.

FITZGERALD, J.; LARSEN, P. G. **Modelling Systems: Practical Tools and Techniques in Software Development**. 2 ed. New York: Cambridge University Press, 2009. 304p.

FRATERNALI, P.; TISI, M.; BONGIO, A. Automating function point analysis with model driven development. Proceedings of the 2006 conference of the Center for Advanced Studies on Collaborative research – CASCON '06. **Anais...** . p.18, 2006. Toronto: ACM Press. Disponível em: <<http://www.scopus.com/inward/record.url?eid=2-s2.0-36949039742&partnerID=tZOtx3y1>>. Acesso em: 29 Jul. 2014.

FU, Y. F. et al. A Software Size Estimation Method Based on Improved FPA. 2010 Second World Congress on Software Engineering. **Anais...** . v. 2, p.228–233, 2010. Wuhan: IEEE. Disponível em: <<http://www.scopus.com/inward/record.url?eid=2-s2.0-79952949291&partnerID=tZOtx3y1>>. Acesso em: 29 Jul. 2014.

GARCÍA, F. et al. Towards a consistent terminology for software measurement. **Information and Software Technology**, v. 48, n. 8, p. 631–644, 2006.

GARRIGÓS, I.; CASTELEYN, S.; GÓMEZ, J. A Structured Approach to Personalize Websites Using the OO-H Personalization Framework. Web Technologies Research and Development – APWeb 2005. **Anais...** . v. 3399, p.695–706, 2005. Shanghai: Springer Berlin Heidelberg. Disponível em: <http://link.springer.com/chapter/10.1007%2F978-3-540-31849-1_67>. .

GERLEC, C.; KRAJNC, A.; GOLJAT, U. A method for evaluating the actual functional size in model driven software development. 2010 2nd International Conference on Software Technology and Engineering. **Anais...** . v. 1, 2010. San Juan: IEEE. Disponível em: <<http://www.scopus.com/inward/record.url?eid=2-s2.0-78649993675&partnerID=tZOtx3y1>>. Acesso em: 29 Jul. 2014.

HARPUR, V.; KAINDL, H.; KRAMER, S. Extending Function Point Analysis of Object-Oriented Requirements Specifications. 11th IEEE International Software Metrics Symposium. **Anais...** . v. 2005, p.10–39, 2005. Como: IEEE. Disponível em: <<http://www.scopus.com/inward/record.url?eid=2-s2.0-33749069898&partnerID=tZOtx3y1>>. Acesso em: 29 Jul. 2014.

HAUG, M.; OLSEN, E. W.; BERGMAN, L. **Software Process Improvement: Metrics, Measurement, and Process Modelling**. 1 ed. Berlim: Springer, 2001. 391 p.

HEVNER, A. R.; MARCH, S. T.; PARK, J.; RAM, S. Design Science in Information System Research. **MIS Quarterly**, vol. 28, n. 1, p. 75-105, 2004.

HUANG, H.; REN, A.; YANG, Z. Conversion of Function Points between NESMA and IFPUG method. Proceedings – 2012 6th International Conference on New Trends in Information Science, Service Science and Data Mining. **Anais...** . p.89–93, 2012. Taipei. Disponível em: <<http://www.scopus.com/inward/record.url?eid=2-s2.0-84881018340&partnerID=tZOtx3y1>>.

IFPUG. Manual de Práticas de Contagem de Pontos de Função. V.4.3.1. Trad. SILVEIRA,

M. et. al. São Paulo: IFPUG – International Function Point Users Group. 2010.

INSFRÁN, E.; PASTOR, O.; WIERINGA, R. Requirements Engineering-Based Conceptual Modeling. **Requirements Engineering**, v. 7, n. 2, p. 61–72, 2002. Disponível em: <<http://link.springer.com/article/10.1007%2Fs007660200005>>.

IORIO, T. IFPUG Function Point Analysis in an UML Framework. In: **Software Metrics European Forum**, Jan. 2004, Roma. Disponível em: <<http://www.dpo.it/resources/papers/2004-smef-iorio-en.pdf>>. Acesso em: 29 set 2014. 10 p.

ISO/IEC/IEEE 24765: 2010. **International Standard: Systems and software engineering — Vocabulary**. 418 p.

JAMES, R.; IVAR, J.; GRADY, B. **The unified modeling language reference manual**. 2 ed. Addison-Wesley Professional, 2004.

JILANI, A. A. A.; NADEEM, A.; KIM, T. H.; CHO, E. S. Formal representations of the data flow diagram: A survey. Proceedings of the 2008 Advanced Software Engineering and its Applications, ASE 2008. **Anais...** . p.153–158, 2008. Disponível em: <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4721332&tag=1> . .

KAMPSTRA, P.; VERHOEF, C. Reliability of Function Point Counts. **Communications of the ACM**, Amsterdam, p. 85-97, [2009]. Disponível em: <<http://www.cs.vu.nl/~x/rofpc/rofpc.pdf>>.

KAN, S. H. Metrics and Models in Software Quality Engineering. 2 ed. Boston: Addison Wesley, 2002. 560 p.

KEMERER, F. C. Reliability of Function Points Measurement: A field experiment. **Communications of the ACM**, [s.l.], v. 36, n. 2, p. 85-97, Fev. 1993. Disponível em: <<http://dl.acm.org/citation.cfm?id=151230>> .

KIM, J.; PARK, S.; SUGUMARAN, V. **Natural Language Processing and Information Systems**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004.

KITCHENHAM, B; Guidelines for Performing Systematic Literature Reviews in Software Engineering, Technical Report EBSE-2007-01, Keele University, United Kingdom, 2007.

_____. Department of Computer Science, Keele University (Hrsg.): Procedures for Performing Systematic Reviews, 2004.

KLUSENER, S. Source code based function point analysis for enhancement projects. International Conference on Software Maintenance, 2003. ICSM 2003. Proceedings. **Anais...** . p.373–376, 2003. IEEE Comput. Soc. Disponível em: <<http://www.scopus.com/inward/record.url?eid=2-s2.0-77956618492&partnerID=tZOtx3y1>>. Acesso em: 29/7/2014.

KRALJ, T. et al. Improved standard FPA method—resolving problems with upper boundaries in the rating complexity process. **Journal of Systems and Software**, v. 77, n. 2,

p. 81–90, Ago. 2005. Disponível em: <<http://www.scopus.com/inward/record.url?eid=2-s2.0-18144375951&partnerID=tZOtx3y1>>. Acesso em: 29 Jul. 2014.

KUSUMOTO, S. et al. Function point measurement from Java programs. Proceedings – International Conference on Software Engineering. **Anais...** . p.576–582, 2002. Orlando. Disponível em: <<http://www.scopus.com/inward/record.url?eid=2-s2.0-0036040435&partnerID=tZOtx3y1>>. Acesso em: 29 Jul. 2014.

KUIJPERS, C. Automated FPA (eFPA) in SAP Environment - Visions and Experiences of Automated Function Point Analysis. **2014 Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement**, p. 72–78, 2014. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7000079>>.

LAMMA, E.; MELLO, P.; RIGUZZI, F. A system for measuring function points from an ER-DFD specification. **Computer Journal**, v. 47, n. 3, p. 358–372, 2004. Disponível em: <<http://www.scopus.com/inward/record.url?eid=2-s2.0-2442540031&partnerID=tZOtx3y1>>. Acesso em: 22 Abr. 2015.

LAVAZZA, L. Automated Function Points: Critical Evaluation and Discussion. **2015 IEEE/ACM 6th International Workshop on Emerging Trends in Software Metrics**, , n. x, p. 35–43, 2015. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7181589>>.

LAVAZZA, L.; DEL BIANCO, V. Functional size measurement based on problem frames. Proceedings of the 3rd international workshop on Applications and advances of problem frames – IWAAPF '08. **Anais...** . p.44, 2008. Leipzig: ACM Press. Disponível em: <<http://www.scopus.com/inward/record.url?eid=2-s2.0-57049144750&partnerID=tZOtx3y1>>. Acesso em: 29 Jul. 2014.

LAVAZZA, L.; GARAVAGLIA, C. Using function points to measure and estimate real-time and embedded software: Experiences and guidelines. 3rd International Symposium on Empirical Software Engineering and Measurement. **Anais...** . p.100–110, 2009. Lake Buena Vista: IEEE. Disponível em: <<http://www.scopus.com/inward/record.url?eid=2-s2.0-72449179859&partnerID=tZOtx3y1>>. Acesso em: 29 Jul. 2014.

LAVAZZA, L. A.; DEL BIANCO, V.; GARAVAGLIA, C. Model-based functional size measurement. Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement. **Anais...** . p.100, 2008. Kaiserslautern: ACM Press. Disponível em: <<http://www.scopus.com/inward/record.url?eid=2-s2.0-62949189583&partnerID=tZOtx3y1>>. Acesso em: 29 Jul. 2014.

LAVAZZA, L.; MORASCA, S.; ROBILOLO, G. Towards a simplified definition of Function Points. **Information and Software Technology**, v. 55, n. 10, p. 1796–1809, Out. 2013. Disponível em: <<http://www.scopus.com/inward/record.url?eid=2-s2.0-84880809706&partnerID=tZOtx3y1>>. Acesso em: 29 Jul. 2014.

LOW, G. C.; JEFFERY, D. R. Function points in the estimation and evaluation of the software process. **IEEE Transactions on Software Engineering**, v. 16, n. 1, p. 64–71, 1990. Disponível em:

<<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=44364>>. Acesso em: 15/1/2015.

MATIJEVIĆ, T.; OGNJANOVIĆ, I.; ŠENDELJ, R. Enhancement of software projects' Function Point Analysis based on conditional non-functional judgments. 2012 Mediterranean Conference on Embedded Computing. **Anais...** . p.283–287, 2012. Bar. Disponível em: <<http://www.scopus.com/inward/record.url?eid=2-s2.0-84866999400&partnerID=tZOtx3y1>>. Acesso em: 29 Jul. 2014.

MARZOUGH, F. et al. A decision model for estimating the effort of software projects using Bayesian theory. 2010 2nd International Conference on Software Technology and Engineering. **Anais...** . v. 2, 2010. San Juan: IEEE. Disponível em: <<http://www.scopus.com/inward/record.url?eid=2-s2.0-78649993673&partnerID=tZOtx3y1>>. Acesso em: 29 Jul. 2014.

MICHAELIS. Dicionário Prático: Língua Portuguesa. 1 ed. Melhoramentos, 2011. 976 p.

MILLER, Rick. **C# for artists: he Art, Philosophy, And Science Of Object**. Virginia: Pulp Free Press, 2008.

MIYAWAKI, T.; IJIMA, J.; HO, S. Measuring function points from VDM-SL specifications. 2008 International Conference on Service Systems and Service Management. **Anais...** . p.1–6, 2008. Melbourne: IEEE. Disponível em: <<http://www.scopus.com/inward/record.url?eid=2-s2.0-52249113531&partnerID=tZOtx3y1>>. Acesso em: 29 Jul. 2014.

National Measurement System. Measurements in daily life. Londres: National Measurement Office, [2009]. 9 slides, color. Disponível em: <<http://www.npl.co.uk/upload/pdf/measurement-matters.pdf>>. Acesso em: 20 nov. 2014.

POW-SANG, J. A.; VILLANUEVA, D.; FLORES, L.; RUSU, C. A Conversion Model and a Tool to Identify Function Point Logic Files Using UML Analysis Class Diagrams. 2013 Joint Conference of the 23rd International Workshop on Software Measurement and the 8th International Conference on Software Process and Product Measurement. **Anais...** . p.126–134, 2013. IEEE. Disponível em: <<http://www.scopus.com/inward/record.url?eid=2-s2.0-84893569614&partnerID=tZOtx3y1>>. Acesso em: 1/2/2015.

PRESSMAN, R. S. Engenharia de Software: Uma Abordagem Profissional. 7. ed. [s.l]: Bookman, 2011. 752 p.

RAO, K. K.; RAJU, G. S. Error correction in function point estimation using soft computing technique. Proceedings of the International Conference on Advances in Computing and Artificial Intelligence. **Anais...** . p.194–198, 2011. Rajpura/Punjab: ACM Press. Disponível em: <<http://www.scopus.com/inward/record.url?eid=2-s2.0-80052318170&partnerID=tZOtx3y1>>. Acesso em: 29 Jul. 2014.

RAO, K. K. et al., .Measuring the Function Points from the Points of Relationships of UML. 2008 International Conference on Computer and Electrical Engineering. **Anais...** . p.748–752, 2008. Phuket: IEEE. Disponível em: <<http://www.scopus.com/inward/record.url?eid=2-s2.0->

62949152831&partnerID=tZOtx3y1>. Acesso em: 29 Jul. 2014.

STANDISH GROUP. CHAOS MANIFESTO 2013: Think Big, Act Small. 2013.

SISTEMA DE ADMINISTRAÇÃO DOS RECURSOS DE INFORMAÇÃO E INFORMÁTICA. – Portaria SLTI/MP Nº 31, de 29 novembro de 2010. *Document*, Brasília, n. 31, nov. 2010

TOKA, D.; TURETKEN, O. Accuracy of Contemporary Parametric Software Estimation Models: A Comparative Analysis. 2013 39th Euromicro Conference on Software Engineering and Advanced Applications. **Anais...** . p.313–316, 2013. Santander: IEEE. Disponível em: <<http://www.scopus.com/inward/record.url?eid=2-s2.0-84889043717&partnerID=tZOtx3y1>>. Acesso em: 20/11/2014.

