



Pós-Graduação em Ciência da Computação

***FELIPE DE OLIVEIRA BARBALHO***

**Uma Abordagem Comparativa entre duas Técnicas de  
Medição de Software**

**Dissertação de Mestrado Profissional**



Universidade Federal de Pernambuco  
posgraduacao@cin.ufpe.br  
[www.cin.ufpe.br/~posgraduacao](http://www.cin.ufpe.br/~posgraduacao)

RECIFE  
2014

***Felipe de Oliveira Barbalho***

***Uma Abordagem Comparativa entre duas Técnicas  
de Medição de Software***

Este trabalho foi apresentado à Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Mestre Profissional em Ciência da Computação.

***ORIENTADOR: Prof. Dr. Alexandre Marcos Lins de Vasconcelos***

***RECIFE  
2014***

Catálogo na fonte  
Bibliotecária Monick Raquel Silvestre da S. Portes, CRB4-1217

B228a Barbalho, Felipe de Oliveira  
Uma abordagem comparativa entre duas técnicas de medição de software /  
Felipe de Oliveira Barbalho. – 2014.  
102 f.: il., fig., tab.

Orientador: Alexandre Marcos Lins de Vasconcelos.  
Dissertação (Mestrado) – Universidade Federal de Pernambuco. CIn,  
Ciência da Computação, Recife, 2014.  
Inclui referências e apêndice.

1. Engenharia de software. 2. Gerenciamento de projetos. 3. Medição de  
software. I. Vasconcelos, Alexandre Marcos Lins de (orientador). II. Título.

005.1

CDD (23. ed.)

UFPE- MEI 2016-124

Dissertação de Mestrado Profissional apresentada por **Felipe de Oliveira Barbalho** à Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, sob o título, “**Uma Abordagem Comparativa entre duas Técnicas de Medição de Software**”, orientada pelo Professor Alexandre Marcos Lins de Vasconcelos e aprovada pela Banca Examinadora formada pelos professores:

---

Prof. Hermano Perrelli de Moura  
Centro de Informática / UFPE

---

Prof. Cristine Martins Gomes de Gusmão  
Centro de Tecnologia e Geociências / UFPE

---

Prof. Alexandre Marcos Lins de Vasconcelos  
Centro de Informática / UFPE

Visto e permitida a impressão.  
Recife, 28 de abril de 2014.

---

Prof<sup>a</sup>. EDNA NATIVIDADE DA SILVA BARROS  
Coordenadora da Pós-Graduação em Ciência da Computação do  
Centro de Informática da Universidade Federal de Pernambuco.

## **Agradecimentos**

A princípio, gostaria de agradecer ao meu pai Marcilio, por todo esforço e dedicação para conseguir me dar uma educação com qualidade em toda minha vida. Sem ele eu não teria chegado aqui.

A minha mãe Fátima por todos os conselhos e apoio, me incentivando ao crescimento profissional e educacional, e tendo a certeza que sempre pude contar com ela em qualquer momento.

A Malu por todo o companheirismo durante esse meu período de dedicação ao trabalho, sempre me apoiando e estando ao meu lado, com sua presença bastante confortante nos momentos difíceis.

A minha irmã Milena, e meus amigos que me ajudaram de alguma forma nessa jornada, se privando de momentos pessoais para me apoiar, e compreendendo minha ausência quando foi necessário.

Ao meu amigo Carlos Diego, sempre presente desde o início do mestrado, e contribuindo com dedicação e boa vontade na revisão do trabalho.

Ao Prof. Dr. Alexandre Marcos Lins de Vasconcelos por toda a solicitude e empenho, me orientando nesta pesquisa, apostando e acreditando em minha capacidade como profissional e pesquisador.

## Resumo

Há uma tendência das organizações que atuam no desenvolvimento e manutenção de software em utilizarem Pontos de Função como medida padrão em seus processos de medição. Atualmente a análise de pontos de função é a técnica mais utilizada entre as organizações brasileiras. Por outro lado, o COSMIC é uma técnica de medição considerada como a segunda geração em método de medição funcional de projetos de desenvolvimento e manutenção de software, sendo precisa e abrangente para medir software e estimar esforço de desenvolvimento, despertando o interesse do mercado de desenvolvimento de software por essa métrica.

Quando considerada a migração das bases históricas de medição de software das organizações para uma nova unidade de medida, há como consequência um alto custo para se medir os projetos legados. Nesse sentido, o objetivo desse trabalho é propor uma alternativa para migração de bases históricas de Pontos de Função para o COSMIC. A abordagem propõe uma fórmula de conversão, o qual foi encontrada a partir da medição de quatro projetos de desenvolvimento de software analisados nessas duas técnicas, para atualizar seus dados sem o custo de refazer toda sua base histórica.

**Palavras-chave:** Gerenciamento de Projetos. Métricas. Medição de Software. APF. Ponto de Função. COSMIC.

## Abstract

There is a tendency in software development and maintenance organizations in using Function Points as a standard measure in their measurement processes. Currently the analysis of function points is the most used technique among Brazilian organizations. On the other hand, COSMIC is a measurement technique considered as the second-generation method of functional measurement for software development and maintenance projects, with an accurate and comprehensive approach to measure and estimate software development effort, arousing interest of software development market for this metric.

When organizations are considering the migration of their historical measure basis for a new measurement unit, there is a consequent high cost to measure the legacy projects. In this sense, the purpose of this work is to propose an alternative to migrate historical bases of Function Points to COSMIC. The approach proposes a conversion factor, which was found from the measurement of four software development projects analyzed in these two techniques to update its data without the cost of redoing the entire historical basis.