TESCHE E VASCONCELOS. Cadastro de clientes. 2014a. **Galeria de Imagens** – CIAF Locadora. Disponível em: <<http://www.tvsistemas.com.br/screenshotsCiafLocadora.html>>. Acesso em: 12 dez. 2014.

_____. Cadastro de dependentes. 2014b. **Galeria de Imagens** – CIAF Locadora. Disponível em: <<http://www.tvsistemas.com.br/screenshotsCiafLocadora.html>>. Acesso em: 12 dez. 2014.

UEMURA, T.; KUSUMOTO, S.; INOUE, K. Function-point analysis using design specifications based on the Unified Modelling Language. **Journal of Software Maintenance and Evolution: Research and Practice**, v. 13, n. 4, p. 223–243, Ago. 2001. Disponível em: <<http://www.scopus.com/inward/record.url?eid=2-s2.0-0034821943&partnerID=tZOtx3y1>>. Acesso em: 29 Jul. 2014.

VASCONCELOS, A. Introdução a Métricas de Software. 2005. 57 slides, color. Disponível em <www.cin.ufpe.br/~if720/slides/introducao-a-metricas-de-software.ppt>. Último acesso em: 27 Abr. 2014.

VAZQUEZ, C. E.; SIMÕES, G. S.; ALBERT, R. M. Análise de Pontos de Função: Medição, Estimativas e Gerenciamento de Projetos de Software. 13 ed. São Paulo: Editora Erica, 2013. 272 p.

WIERINGA, R. **Design Methods for Reactive Systems**: Yourdon, Statemate, and the UML. San Francisco: Morgan Kaufmann, 2003. 500 p.

_____. **Requirements Targeting Software and Systems Engineering**. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998.

WOHLIN, C. et al. Experimentation in software engineering. Berlim: Springer, 2000. 236 p

WANG, Y. The Measurement Theory for Software Engineering. Canadian Conference on Electrical and Computer Engineering. **Anais...** . v. 2, p.1321–1324, 2003. Montreal: IEEE. Disponível em: <<http://www.scopus.com/inward/record.url?eid=2-s2.0-0141428011&partnerID=tZOtx3y1>>. .

XIA, W. et al. A new calibration for Function Point complexity weights. **Information and**

Software Technology, v. 50, n. 7-8, p. 670–683, Jun. 2008. Disponível em:
<<http://www.scopus.com/inward/record.url?eid=2-s2.0-42649134811&partnerID=tZOtx3y1>>. Acesso em: 29 Jul. 2014.

YA-FANG, F. et al. A Software Size Estimation Method Based on Improved FPA. 2010 **Second World Congress On Software Engineering**, Wuhan, v. 2, p.228-233, dec. 2010. Disponível em:
<http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=5718383&url=http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5718383>. Last accessed on: 05 may 2014.

Apêndice A – Especificação de requisitos do SISRH

A seguir, é apresentada a especificação de requisitos¹⁹ do Software de Recursos Humanos (SISRH), usado como base para elaboração da Árvore de Refinamento de Funções (ver figura 6) e da Árvore de pontos de função (ver figura 7). A especificação de requisitos, foi baseada em estudo de caso elaborado por IFPUG (2010). Deve-se ressaltar que essa especificação foi elaborada apenas com o propósito de servir como base para os exemplos e figuras apresentados nesse trabalho, não tendo sido aplicada em experimento com os respondentes.

➤ Objetivos deste documento

Descrever e especificar as necessidades do projeto, com o intuito de servir de agente facilitador no gerenciamento dos requisitos do projeto e no posterior desenvolvimento do SISRH.

➤ Escopo do produto

Uma organização, independente de seu produto fim, é um ambiente que precisa de uma gerência eficaz de informação de seus empregados, para prover um serviço com qualidade. Diante desse contexto, a missão do SISRH é integrar todas as plataformas de gestão de recursos humanos da organização, contribuindo para a potencialização do capital humano. O SISRH tem como principais funções o: *(i)* Gerenciamento de Empregados; *(ii)* Gerenciamento de Cargos e; *(iii)* Gerenciamento de Funções.

➤ Requisitos funcionais

→ Gerenciamento de empregados

RF1. O software deve permitir cadastrar um novo empregado com seus dados básicos (código; nome; CPF; sexo; estado civil; RG; nascimento; CEP; endereço; número; complemento; bairro; cidade; estado; telefones; e-mail; categoria; status (ativo/bloqueado);

¹⁹ Existem diversos formatos, sugeridos por distintos autores na literatura, para a especificação de requisitos de um software. Diante desse contexto, deve-se ressaltar que não foi adotado nenhum modelo específico para o desenvolvimento desta especificação de requisitos. O intuito não é descrever detalhadamente o SISRH, apenas descrever as informações necessárias para entendimento do exposto nessa dissertação de mestrado.

nome dependente; CPF dependente; RG dependente; telefone dependente, foto e idade) e de seus dependentes (nome, CPF, RG e telefone), assim como dados específicos de empregados assalariados (data do 13^a salário, período de férias) ou horistas (taxa por hora, número de horas).

RF2. O software deve permitir alterar todos os dados básicos do empregado e de seus dependentes, assim como dados específicos de empregados assalariados ou horistas.

RF3. O software deve permitir excluir um empregado que não esteja ocupando um cargo. Quando um empregado é excluído, todos os dados do empregado assalariados ou de empregados horistas, assim como de seus dependentes também devem ser excluídos.

RF4. O software deve permitir consultar todos os dados básicos do empregado e de seus dependentes, assim como todos os dados específicos de empregados assalariados ou horistas.

→ Gerenciamento de Cargos

RF5. O software deve permitir cadastrar um novo cargo com seus dados (CBO; cargo; salário e descrição).

RF6. O software deve permitir alterar todos os dados de um cargo.

RF7. O software deve permitir excluir todos os dados de um cargo.

RF8. O software deve permitir consultar todos os dados de um cargo.

→ Gerenciamento de Funções

RF9. O software deve permitir cadastrar uma nova função com seus dados (Código; descrição de função; data de efetivação e nota de desempenho).

RF10. O software deve permitir alterar todos os dados de uma função.

RF11. O software deve permitir excluir todos os dados de uma função. Somente deve ser permitida a exclusão de uma função que não esteja relacionada a nenhum cargo ou empregado. Quando uma função é excluída, os dados dos empregados e do cargo não devem ser excluídos.

RF12. O software deve permitir consultar todos os dados de uma função.

RF13. O software deve permitir consultar uma prévia das taxas e descontos que serão realizadas em folha salarial do empregado. Para isso, o software deverá acessar a entidade Taxas de outro software, o Sistema de Taxas (SISTA).

Apêndice B - Comparativo entre as abordagens propostas por trabalhos relacionados

Quadro 8 – Resumo dos mapeamentos propostos entre APF e Diagramas UML (Parte I)

Conceito	Rao, Nagaraj e Ahuja (2008)	Harput, Kaindl e Kramer (2005)
Fronteira	D. de Casos de Uso: definida entre ator e software	[não definido]
Função de dados	D. de Sequência: elementos não atores que trocam mensagens são candidatos a função de dados	D. de Classes: classes únicas, em uma hierarquia de generalização e em uma agregação podem ser mapeadas como um ALI ou AIE
ALI	D. de Classes: funções de dados com operações de troca de atributos de outros elementos	D. de Classes: classes ou grupos de classes no modelo de informações
AIE	[não definido]	D. de Classes: algumas classes ou grupos de classes no modelo de domínio, exceto às mapeadas como ALI
Função de transação	D. de Sequência: todas mensagens trocadas nos diagramas	D. de Casos de Uso: todos os casos de uso, incluindo suas pré-condições e suas pós-condições D. de Sequência: todas mensagens nos diagramas, incluindo seus “retornos”
EE	D. de Sequência: mensagens enviadas de um ator para uma função de dados	[não definido]
SE	D. de Sequência: mensagens enviadas de uma função de dados para um ator, que retorna mensagens ou contém uma combinação de EE e CE	[não definido]
CE	D. de Sequência: mensagens enviadas de uma função de dados para um ator	[não definido]

Quadro 9 – Resumo dos mapeamentos propostos entre APF e Diagramas UML (Parte II)

Conceito	Uemura, Kusumoto e Inoue (2001)	Chamundeswari e Babu (2008)
Fronteira	D. de Sequência: definida pelo tipo de elemento que aparece no diagrama (atores estão fora da fronteira; outras entidades, dentro)	D. de Classes: definida a partir de um conjunto de classes que executa diferentes funcionalidades do software
Função de dados	D. de Sequência: elementos que possuem atributos e trocam dados com entidade não atores	D. de Classes: todas as classes do diagrama
ALI	D. de Sequência: elementos que possuem atributos alterados pelas operações de outros elementos	D. de Classes: todas as classes dentro da fronteira
AIE	D. de Sequência: elementos que não possuem atributos alterados pelas operações de outros elementos	D. de Classes: classes dentro da fronteira externa (ou seja, um conjunto de classes referenciadas pelo software)
Função de transação	D. de Sequência: sequências de mensagem disparadas por um elemento do tipo ator	D. de Classes: todos métodos que ativam uma determinada operação
EE	D. de Sequência: elemento do tipo ator envia uma mensagem com argumentos a uma função de dados	D. de Classes: métodos da classe
SE	D. de Sequência: uma função de dados envia uma mensagem para um elemento do tipo ator, e os argumentos da mensagem são diferentes dos dados existentes na função de dados	[não mapeado]
CE	D. de Sequência: uma função de dados envia uma mensagem para um elemento do tipo ator, e os argumentos da mensagem são os mesmos dados existentes na função de dados	[não mapeado]

Quadro 10 – Resumo dos mapeamentos propostos entre APF e Diagramas UML (Parte III)

Conceito	Cantone; Pace; Calavaro (2004)	Pow-Sang et al., 2013)
Fronteira	[não definido]	[não definido]
Função de dados	D. de classes: São candidatos a funções de dados classes com o estereótipo “Entidade”.	D. de classes: As relações de agregação, composição, associação e generalização entre as classes do diagrama são analisadas para identificar as funções de dados.
ALI	D. de classes: Todas as classes do Diagrama de classes que não foram classificados como AIE.	[não definido]
AIE	D. de classes: Arquivos mapeados como classes que encapsulam componentes externos.	[não definido]
Função de transação	D. de sequência: Sequência de mensagens, na qual, a primeira é enviada por um objeto-ator e a última é recebida por um objeto-ator ou um objeto não ator.	[não definido]
EE	D. de sequência: Ocorre quando todas as mensagens da sequência (classificada como processo elementar) possuem argumentos.	[não definido]
SE	D. de sequência: Ocorre quando os argumentos da sequência de mensagem (classificada como um processo elementar) não incluem todos os atributos dos objetos lidos.	[não definido]
CE	D. de sequência: Ocorre quando os argumentos da sequência de mensagem (classificada como um processo elementar) incluem todos os atributos dos objetos lidos.	[não definido]

Quadro 11 – Resumo dos mapeamentos propostos entre APF e Diagramas UML (Parte IV)

Conceito	Lavazza, Bianco, Caravaglia (2008)
Fronteira	Diagrama de componentes: A fronteira do software é representada por um componente que encapsula demais componentes e deve ser definido pelo responsável pela elaboração do diagrama.
Função de dados	Diagrama de componentes: Cada componente presente no diagrama com estereotipo <LogicData>. (O Diagrama de componentes deve ser construído por um especialista em APF – ou com seu auxílio – para identificar corretamente quais componentes representam arquivos lógicos).
ALI	Diagrama de componentes: Componentes com estereotipo de <LogicData> dentro do componente definido como fronteira do software.
AIE	Diagrama de componentes: Componentes com estereotipo de <LogicData> fora do componente definido como fronteira do software.
Função de transação	Diagrama de casos de uso: Cada caso de uso presente no diagrama. (O Diagrama de casos de uso deve ser construído por um especialista em APF – ou com seu auxílio – para representar corretamente os casos de uso no nível de abstração necessário)
EE	Diagrama de sequência: O usuário deveria informar para cada Diagrama qual a intenção primária do Diagrama. O tipo de função era identificado a partir dessa informação. A proposta não apresenta as regras para identificar cada tipo de função de transação, a partir do preenchimento dessa informação pelo usuário.
SE	D. de sequência: Ocorre quando os argumentos da sequência de mensagem (classificada como um processo elementar) não incluem todos os atributos dos objetos lidos.
CE	D. de sequência: Ocorre quando os argumentos da sequência de mensagem (classificada como um processo elementar) incluem todos os atributos dos objetos lidos.

Fonte: Marcos Freitas Junior, 2015

Quadro 12 – Mapeamento de conceitos de APF para MCbER

Conceito	Abrahão e Insfrán (2008)
Fronteira	Uma linha imaginária traçada no Diagrama de casos de uso
Escopo	Todos os casos de uso do Diagrama de casos de uso
Função de dados	[Não se aplica]
ALI	Representa uma entidade classe
AIE	Representa uma entidade classe que está fora do software, mas é referenciada por ele
Função de transação	[Não se aplica]
EE	Representa uma mensagem do tipo «signal», com um valor de entrada, ou uma mensagem dos tipos «service/destroy», «service/update», «service/new», «connect»
SE	Representa uma mensagem do tipo «query» com uma mensagem do tipo «signal», ou uma mensagem do tipo «query» com processamento adicional.
CE	Representa uma mensagem do tipo «query»

Fonte: Marcos Freitas Junior, 2015

Quadro 13 – Pontuações atribuídas às abordagens de melhoria em relação à cobertura de APF (Parte I)

Proposta melhoria:	A	B	C	D	E	F	G
Artefato usado:	1, 2, 3	2	1, 2, 3	1	1, 2, 3	4	2
1 Englobam todos os passos para a identificação de função de transação?	P	P	P	P	P	P	-
1.1 Identificam o processo elementar?	S	S	S	S	S	S*	-
1.2 Realizam o teste de unicidade?	N	N	N	N	N	N	-
1.3 Classificam o processo elementar? (Todas as variações possíveis)	S	N	N	S	S	S	-
1.4 Determinam a complexidade e contribuição?	N	P	P	N	S	S	-
2 Englobam todos os passos para a identificação de DER?	N	N	P	N	N	P	-
2.1 Identificam os atributos únicos reconhecidos pelos usuários?	N	N	N	N	N	N	-
2.2 Identificam possíveis atributos repetidos?	N	N	S	N	N	N	-
2.3 Consideram um DER adicional para a habilidade de iniciar ação?	N	N	S	N	N	S	-
2.4 Consideram um DER adicional para a habilidade de exibir mensagem?	N	N	S	N	N	N	-
3 Englobam todos os passos para a identificação de ALR?	N	S	S	N	N	S	-
PONTUAÇÃO	2, 0	2, 5	5, 5	2, 0	3, 0	5, 5	-

4	Englobam todos os passos para a identificação de funções de dados?	N	N	P	N	N	P	P
4.1	Identificam todos os dados ou informações de controle reconhecidas pelos usuários logicamente relacionados no escopo da contagem?	S	S	S	N	S	S	S
4.2	Excluem as entidades não mantidas por qualquer aplicação?	N	N	N	N	N	N	N
4.3	Agrupam em arquivos lógicos as entidades dependentes?	N	N	S	N	N	S	S
4.4	Excluem aquelas entidades classificadas como dados de código?	N	N	N	N	N	N	N
4.5	Excluem as entidades que não contenham atributos exigidos pelo usuário?	N	N	N	N	N	N	N
4.6	Removem as entidades associativas que contenham atributos adicionais não exigidos pelo usuário e entidades associativas que contenham apenas chaves estrangeiras com as entidades principais?	N	N	N	N	N	N	N
5	Classificam os Arquivos Lógicos em ALI ou AIE?	P	S	S	S	S	S*	N
6	Englobam todos os passos para a identificação de DER?	N	P	N	N	N	P	P
6.1	Identificam os atributos únicos reconhecidos pelos usuários?	N	N	N	N	N	N	N
6.2	Identificam possíveis atributos repetidos?	N	N	N	N	N	N	N

6.3	Consideram um DER adicional para o relacionamento entre duas entidades?	N	S	S	N	N	N	S
7	Englobam todos os passos para a identificação de RLR?	N	P	P	N	P	P	P
7.1	Análise de entidades atributivas?	N	N	S	N	S	S	S
7.2	Análise de entidades subtipos?	N	S	S	N	S	S	S
7.3	Análise de entidades associativas?	N	N	P	N	N	N	P
PONTUAÇÃO		1,5	4,0	6,5	1,0	4,0	6,5	5,5

* Realizado, necessariamente, com o apoio de responsável com conhecimento em APF.

LEGENDA

Sigla	Descrição	Sigla	Descrição	Sigla	Descrição
A	(RAO; NAGARAJ; AHUJA, 2008)	F	(LAVAZZA; BIANCO; CARAVAGLIA, 2008)	4	Diagrama de Componentes
B	(CHAMUNDESWARI; BABU, 2008)	G	(POW-SANG ET AL, 2013)	S	SIM
C	(HARPUT; KAINDL; KRAMER, 2005)	1	Diagrama de Sequência	N	NÃO
D	(UEMURA; KUSUMOTO; INOUE, 2001)	2	Diagrama de Classe	P	PARCIALMENTE
E	(CANTONE; PACE; CALAVARO, 2004)	3	Diagrama de Caso de Uso		

Quadro 14 – Pontuações atribuídas às abordagens de melhoria em relação à cobertura de APF (Parte II)

Proposta melhoria:	H	I	J	K	L	M	N	O	Q
Artefato usado:	5	6	7	8	9	9	10	11	-
1 Englobam todos os passos para a identificação de função de transação?	P	P	P	S	P	P	P	P	S
1.1 Identificam o processo elementar?	S	N	S	S	S	N	N	N	S
1.2 Realizam o teste de unicidade?	N	N	N	S	N	N	N	N	S
1.3 Classificam o processo elementar? (Todas as variações possíveis)	S	S	S	S	S	S	S	S	S
1.4 Determinam a complexidade e contribuição?	N	S	S	S	S	N	S	S	S
2 Englobam todos os passos para a identificação de DER?	N	N	P	N	N	N	N	P	S
2.1 Identificam os atributos únicos reconhecidos pelos usuários?	N	N	N	N	N	N	N	N	S
2.2 Identificam possíveis atributos repetidos?	N	N	S	N	N	N	N	N	S
2.3 Consideram um DER adicional para a habilidade de iniciar ação?	N	N	S	N	N	N	N	S	S
2.4 Consideram um DER adicional para a habilidade de exibir mensagem?	N	N	S	N	N	N	N	S	S
3 Englobam todos os passos para a identificação de ALR?	S	S	S	S	N	N	S		S
PONTUAÇÃO	3,0	3,0	7,0	5,0	2,0	1,0	3,0	4,0	9,0
4 Englobam todos os passos para a identificação de funções de dados?	N	N	N	P	N	N	P	N	S

7.2	Análise de entidades subtipos?	N	N	N	N	N	N	S	S	S
7.3	Análise de entidades associativas?	N	N	N	N	N	N	S	S	S
PONTUAÇÃO		1,0	1,0	1,0	4,0	1,0	0,0	8,5	5,0	13,0

LEGENDA

Sigla	Descrição	Sigla	Descrição
H	(ABRAHÃO; INSFRÁN, 2008)	5	MCbER
I	(MIYAWAKI; IJIMA; HO, 2008)	6	VDM-SL
J	(CHOI; PARK; SUGURAN, 2006)	7	Requisitos
K	(FRATERNALI; BONGIO, 2006)	8	WebML
L	(EDGAWA ET AL., 2011)	9	Código Fonte
M	(KLUSENER, 2003)	10	ER+DFD
N	(LAMMA; MELLO; RIGUZZI, 2004)	11	OO-H
O	(ABRAHÃO ET AL., 2007)	S	SIM
Q	ABORDAGEM "IDEAL"	N	NÃO
		P	PARCIALMENTE

Fonte: Marcos Freitas Junior, 2015

Apêndice C – Artigos publicados e relacionados à dissertação

Relacionados a esse trabalho foram publicados antes e/ou durante o seu desenvolvimento dois artigos, listados abaixo:

- **Artigo 1:** *Improvements to the Function Point Analysis Method: A Systematic Literature Review*
 - **Autores:** Marcelo Fantinato; Marcos de Freitas Junior; Violeta Sun.
 - **Estado:** Publicado.
 - **Data de publicação:** 23 de Julho de 2015.
 - **Período:** IEEE Transactions on Engineering Management.
 - **JCR:** 1.103 (data última consulta: 20-07-2015).
 - **Resumo:** Function Point Analysis (FPA) is a standardized method to measure the functional size of a software. This method is currently recommended by governments and organizations as a standard method to be adopted for this type of measurement. This paper presents a compilation of improvements, focused on increasing the accuracy of the FPA method, which have been proposed over the past 13 years. The methodology used was a Systematic Literature Review (SLR) which was conducted with four research questions aligned with the objectives of this study. As a result of the SLR, of the 1600 results returned by the search engines, 454 primary studies were pre-selected according to the criteria established for the SLR. Among these studies, only 18 specifically referred to accuracy improvements for FPA, which was the goal of this work. It was found that the step for calculating the functional size exhibited the highest number of problems, indicating the need to revise FPA in order to encompass the possible improvements suggested by the researchers.
 - **Relação com o presente trabalho:** A revisão sistemática realizada, embora distinta da apresentada no capítulo 2 deste trabalho, possibilitou com que fosse possível obter maior entendimento do estado da arte para a medida de APF, bem como possibilitou que nova revisão sistemática da literatura fosse realizada com maior perfeição em termos de escopo.

- **Artigo 2:** *Uso da análise de ponto de função em projetos de desenvolvimento de software*
 - **Autores:** Edmir Parada Vasques Prado; Marcos de Freitas Junior; Violeta Sun.
 - **Estado:** Publicado.
 - **Data de publicação:** 12 de Julho de 2012.
 - **Simpósio:** I Simpósio Internacional em Gerenciamento de Projetos 2012.
 - **Trilha de Conferência:** Artigos completos (para o qual foi submetido o presente artigo), Relatório técnico e pôster.
 - **Resumo:** Esta pesquisa tem como objetivo analisar o processo de contagem de pontos de função da medida de APF (Análise de Ponto de Função) realizado por profissionais certificados para essa atividade. Ela representa um estudo exploratório e qualitativo. A pesquisa fez uma revisão bibliográfica sobre APF e usou o estudo de caso como método de pesquisa. Os dados foram coletados por meio de entrevistas e pela análise, feita pelos entrevistados, de um cenário de desenvolvimento de software construído para avaliar a contagem de pontos de função. Foi possível identificar que há divergências nas contagens realizadas pelos informantes, que identificaram tamanhos funcionais, para o cenário proposto, com diferenças de até 90%.
 - **Relação com o presente trabalho:** O estudo de caso realizado possibilitou que fossem coletadas evidências (não sistemáticas) de que existiam divergências entre contagens realizadas por distintos especialistas. Essa é a principal motivação para a criação de APF-APF no capítulo 5.

Apêndice D – Protocolo de revisão sistemática da literatura

A seguir, o protocolo para a execução da RSL (descrito de modo resumido no capítulo 4) é detalhado.

Para a execução da RSL foram elaboradas quatro questões de pesquisa conforme descritas no quadro 15. As questões de pesquisa foram definidas considerando, como principal objetivo, a necessidade de identificar trabalhos que tinham como propósito aumentar a confiabilidade da aplicação da medida APF.

Quadro 15 – Questões de pesquisa da RSL

#	Questão de pesquisa	Motivação
1	Existem problemas sendo relatados para APF?	Identificar problemas potenciais em APF, apontados por pesquisadores
2	Quais melhorias estão sendo propostas para APF em termos de confiabilidade?	Identificar os principais pontos de APF que carecem de melhorias
3	As melhorias propostas se mostraram efetivas?	Verificar se as melhorias propostas se mostraram mais efetivas quando comparadas à APF original
4	Quais limitações existem para as melhorias propostas?	Identificar possíveis problemas ou fragilidades para as melhorias propostas

Fonte: Marcos Freitas Junior, 2015

Considerando as recomendações de Kitchenham [10], as questões de pesquisa foram definidas com base em: (i) uma determinada intervenção, ou seja, uma tecnologia de software destinada a problemas específicos; (ii) uma população, ou seja, o que é afetado pela intervenção; e (iii) os resultados, ou seja, as implicações da pesquisa relacionadas a fatores importantes para praticantes. Assim, especificamente para esta RSL, as questões de pesquisa foram definidas com base em:

- **Intervenção:** medida APF;
- **População:** estudos primários que definem abordagens para melhorar a acurácia do tamanho funcional do software calculado com APF;
- **Resultados:** visão abrangente das abordagens propostas que visavam melhorar os resultados gerados com APF.

A busca dos estudos primários foi realizada usando quatro bases de dados bibliográficas, sendo elas: ACM Digital Library, IEEEExplore, Scopus e Science Direct. A busca foi inicial foi realizada no período de outubro a dezembro de 2013 e posteriormente, em segunda iteração, em janeiro e fevereiro de 2015. Essas bases de dados foram escolhidas por possuírem pelo menos uma máquina de busca online com opções para: (i) pesquisa avançada por meio de palavras-chave e (ii) filtragem de resultados por ano de publicação e por área ou tipo de publicação. As strings de busca específicas usadas em cada uma das quatro base de dados estão apresentadas no quadro 16.

Quadro 16 – Strings de busca

Base	String de Busca	Artigos Retornados	1 ^a iteração	2 ^a iteração
Scopus	(TITLE-ABS-KEY(“function point” OR “function points” OR “function-point” OR “function-points”)) AND PUBYEAR > 2001 AND (LIMIT-TO(DOCTYPE, “cp”) OR LIMIT-TO(DOCTYPE, “ar”) OR LIMIT-TO(DOCTYPE, “ip”)) AND (LIMIT-TO(LANGUAGE, “English”)) AND (LIMIT-TO(SUBJAREA, “COMP”) OR LIMIT-TO(SUBJAREA, “ENGI”) OR LIMIT-TO(SUBJAREA, “BUSI”)) AND (LIMIT-TO(SRCTYPE, “p”) OR LIMIT-TO(SRCTYPE, “j”) OR LIMIT-TO(SRCTYPE, “k”)) AND (LIMIT-TO(LANGUAGE, “English”)).	702	105	12
IEEE Xplore	(“function point”) OR (“function points”) OR (“function-point”) OR (“function-points”) – Publication Year: 2002-2014; Content Type: Conference Publications, Journals & Magazines, Early Access Articles, Standards.	187	48	8
ACM Digital Library	(“function point” OR “function points” OR “function-point” OR “function-points”). Publication Year: 2002...2014.	148	35	6
Science Direct	pub-date > 2001 and “function point” OR “function points” OR “function-point” OR “function-points{Journals(Computer Science, Engineering)}”	563	22	1

Fonte: Marcos Freitas Junior, 2015

A busca foi realizada em duas iterações. Na primeira iteração, foram selecionados os resultados que mais se aproximaram do objetivo desta RSL. Para isso, houve a leitura

cuidadosa dos títulos dos artigos. Em caso de dúvidas referentes à aplicabilidade de artigos para esta RSL, houve também a leitura dos resumos. Nessa primeira iteração, foram selecionados 44 artigos: 24 da base Scopus, nove da base IEEEEXplorer; quatro da base ACM Digital Library, e sete da base Science Direct.

Na segunda iteração, os artigos foram lidos integralmente para determinar se os estudos selecionados, na primeira iteração, estavam, de fato, em consonância com os objetivos definidos para esta RSL. Como resultado final, foram selecionados 15 artigos: 12 da Scopus, 8 da IEEEEXPlore; 6 da ACM, e 1 da Science Direct. As bases de dados trouxeram artigos duplicados entre si que foram considerados apenas uma vez na análise de dados.

A seleção dos artigos foi baseada nos seguintes critérios de inclusão e de exclusão definidos para refinar a seleção de artigos considerando os objetivos desta RSL e também para garantir a qualidade dos estudos primários sendo selecionados:

a. Critérios de inclusão:

- I. Artigo com foco em APF cujo principal objetivo é identificar alguma limitação ou deficiência na medida em termos de acurácia;
- II. Artigo com foco em APF cujo principal objetivo é – ao identificar uma determinada limitação ou deficiência da medida – propor melhorias nos procedimentos existentes na medida para solucionar os problemas de acurácia identificados;
- III. Artigo cujo objetivo principal é investigar a relação de FPA a um determinado paradigma de modelagem (OO), linguagem de modelagem (UML) ou a uma tecnologia com o intuito de interpretar as regras existentes no CPM para um determinado contexto;
- IV. Artigo publicado a partir de 1 de janeiro de 2001. Esta data foi definida, baseada no na última versão do CPM versão 4.1, lançada em 1999. Esta foi a última versão lançada sem incluir novas regras de contagens. Os versionamentos posteriores (4.1.1; 4.2; 4.2.1; 4.3 e 4.3.1) não modificaram ou incluíram qualquer regra publicada anteriormente. Apenas foram incluídos novos exemplos e foram realizados ajustes pontuais da redação do manual. Assim, considerou-se que trabalhos submetidos antes de 2001 possivelmente, apresentavam artigos referentes a medida em sua versão antiga. Por este motivo, os mesmos não foram considerados;
- V. Artigo publicado e disponível integralmente online em bases de dados científicas;

- VI. Artigos publicados em periódico ou em anais de evento, com revisão por pares;
- VII. Artigo publicado inteiramente no idioma inglês.

b. Critérios de exclusão:

- VIII. Artigo que propõe uma nova medida para a medição funcional de software alternativa a FPA;
- IX. Artigo que apresenta um estudo secundário;
- X. Artigo cujo objetivo é realizar melhorias em procedimentos de APF para aumentar a rapidez de cálculo ou quaisquer outros fatores sem que esses tornem a aplicação de FPA mais acurada;
- XI. Artigo cujo objetivo é investigar a relação de APF (ou compará-la) com outras medidas de medição funcional;
- XII. Artigo cujo objetivo é investigar a conversibilidade do tamanho funcional medido com APF com o tamanho gerado por outras medidas de medição funcional;
- XIII. Artigos que proponham melhorias para o passo de calcular o tamanho funcional ajustado, seja para alterar a fórmula proposta, bem como para alterar, incluir ou remover as características gerais do software.

Os critérios de inclusão II, III, IV e V foram aplicados durante a primeira iteração, na seleção dos estudos primários, por meio da avaliação dos títulos dos artigos e dos seus resumos, bem como nos filtros de pesquisas das máquinas de busca. Os critérios de inclusão I e V e todos os critérios de exclusão foram aplicados durante a segunda iteração, na seleção dos estudos primários, quando foi possível avaliar o conteúdo dos estudos primários selecionados.

Após a seleção final dos estudos primários, os mesmos foram lidos na íntegra novamente e resumidos para documentação das principais informações. Em paralelo, os dados necessários para responder as questões de pesquisa desta RSL foram extraídos dos estudos primários.

Durante a extração de dados, para cada um dos 15 estudos primários selecionados, foram preenchidos formulários estruturados. Além de informações básicas sobre os artigos (incluindo: dados bibliográficos data de publicação, resumo, etc.), foram armazenadas a síntese do artigo e reflexões – dos pesquisadores conduzindo a RSL – sobre o conteúdo e as conclusões do estudo.

O quadro 17 lista os 15 estudos primários selecionados que estão aderentes aos critérios de inclusão e exclusão definidos.

Quadro 17 – Estudos primários selecionados

Estudo primário selecionado	Ano	Veículo	Fonte(s)
Measuring the Function Points from the Points of Relationships of UML	2008	Conferência	S, I
An Extended Function Point Approach for Size Estimation of Object-Oriented Software	2010	Conferência	R
Extending Function Point Analysis of Object-Oriented Requirements Specifications	2005	Conferência	S, I
Function-point analysis using design specifications based on the Unified Modelling Language	2001	Periódico	S, A
Applying function point to unified modeling language: conversion model and pilot study	2004	Conferência	S, I
Model-based functional size measurement	2008	Conferência	S, A
A Conversion Model and a Tool to Identify Function Point Logic Files Using UML Analysis Class Diagrams	2013	Conferência	S, I
A Metamodeling Approach to Estimate Software Size from Requirements Specifications	2008	Conferência	I, A
Measuring function points from VDM-SL specifications	2008	Conferência	S, I
Function Point Extraction Method from Goal and Scenario Based Requirements Text*	2006	Conferência	S
Automating function point analysis with model driven development	2006	Conferência	S, I, A
Function point measurement from Web application source code based on screen transitions and database accesses	2011	Periódico	D, A
Source code based function point analysis for enhancement projects	2003	Conferência	S, I
A system for measuring function points from an ER-DFD specification	2004	Periódico	S
A model-driven measurement procedure for sizing web applications: Design, automation and validation	2007	Conferência	S, A

Fonte: Marcos Freitas Junior, 2015

Apêndice E – Diagramas de classe do protótipo de ferramenta

A seguir, o Diagrama de classes elaborado para o protótipo de ferramenta é apresentado em detalhes. Por questões de legibilidade, optou-se por apresentá-los seccionados por componente do MVC, de modo que cada bloco apresente uma das partes consideradas principais na lógica do protótipo de ferramenta elaborado. O Diagrama de Classes completo pode ser visualizado no seguinte endereço eletrônico: <https://dl.dropboxusercontent.com/u/18910158/Diagrama%20de%20Classes.pdf>

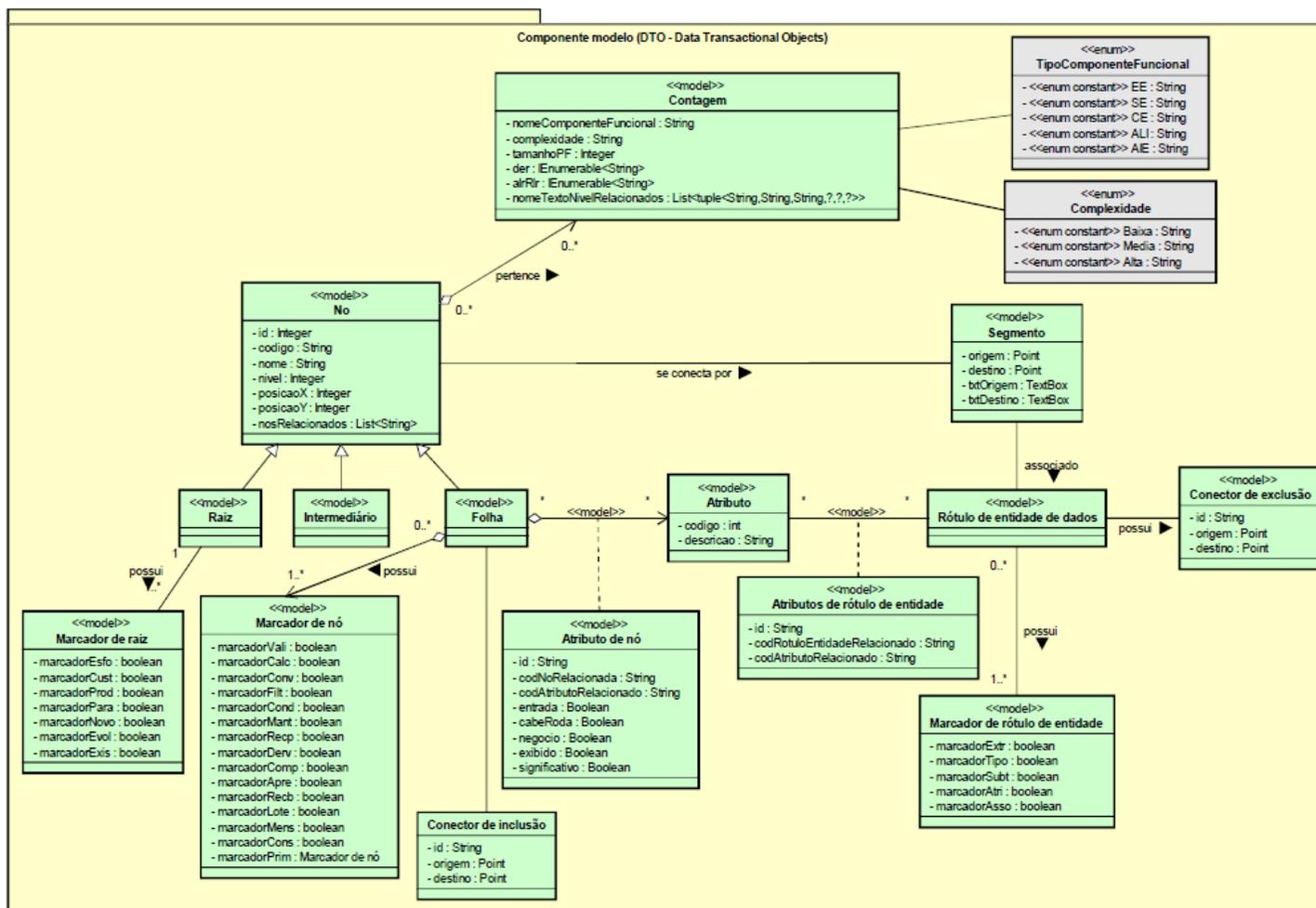
➤ Componente modelo

O componente modelo é composto de 17 classes divididas em dois blocos denominados “DataTransactionalObjects” e “Serviços”.

O primeiro bloco, apresentado na figura 15 contém as classes base que representam os componentes incorporados na Árvore de pontos de função. As classes desse bloco são usadas apenas para encapsular dados do protótipo da ferramenta quando é necessário enviá-los para outras classes. Os únicos métodos executados por essas classes são os *getters* e *setters* (omitidos da figura para possibilitar maior legibilidade), sendo que não é executada nenhuma regra de negócio. A principal classe desse bloco é a “Contagem”, responsável por armazenar os resultados da medição calculados pelo protótipo da ferramenta após a modelagem da Árvore de pontos de função incluindo as funções de transação, dados, DERs, ALRs e RLRs identificados. Associada a essa classe está a classe “No” que representam os nós “Raiz”, “Intermediários” e “Folha” da Árvore de pontos de função, sendo que, a cada nó conectam-se os componentes da Árvore de pontos de função aplicáveis.

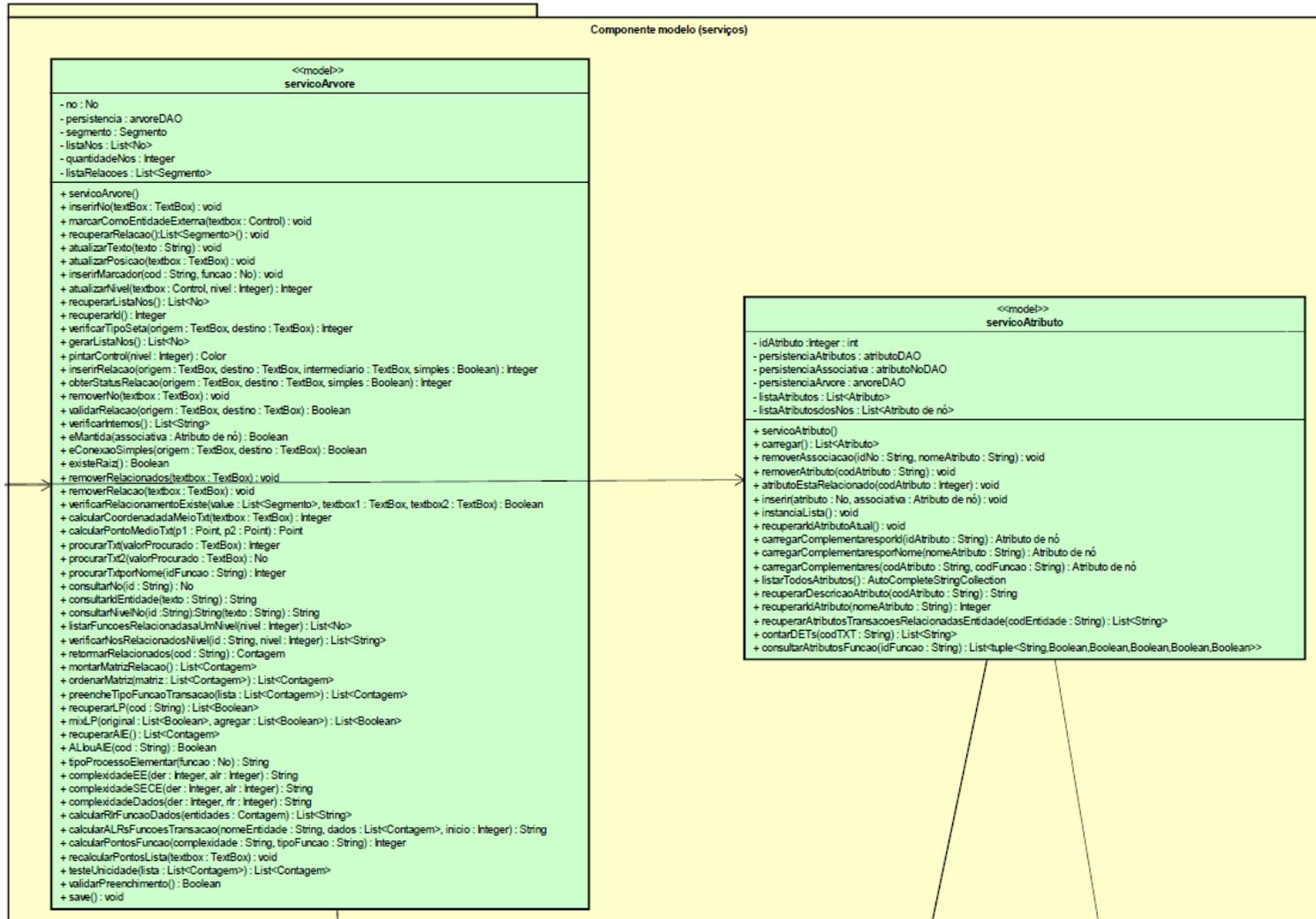
O segundo bloco, apresentado na figura 16 é composto de duas classes, “servicoArvore” e “servicoAtributo” responsáveis por centralizar as regras de negócio do protótipo da ferramenta. A primeira classe centraliza regras relacionadas a estrutura da Árvore de pontos de função, seus componentes e o cálculo do tamanho funcional, enquanto a segunda centraliza regras relacionadas aos atributos das funções elementares cadastrados pelos usuários. Ambas, possibilitam que as operações executadas na interface da ferramenta sejam validadas e interpretadas.

Figura 15 – Diagrama de casos de uso do bloco DataTransactionalObjects



Fonte: Marcos Freitas Junior, 2015

Figura 16 – Diagrama de casos de uso do bloco Serviços

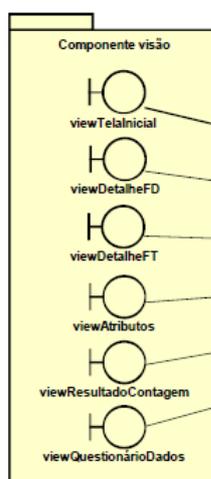


➤ Componente visão

O componente visão é composto por seis classes apresentadas na figura 17. As classes “viewTelaInicial”, “viewDetalheFD”, “viewDetalheFT” e “viewQuestionárioDados” são responsáveis por coletar dados do software modelado para formar a estrutura da Árvore de pontos de função. Por sua vez, as classes “viewAtributos” e “viewResultadoContagem” são responsáveis por apresentar, respectivamente, o resultado da contagem de pontos de função e os atributos cadastrados na Árvore de pontos de função.

Em nenhuma das classes pertencentes ao componente visão são executadas regras de negócio, de modo que todas são responsáveis apenas por apresentar informações ao usuário, como por exemplo uma mensagem informando que foi executada uma operação indevida na Árvore de pontos de função, ou coletar dados que são posteriormente enviados as classes “servicoArvore” ou “servicoAtributo” via classes do bloco DataTransactionalObjects.

Figura 17 – Diagrama de casos de uso do componente visão



Fonte: Marcos Freitas Junior, 2015.

O quadro 18 apresenta a descrição da principal responsabilidade para cada classe apresentada na figura 16.

➤ Componente persistência

O componente persistência é composto por três classes apresentadas na figura 18. As classes “ArvoreDAO”, “AtributosNoDAO” e “AtributosDAO” são responsáveis por executar

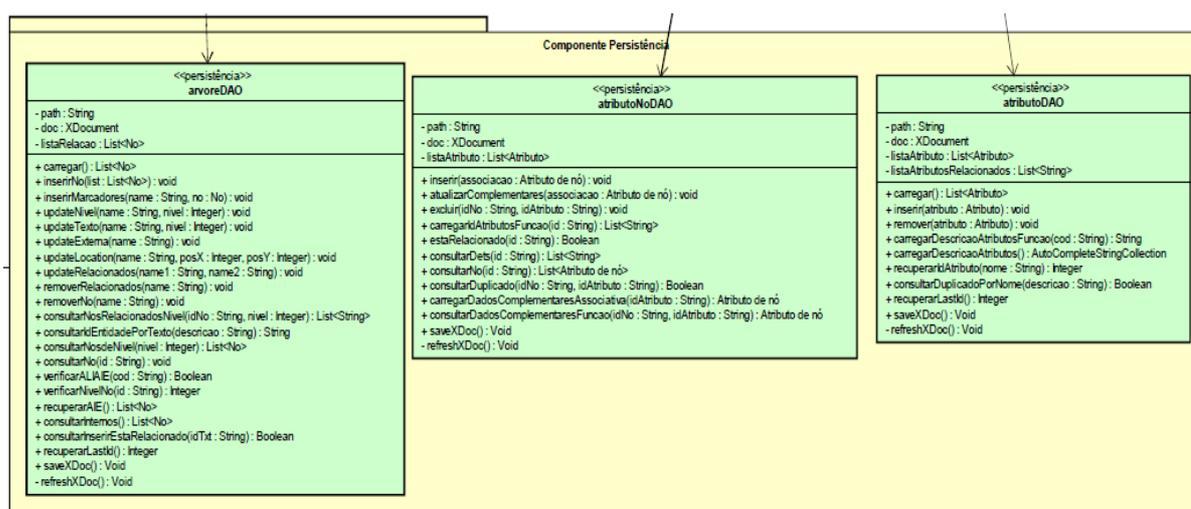
as operações no banco de dados relacionadas, respectivamente, com os componentes da Árvore de pontos de função, a associação dos nós folha aos atributos e aos atributos de nó e rótulo de entidade de dados.

Quadro 18 – Descrição das classes

Classe	Descrição
viewTelaInicial	Possibilita ao usuário modelar a estrutura de Árvore de pontos de função.
viewDetalheFD	Possibilita ao usuário informar os marcadores e atributos do rótulo de entidade de dados.
viewDetalheFT	Possibilita ao usuário informar os marcadores e atributos para um nó folha.
viewAtributos	Possibilita ao usuário visualizar todos os atributos de nó folha ou rótulo de entidade de dados cadastrados na árvore.
viewResultadoContagem	Possibilita ao usuário visualizar o resultado da contagem de pontos de função.
viewQuestionárioDados	Possibilita ao usuário informar a dependência entre as entidades do software.

Fonte: Marcos Freitas Junior, 2015.

Figura 18 – Diagrama de casos de uso do componente persistência



Fonte: Marcos Freitas Junior, 2015.

Apêndice F – Modelagens da Árvore de pontos de função entregue pelos respondentes após o experimento

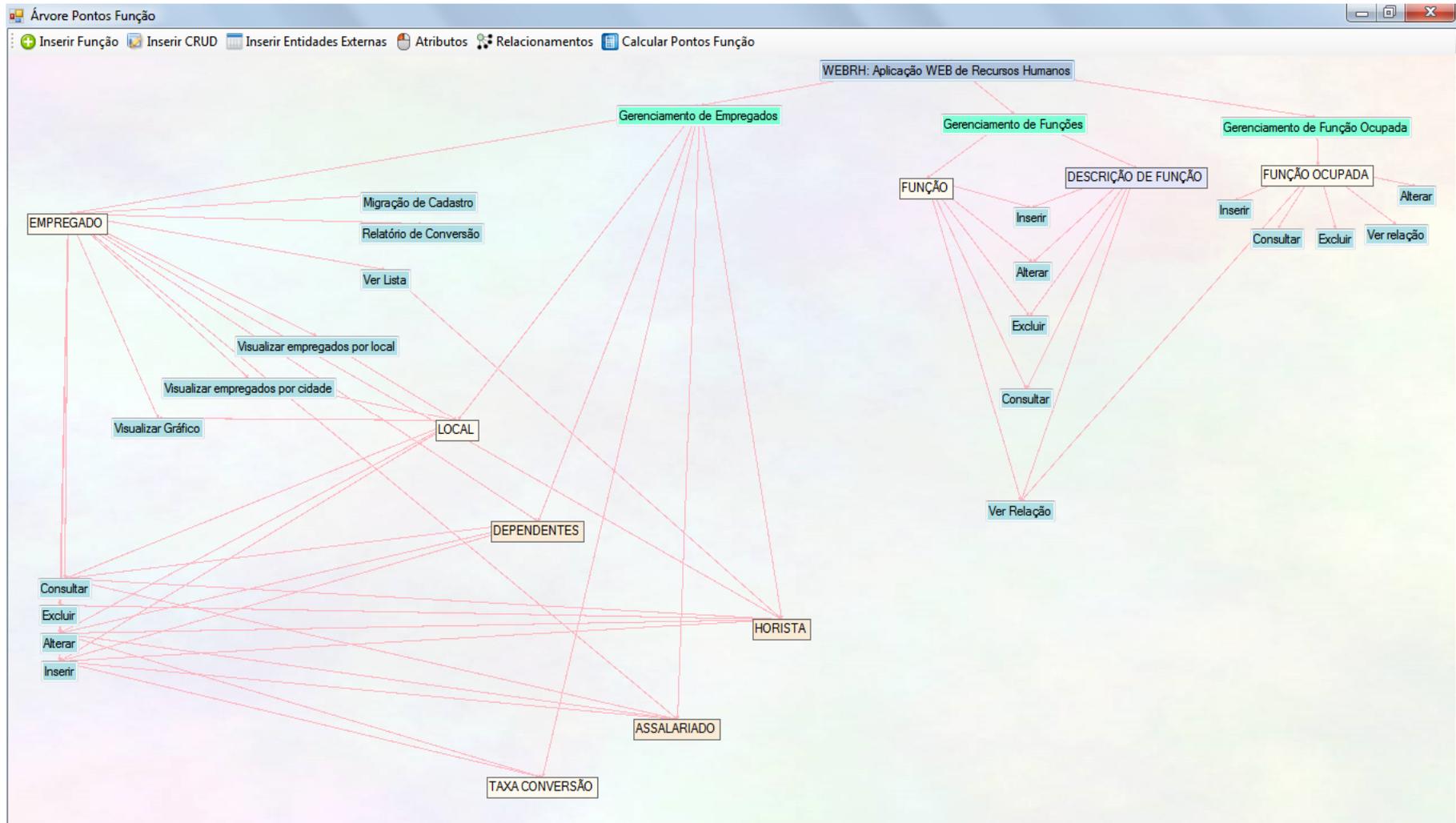
Esse apêndice, apresenta o detalhamento da contagem de pontos de função elaborada pelo IFPUG para o estudo de caso do Sistema de RH e as modelagens da Árvore de pontos de função e seu detalhamento realizado pelos participantes. O quadro 19 apresenta a contagem de pontos de função oficial do IFPUG.

Quadro 19 – Detalhamento da contagem de pontos de função oficial do estudo de caso do Sistema de RH

Nome Função de dados	Tipo	RLR	DERs	Complexidade	Tamanho	Subtotal
Empregado (Horista/ Assalariado / Dependente)	ALI	3	16	Baixa	7 PF	31 PF
Função (Descrição de função)	ALI	2	8	Baixa	7 PF	
Função Ocupada	ALI	1	5	Baixa	7 PF	
Local	AIE	1	7	Baixa	5 PF	
Taxa de Conversão	AIE	1	3	Baixa	5 PF	
Nome Função de transação	Tipo	ALR	DERs	Complexidade	Tamanho	Subtotal
Adicionar Empregado (Empregado / Taxa / Local)	EE	3	20	Alta	6 PF	43 PF
Alterar Empregados (Empregado / Taxa / Local)	EE	3	20	Alta	6 PF	
Excluir Empregados (Empregados / Função Ocupada)	EE	2	3	Baixa	3 PF	
Incluir Função (Função)	EE	1	9	Baixa	3 PF	
Alterar Função (Função)	EE	1	10	Baixa	3 PF	

Excluir Função (Função)	EE	2	3	Baixa	3 PF	
Incluir Função Ocupada (Função Ocupada / Empregado / Função)	EE	3	8	Alta	6 PF	
Alterar Função Ocupada (Função Ocupada / Empregado / Função)	EE	3	8	Alta	6 PF	
Excluir Função Ocupada (Função Ocupada / Empregado / Função)	EE	3	4	Média	4 PF	
Migração de Empregados (Empregado)	EE	1	8	Baixa	3 PF	
Listar Empregados (Empregados / Taxa / Local)	SE	3	17	Média	5 PF	32 PF
Listar Funções (Funções)	SE	1	9	Baixa	4 PF	
Listar Funções Ocupadas (Função Ocupada / Empregado / Função)	SE	3	8	Média	5 PF	
Consultar por Locais (Empregado / Local)	SE	2	9	Média	5 PF	
Consulta Por Local e Cidade (Empregado / Local)	SE	2	10	Média	5 PF	
Visualizar Gráfico Percentual País (Empregado / Local)	SE	2	3	Baixa	4 PF	
Listar Empregados Convertidos (Empregado)	SE	1	3	Baixa	4 PF	
Lista DropDown Locais (Local)	CE	1	2	Baixa	3 PF	19 PF
Lista DropDown Taxas (Taxa)	CE	1	2	Baixa	3 PF	
Consultar Empregado (Empregado / Taxa / Local)	CE	3	20	Alta	6 PF	
Consultar Função (Função)	CE	1	10	Baixa	3 PF	
Consultar Função Ocupada (Empregado / Função / Função Ocupada)	CE	3	8	Média	4 PF	
TOTAL DE PONTOS DE FUNÇÃO DO SISTEMA DE RH						125 PF

Figura 19 – Modelagem da Árvore de pontos de função do participante 1 via protótipo de ferramenta



Fonte: Marcos Freitas Junior, 2015.

Figura 20 – Detalhamento da contagem de pontos de função do participante 1 via protótipo de ferramenta

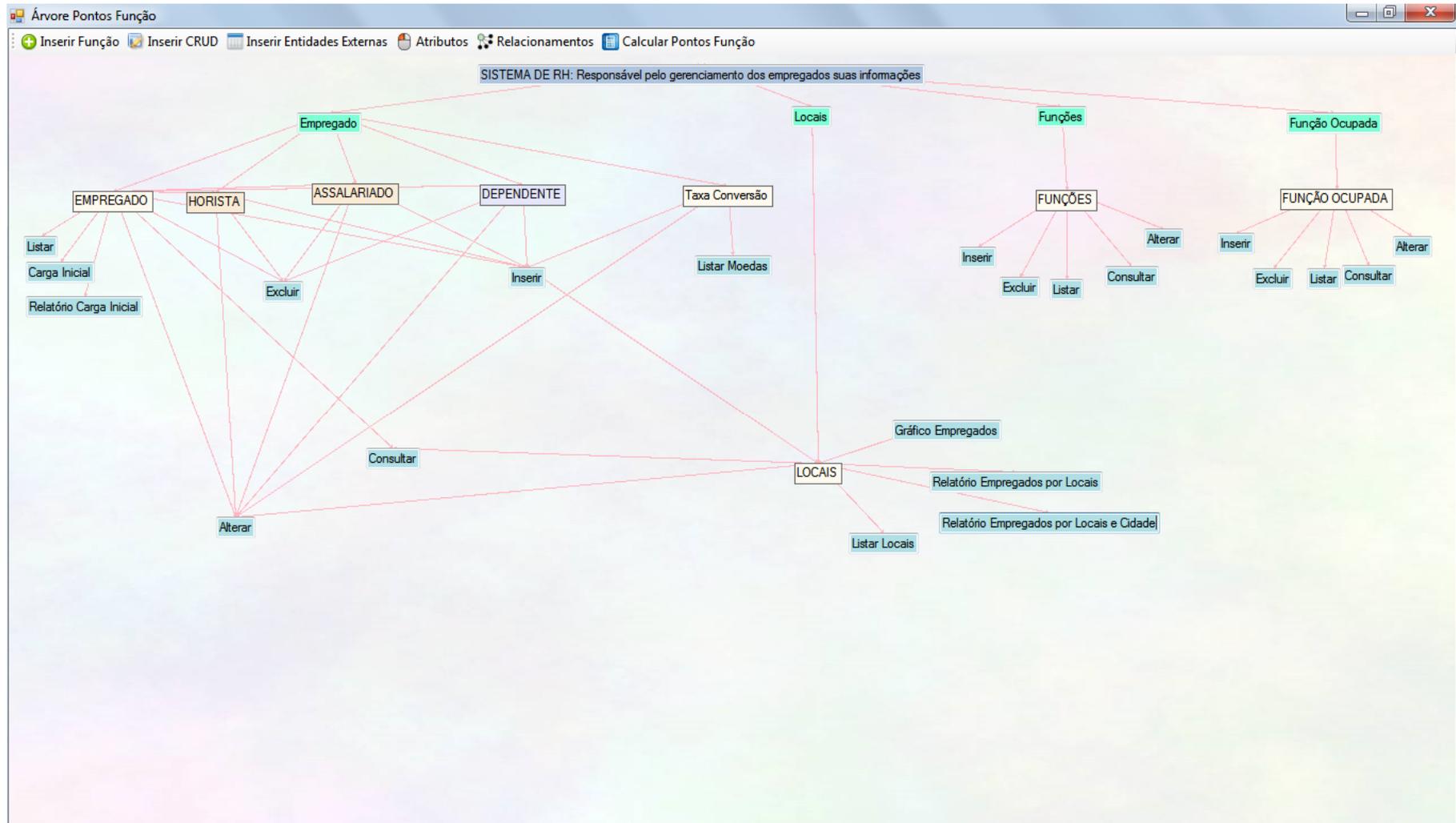
Resultado da contagem: 104 pontos de função!

Nome da Função de Transação	Tipo	ALRs/RLRs	DERs	Complexidade	Tamanho
EMPREGADO (HORISTA / DEPENDENTES / ASSALARIADO)	ALI	3	14	Baixa	7
FUNÇÃO (DESCRIÇÃO DE FUNÇÃO / FUNÇÃO)	ALI	2	4	Baixa	7
FUNÇÃO OCUPADA (FUNÇÃO OCUPADA)	ALI	1	5	Baixa	7
LOCAL (LOCAL)	AIE	1	7	Baixa	5
TAXA CONVERSÃO (TAXA CONVERSÃO)	AIE	1	3	Baixa	5
Alterar (FUNÇÃO OCUPADA)	EE	1	8	Baixa	3
Excluir (FUNÇÃO OCUPADA)	EE	1	4	Baixa	3
Consultar (FUNÇÃO OCUPADA)	CE	1	7	Baixa	3
Inserir (FUNÇÃO OCUPADA)	EE	1	8	Baixa	3
Ver relação (FUNÇÃO OCUPADA)	CE	1	7	Baixa	3
Migração de Cadastro (EMPREGADO)	EE	1	4	Baixa	3
Relatório de Conversão (EMPREGADO)	CE	1	4	Baixa	3
Excluir (EMPREGADO)	EE	1	3	Baixa	3
Ver Lista (EMPREGADO)	CE	1	15	Baixa	3
Inserir (FUNÇÃO)	EE	1	9	Baixa	3
Alterar (FUNÇÃO)	EE	1	9	Baixa	3
Excluir (FUNÇÃO)	EE	1	3	Baixa	3
Consultar (FUNÇÃO)	CE	1	9	Baixa	3
Visualizar empregados por local (EMPREGADO / LOCAL)	SE	2	9	Média	5
Visualizar empregados por cidade (EMPREGADO / LOCAL)	SE	2	10	Média	5
Visualizar Gráfico (EMPREGADO / LOCAL)	SE	2	3	Baixa	4
Ver Relação (FUNÇÃO)	SE	1	9	Baixa	4
Consultar (EMPREGADO / LOCAL)	CE	2	17	Média	4
Inserir (EMPREGADO / LOCAL / TAXA CONVERSÃO)	EE	3	18	Alta	6
Alterar (EMPREGADO / LOCAL / TAXA CONVERSÃO)	EE	3	18	Alta	6

Detalhar

Fonte: Marcos Freitas Junior, 2015.

Figura 21 – Modelagem da Árvore de pontos de função do participante 2 via protótipo de ferramenta



Fonte: Marcos Freitas Junior, 2015.

Figura 22 - Detalhamento da contagem de pontos de função do participante 2 via protótipo de ferramenta

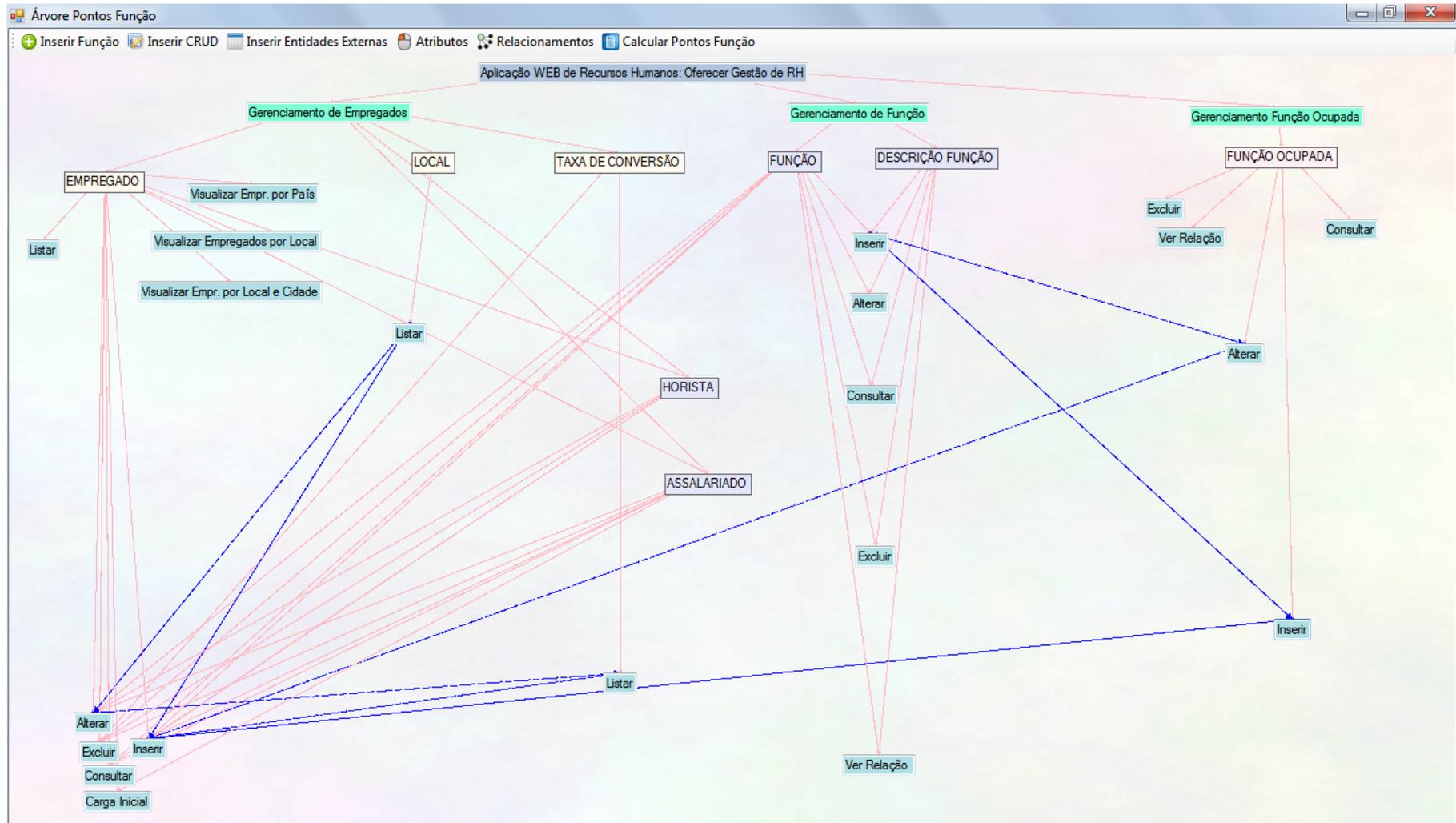
Resultado da contagem: 109 pontos de função!

Nome da Função de Transação	Tipo	ALRs/RLRs	DERs	Complexidade	Tamanho
EMPREGADO (DEPENDENTE / ASSALARIADO / HORISTA)	ALI	3	14	Baixa	7
FUNÇÕES (FUNÇÕES)	ALI	1	3	Baixa	7
FUNÇÃO OCUPADA (FUNÇÃO OCUPADA)	ALI	1	3	Baixa	7
LOCAIS (LOCAIS)	AIE	1	7	Baixa	5
Taxa Conversão (Taxa Conversão)	AIE	1	3	Baixa	5
Listar (EMPREGADO)	SE	1	16	Baixa	4
Inserir (FUNÇÕES)	EE	1	9	Baixa	3
Alterar (FUNÇÕES)	EE	1	10	Baixa	3
Excluir (FUNÇÕES)	EE	1	3	Baixa	3
Consultar (FUNÇÕES)	CE	1	10	Baixa	3
Listar (FUNÇÕES)	SE	1	8	Baixa	4
Inserir (FUNÇÃO OCUPADA)	EE	1	8	Baixa	3
Alterar (FUNÇÃO OCUPADA)	EE	1	8	Baixa	3
Excluir (FUNÇÃO OCUPADA)	EE	1	4	Baixa	3
Consultar (FUNÇÃO OCUPADA)	CE	1	8	Baixa	3
Listar (FUNÇÃO OCUPADA)	CE	1	7	Baixa	3
Relatório Empregados por Locais (LOCAIS)	SE	1	7	Baixa	4
Gráfico Empregados (LOCAIS)	CE	1	3	Baixa	3
Carga Inicial (EMPREGADO)	EE	1	8	Baixa	3
Relatório Carga Inicial (EMPREGADO)	SE	1	3	Baixa	4
Listar Moedas (Taxa Conversão)	CE	1	5	Baixa	3
Listar Locais (LOCAIS)	CE	1	2	Baixa	3
Relatório Empregados por Locais e Cidade (LOCAIS)	SE	1	9	Baixa	4
Consultar (EMPREGADO / LOCAIS)	CE	2	18	Média	4
Excluir (EMPREGADO)	EE	1	3	Baixa	3
Inserir (EMPREGADO / LOCAIS / Taxa Conversão)	EE	3	19	Alta	6
Alterar (EMPREGADO / LOCAIS / Taxa Conversão)	EE	3	16	Alta	6

Detalhar

Fonte: Marcos Freitas Junior, 2015.

Figura 23 – Modelagem da Árvore de pontos de função do participante 3 via protótipo de ferramenta



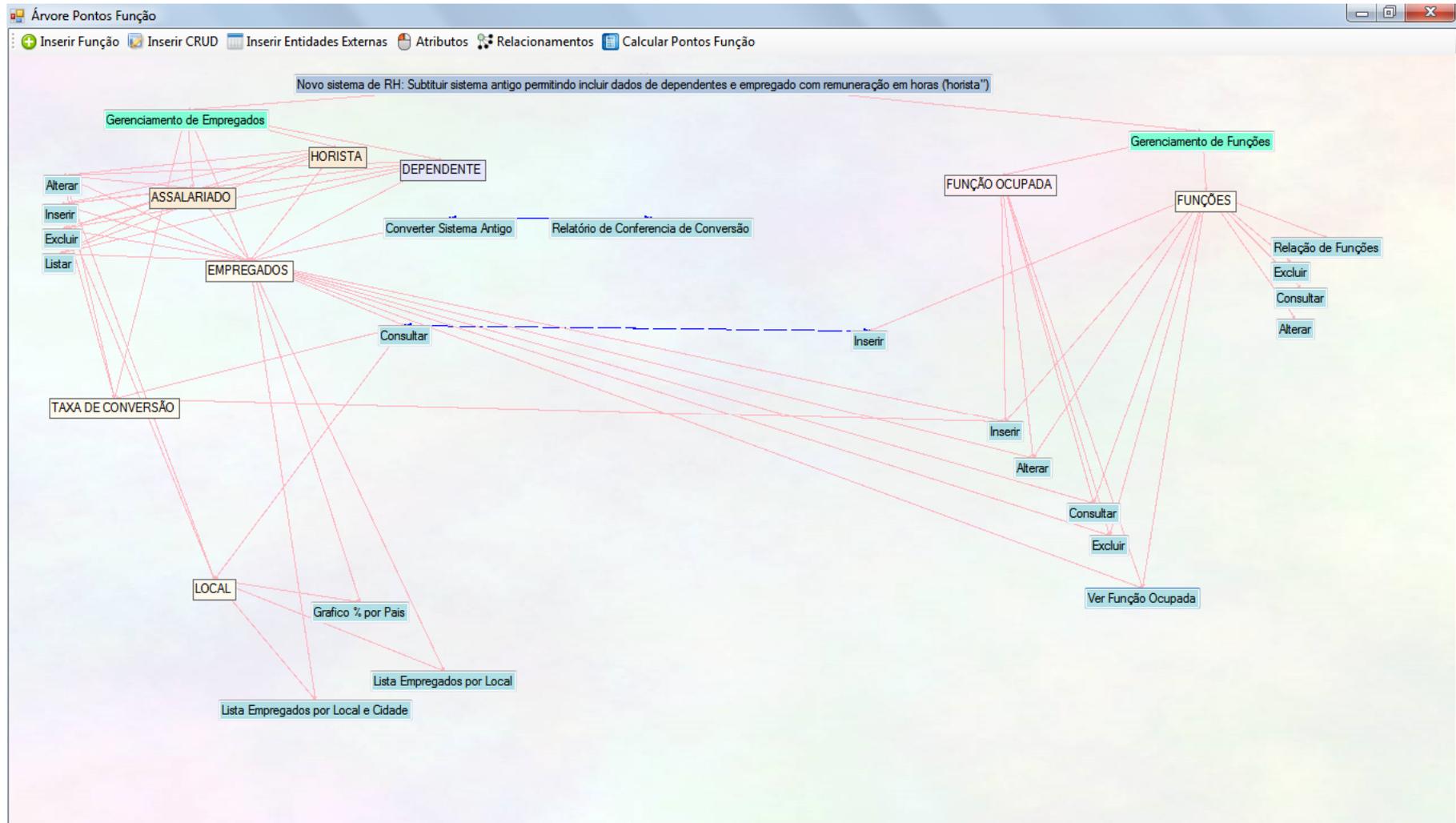
Fonte: Marcos Freitas Junior, 2015.

Figura 24 – Detalhamento da contagem de pontos de função do participante 3 via protótipo de ferramenta

Resultado da contagem: 88 pontos de função!						
Nome da Função de Transação	Tipo	ALRs/RLRs	DERs	Complexidade	Tamanho	
FUNÇÃO OCUPADA (FUNÇÃO OCUPADA)	ALI	1	3	Baixa	7	
EMPREGADO (ASSALARIADO / HORISTA / EMPREGADO)	ALI	3	10	Baixa	7	
DESCRIÇÃO FUNÇÃO (FUNÇÃO / DESCRIÇÃO FUNÇÃO)	ALI	2	5	Baixa	7	
LOCAL (LOCAL)	AIE	1	7	Baixa	5	
TAXA DE CONVERSÃO (TAXA DE CONVERSÃO)	AIE	1	3	Baixa	5	
Listar (EMPREGADO)	SE	1	3	Baixa	4	
Excluir (FUNÇÃO OCUPADA)	EE	1	4	Baixa	3	
Consultar (FUNÇÃO OCUPADA)	CE	1	8	Baixa	3	
Ver Relação (FUNÇÃO OCUPADA)	SE	1	8	Baixa	4	
Visualizar Empregados por Local (EMPREGADO)	SE	1	3	Baixa	4	
Visualizar Empr. por Local e Cidade (EMPREGADO)	SE	1	4	Baixa	4	
Visualizar Empr. por País (EMPREGADO)	SE	1	4	Baixa	4	
Alterar (DESCRIÇÃO FUNÇÃO)	EE	1	9	Baixa	3	
Excluir (DESCRIÇÃO FUNÇÃO)	EE	1	3	Baixa	3	
Consultar (DESCRIÇÃO FUNÇÃO)	CE	1	10	Baixa	3	
Ver Relação (DESCRIÇÃO FUNÇÃO)	SE	1	8	Baixa	4	
Carga Inicial (EMPREGADO)	EE	1	6	Baixa	3	
Consultar (EMPREGADO / DESCRIÇÃO FUNÇÃO)	SE	2	18	Média	5	
Excluir (EMPREGADO / DESCRIÇÃO FUNÇÃO / TAXA DE CONVERSÃO)	EE	3	3	Média	4	
Listar + Inserir + Alterar (TAXA DE CONVERSÃO / EMPREGADO / DESCRIÇÃO FUNÇÃO / LOCAL / FUNÇÃO OCUPADA)	CE	5	29	Alta	6	

Detalhar

Figura 25 – Modelagem da Árvore de pontos de função do participante 4 via protótipo de ferramenta



Fonte: Marcos Freitas Junior, 2015.

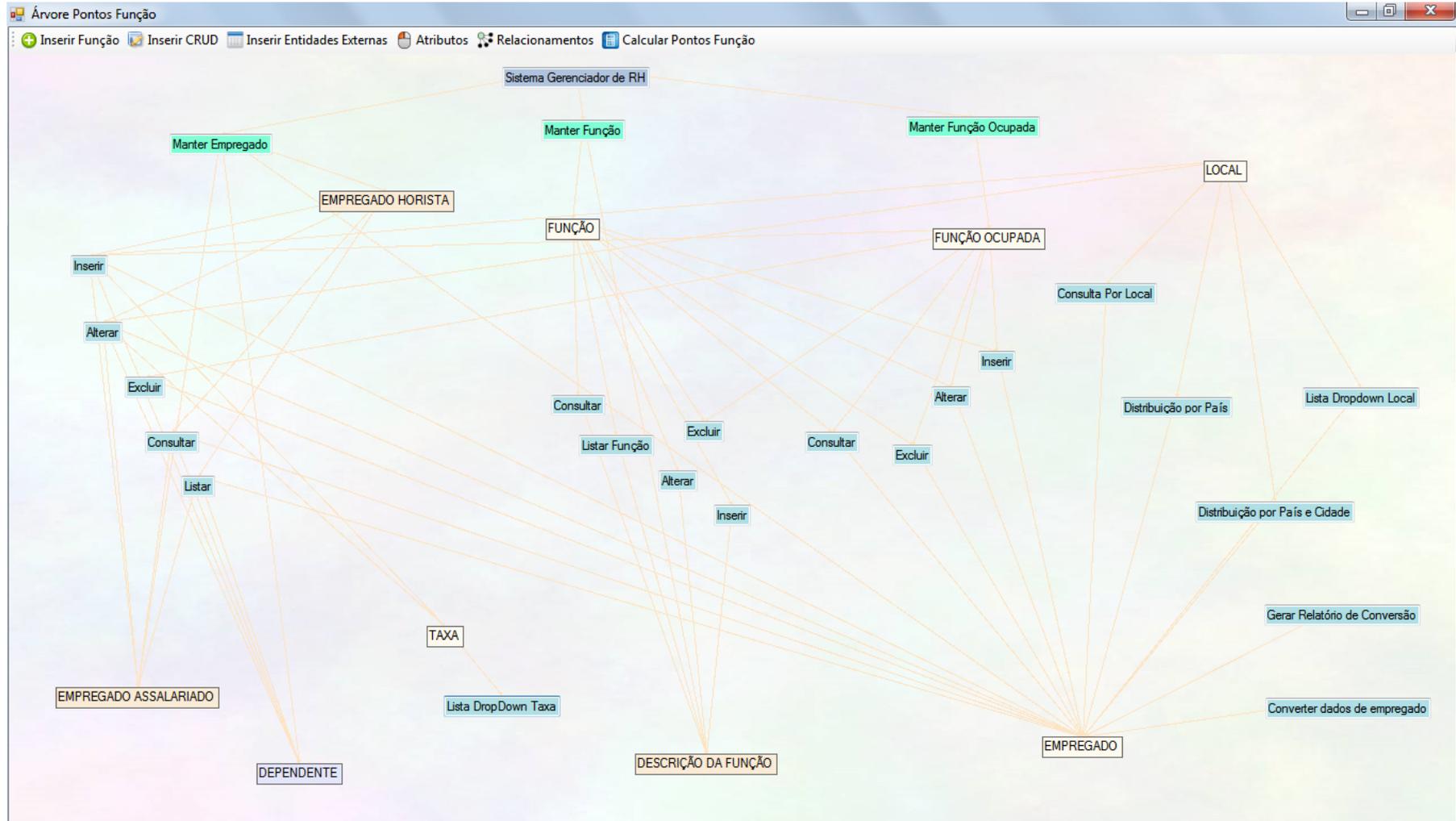
Figura 26 – Detalhamento da contagem de pontos de função do participante 4 via protótipo de ferramenta

Resultado da contagem: 110 pontos de função!						
Nome da Função de Transação	Tipo	ALRs/RLRs	DERs	Complexidade	Tamanho	
FUNÇÃO OCUPADA (FUNÇÃO OCUPADA)	ALI	1	3	Baixa	7	
EMPREGADOS (DEPENDENTE / ASSALARIADO / HORISTA)	ALI	3	8	Baixa	7	
FUNÇÕES (FUNÇÕES)	ALI	1	3	Baixa	7	
LOCAL (LOCAL)	AIE	1	7	Baixa	5	
TAXA DE CONVERSÃO (TAXA DE CONVERSÃO)	AIE	1	3	Baixa	5	
Alterar (FUNÇÕES)	EE	1	10	Baixa	3	
Excluir (FUNÇÕES)	EE	1	3	Baixa	3	
Consultar (FUNÇÕES)	CE	1	10	Baixa	3	
Relação de Funções (FUNÇÕES)	SE	1	10	Baixa	4	
Converter Sistema Antigo + Relatório de Conferencia de Conversão (EMPREGADOS)	EE	1	10	Baixa	3	
Lista Empregados por Local (EMPREGADOS / LOCAL)	SE	2	10	Média	5	
Lista Empregados por Local e Cidade (EMPREGADOS / LOCAL)	SE	2	11	Média	5	
Grafico % por Pais (EMPREGADOS / LOCAL)	CE	2	4	Baixa	3	
Inserir + Consultar (FUNÇÕES / EMPREGADOS / LOCAL / TAXA DE CONVERSÃO)	EE	4	27	Alta	6	
Alterar (EMPREGADOS / FUNÇÕES / FUNÇÃO OCUPADA)	EE	3	8	Alta	6	
Excluir (EMPREGADOS / FUNÇÕES / FUNÇÃO OCUPADA)	EE	3	4	Média	4	
Consultar (EMPREGADOS / FUNÇÕES / FUNÇÃO OCUPADA)	CE	3	8	Média	4	
Ver Função Ocupada (EMPREGADOS / FUNÇÕES / FUNÇÃO OCUPADA)	SE	3	9	Média	5	
Excluir (EMPREGADOS)	EE	1	3	Baixa	3	
Listar (EMPREGADOS)	SE	1	18	Baixa	4	
Inserir (EMPREGADOS / FUNÇÕES / TAXA DE CONVERSÃO / FUNÇÃO OCUPADA)	EE	4	7	Alta	6	
Inserir (EMPREGADOS / LOCAL / TAXA DE CONVERSÃO)	EE	3	20	Alta	6	
Alterar (EMPREGADOS / LOCAL / TAXA DE CONVERSÃO)	EE	3	20	Alta	6	

Detalhar

Fonte: Marcos Freitas Junior, 2015.

Figura 27 – Modelagem da Árvore de pontos de função do participante 5 via protótipo de ferramenta



Fonte: Marcos Freitas Junior, 2015.

Figura 28 – Detalhamento da contagem de pontos de função do participante 5 via protótipo de ferramenta

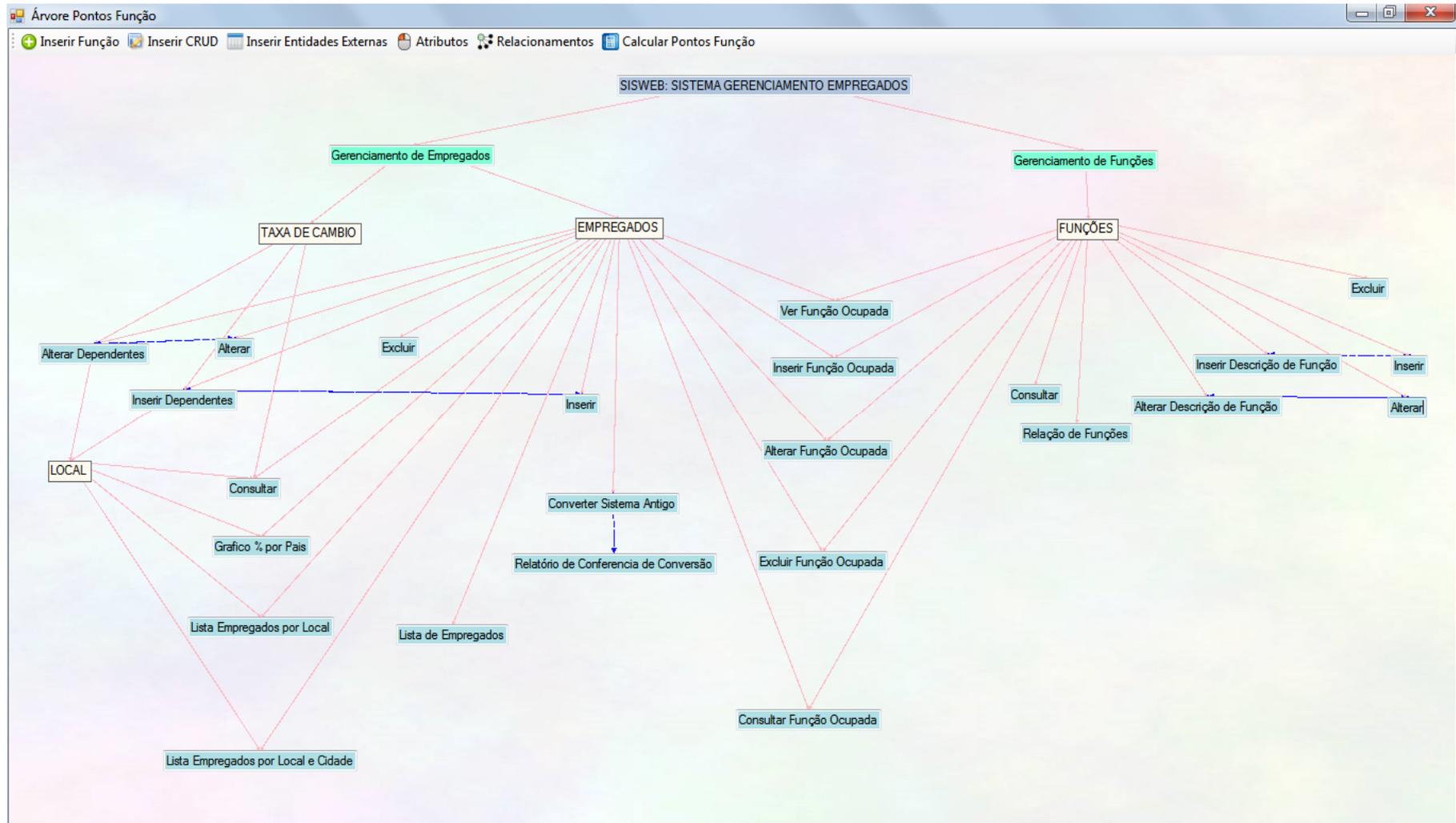
Resultado da contagem: 111 pontos de função!

Nome da Função de Transação	Tipo	ALRs/RLRs	DERs	Complexidade	Tamanho
EMPREGADO (DEPENDENTE / ASSALARIADO / HORISTA)	ALI	3	13	Baixa	7
FUNÇÃO (DESCRIÇÃO DE FUNÇÃO)	ALI	1	7	Baixa	7
FUNÇÃO OCUPADA (FUNÇÃO OCUPADA)	ALI	1	5	Baixa	7
LOCAL (LOCAL)	AIE	1	7	Baixa	5
TAXA (TAXA)	AIE	1	4	Baixa	5
Converter dados de empregado (EMPREGADO)	EE	1	9	Baixa	3
Gerar Relatório de Conversão (EMPREGADO)	SE	1	4	Baixa	4
Lista DropDown Taxa (TAXA)	CE	1	3	Baixa	3
Excluir (EMPREGADO / FUNÇÃO OCUPADA)	EE	2	3	Baixa	3
Lista Dropdown Local (EMPREGADO / LOCAL)	CE	2	5	Baixa	3
Alterar (FUNÇÃO)	EE	1	10	Baixa	3
Excluir (FUNÇÃO / FUNÇÃO OCUPADA)	EE	2	3	Baixa	3
Consultar (FUNÇÃO)	CE	1	10	Baixa	3
Listar Função (FUNÇÃO)	SE	1	9	Baixa	4
Consulta Por Local (LOCAL / EMPREGADO)	SE	2	7	Média	5
Distribuição por País e Cidade (LOCAL / EMPREGADO)	SE	2	11	Média	5
Inserir (FUNÇÃO)	EE	1	7	Baixa	3
Distribuição por País (EMPREGADO / LOCAL)	SE	2	4	Baixa	4
Inserir (FUNÇÃO OCUPADA / EMPREGADO)	EE	2	8	Média	4
Alterar (FUNÇÃO OCUPADA / EMPREGADO)	EE	2	8	Média	4
Excluir (FUNÇÃO OCUPADA / EMPREGADO)	EE	2	4	Baixa	3
Consultar (FUNÇÃO OCUPADA / EMPREGADO)	CE	2	8	Média	4
Consultar (EMPREGADO)	CE	1	19	Baixa	3
Listar (EMPREGADO)	SE	1	17	Baixa	4
Alterar (EMPREGADO / TAXA / LOCAL)	EE	3	19	Alta	6
Inserir (EMPREGADO / TAXA / LOCAL / FUNÇÃO OCUPADA)	EE	4	19	Alta	6

Detalhar

Fonte: Marcos Freitas Junior, 2015.

Figura 29 – Modelagem da Árvore de pontos de função do participante 6 via protótipo de ferramenta



Fonte: Marcos Freitas Junior, 2015.

Figura 29 – Detalhamento da contagem de pontos de função do participante 6 via protótipo de ferramenta

Resultado da contagem: 102 pontos de função!						
Nome da Função de Transação	Tipo	ALRs/RLRs	DERs	Complexidade	Tamanho	
EMPREGADOS (EMPREGADOS)	ALI	1	3	Baixa	7	
FUNÇÕES (FUNÇÕES)	ALI	1	3	Baixa	7	
LOCAL (LOCAL)	AIE	1	7	Baixa	5	
TAXA DE CAMBIO (TAXA DE CAMBIO)	AIE	1	3	Baixa	5	
Excluir (EMPREGADOS)	EE	1	3	Baixa	3	
Lista de Empregados (EMPREGADOS)	SE	1	18	Baixa	4	
Excluir (FUNÇÕES)	EE	1	3	Baixa	3	
Consultar (FUNÇÕES)	CE	1	10	Baixa	3	
Relação de Funções (FUNÇÕES)	SE	1	10	Baixa	4	
Inserir + Inserir Descrição de Função (FUNÇÕES)	EE	1	9	Baixa	3	
Alterar + Alterar Descrição de Função (FUNÇÕES)	EE	1	10	Baixa	3	
Converter Sistema Antigo + Relatório de Conferencia de Conversão (EMPREGADOS)	EE	1	10	Baixa	3	
Inserir Função Ocupada (EMPREGADOS / FUNÇÕES)	EE	2	8	Média	4	
Alterar Função Ocupada (EMPREGADOS / FUNÇÕES)	EE	2	8	Média	4	
Excluir Função Ocupada (EMPREGADOS / FUNÇÕES)	EE	2	4	Baixa	3	
Consultar Função Ocupada (EMPREGADOS / FUNÇÕES)	CE	2	8	Média	4	
Ver Função Ocupada (EMPREGADOS / FUNÇÕES)	SE	2	9	Média	5	
Lista Empregados por Local (EMPREGADOS / LOCAL)	SE	2	10	Média	5	
Lista Empregados por Local e Cidade (EMPREGADOS / LOCAL)	SE	2	11	Média	5	
Gráfico % por País (EMPREGADOS / LOCAL)	SE	2	4	Baixa	4	
Inserir + Inserir Dependentes (EMPREGADOS / LOCAL / TAXA DE CAMBIO)	EE	3	20	Alta	6	
Alterar + Alterar Dependentes (EMPREGADOS / LOCAL / TAXA DE CAMBIO)	EE	3	20	Alta	6	
Consultar (EMPREGADOS / LOCAL / TAXA DE CAMBIO)	CE	3	20	Alta	6	

Detalhar

Fonte: Marcos Freitas Junior, 2015.

Apêndice G – E-mail enviado aos respondentes para realização de experimento

A seguir, é apresentado o conteúdo do e-mail enviado aos participantes da primeira organização, com o detalhamento dos procedimentos para elaboração da Árvore de pontos de função, entrega do artefato elaborado e instruções gerais.

Pessoal, Bom dia!

O Marcos Freitas realizou um trabalho de pesquisa de Mestrado, cujo resultado foi a criação de uma ferramenta que propõe a medição funcional de sistemas a partir de informações declaradas pelo usuário. A (nome omitido) está colaborando com a Prova de Conceito deste trabalho, que contempla o uso desta ferramenta para medir o escopo funcional declarado em um estudo de caso. Para tanto, pedimos a gentileza e colaboração para usar a ferramenta a fim de que ela calcule o tamanho deste escopo do projeto.

Este e-mail acompanha um arquivo.ZIP em anexo, contendo os seguintes itens para realização da tarefa:

- *Documento Apostila_Exercicios_APF_2011-03-17.pdf: Contém o estudo de caso a ser medido: Descrição dos Requisitos: Páginas de 3 a 7; Modelagem de Dados: Pagina 14; Protótipo: Páginas de 21 a 65.*
- *ArvorePontosFunção.exe: Contém o executável da ferramenta a ser usada para medição do escopo do estudo de caso.*
- *Documento contendo a Explicação de como usar a ferramenta na elaboração do artefato "Árvore de pontos de função";*

PROCEDIMENTOS PARA ENTREGA:

As instruções estão no documento descrito acima. Os procedimentos também podem ser consultados via vídeo. Basicamente, o procedimento inclui zipar a pasta (que contém o banco de dados), e enviá-la ao Marcos Freitas via <mailto:marcos.freitas@usp.br>.

- *Data limite para entrega: 30/10 (sexta), no máximo 31/10 (sábado).*

IMPORTANTE:

Para medição do escopo, deve-se usar como insumo apenas as orientações para uso da ferramenta e as informações do estudo de caso. A elaboração da Árvore de pontos de função na ferramenta deve ser realizada no horário de trabalho e as horas lançadas no timesheet.

Muito obrigada!

O conteúdo do e-mail enviado aos participantes da segunda organização, com o detalhamento dos procedimentos para elaboração da Árvore de pontos de função, entrega do artefato elaborado e instruções gerais é apresentado a seguir.

Prezados, Boa noite!

Agradeço o interesse na pesquisa que desenvolvo para a definição de uma abordagem automatizável para contagem de pontos de função. É parte desse e-mail estudo de caso a ser medido.

PROCEDIMENTOS PARA ENTREGA:

As instruções para a elaboração da Árvore de pontos de função estão descritas no documento acima. Os procedimentos também podem ser consultados via vídeo. Basicamente, o procedimento inclui elaborar manualmente a Árvore de pontos de função e enviar para <mailto:marcos.freitas@usp.br>.

IMPORTANTE:

Para medição do escopo, deve-se usar como insumo apenas as informações do estudo de caso. O tempo para a realização dessa atividade deve ser controlado e informado em conjunto com a entrega via e-mail.

Abraços!