**Keywords:** Project Management. Metrics. Software Measurement. FPA. Function Point. COSMIC.

## Lista de Siglas

AIE – Arquivo de Interface Externa;

ALI – Arquivo Lógico Interno;

APF - Análise de Pontos de Função;

CE – Consulta Externa;

CFP – *Cosmic Function Point*;

CMMI - *Capability Maturity Model Integration for Aquisition*;

CPM – Manual de Práticas de Contagens do IFPUG;

EE – Entrada Externa;

IFPUG - *International Function Points Group User*;

ISMA - *International Software Measurement & Analysis Conference*

ISO – *International Standards Organization*;

MIS - Management Information System;

MPS.BR - Melhoria de Processo do Software Brasileiro;

NESMA – *Netherlands Software Metrics Users Association*;

PE – Processo Elementar;

PF – Ponto de Função;

RFU – Requisitos Funcionais do Usuário;

RUP – *IBM Rational Unified Process*

SE – Saída Externa;

SISP – Sistema de Administração dos Recursos de Tecnologia da Informação;

UFP - *Unadjusted Function Points*.

## Lista de Tabelas

Tabela 2.1 – Tabela de complexidade funcional dos ALI e AIE .....	36
Tabela 2.2 – Tabela de contribuição dos pontos de função das funções do tipo dado.....	36
Tabela 2.3 – Tabela de complexidade para Entradas Externas (EE's) .....	40
Tabela 2.4 – Tabela de complexidade para Sidas Externas (SE's) e Consultas Externas (CE's). 41	
Tabela 2.5 – Tabela de contribuição dos pontos de função das funções do tipo transação .....	41
Tabela 2.6 – Tabela de características da APF e do COSMIC .....	67
Tabela 3.1a – Tabela de comparação entre as medições .....	71
Tabela 3.1b – Tabela de comparação entre as medições .....	72
Tabela 3.1c – Tabela de comparação entre as medições .....	73
Tabela 3.1d – Tabela de comparação entre as medições .....	74
Tabela 3.2a – Tabela de comparação entre as medições .....	76
Tabela 3.2b – Tabela de comparação entre as medições .....	77
Tabela 3.3a – Tabela de comparação entre as medições .....	79
Tabela 3.3b – Tabela de comparação entre as medições .....	80
Tabela 3.3c – Tabela de comparação entre as medições .....	81
Tabela 3.4a – Tabela de comparação entre as medições .....	83
Tabela 3.4b – Tabela de comparação entre as medições .....	84
Tabela 3.4c – Tabela de comparação entre as medições .....	85
Tabela 3.5 – Tabela de formação do fator de conversão.....	86
Tabela 4.1 – Tabela comparativa dos resultados das fórmulas de conversão.....	92

## Lista de Figuras

Figura 2.1 - As Quatro Principais Razões para Medição de Software segundo Watts Humphrey. .....	19
Figura 2.2 – Visão Geral do Processo de Contagem de Pontos de Função .....	31
Figura 2.3 – Fronteira da Aplicação.....	34
Figura 2.4 – Estrutura da Técnica COSMIC.....	56
Figura 2.5 – Requisitos Funcionais do Usuário por Camadas.....	57
Figura 2.6 – Processo de Determinação de uma Estratégia de Medição.....	58
Figura 2.7 – Método geral do processo de mapeamento de software COSMIC.....	59
Figura 2.8 – Método geral do processo de medição de software COSMIC .....	61
Figura 2.9 – Movimentação de Dados.....	62

# Sumário

<b>1 Introdução.....</b>	<b>11</b>
1.1. Objetivos.....	13
1.2. Metodologia.....	14
1.3. Contribuição .....	16
1.4. Estrutura do documento.....	16
<b>2 Referencial teórico.....</b>	<b>18</b>
2.1. Métricas de Software .....	18
2.1.1. Introdução .....	18
2.1.2. Métricas como Ferramenta de Apoio a Gestão de Projetos .....	20
2.1.3. Implantação de um Programa de Medição .....	22
2.1.4. Padrão ISO de Medição Funcional.....	27
2.2. Análise de Pontos de Função – APF.....	28
2.2.1. Definições, Conceitos e Regras.....	28
2.2.2. Projetos de Melhoria Utilizando a APF .....	43
2.2.3. Pontos de Função no Brasil.....	45
2.2.4. APF na Contratação de Software .....	47
2.3. Common Software Measurement International Consortium – COSMIC.....	52
2.3.1. Definições, Conceitos e Regras.....	52
2.3.2. Projetos de Melhoria Utilizando COSMIC .....	63
2.3.3. Aderência a Alguns Tipos de Sistema.....	64
2.3.4. Vantagens da aplicação da técnica .....	65
2.4. Considerações finais do capítulo .....	<b>66</b>
<b>3 Estudo de Caso .....</b>	<b>68</b>
3.1. Organização selecionada do estudo de caso .....	68
3.2. Aplicabilidade da Pesquisa.....	68
3.3. Coleta de Dados.....	69
3.3.1. Sistema de Gestão de Aquisições.....	71
3.3.2. Sistema de Controle de Viagens.....	75
3.3.3. Sistema de Gerenciamento de Limites Elétricos (GERLIM) .....	78
3.3.4. Sistema de Gerenciamento de Informações do Programa Mensal de Operação do Setor Elétrico (SGI-PMO).....	82
3.4. Formação da Fórmula de conversão .....	86
3.5. Considerações finais do capítulo .....	87
<b>4 Conclusão.....</b>	<b>88</b>
4.1. Considerações Finais .....	88
4.2. Limitações e Ameaças à Validade.....	90
4.3. Trabalhos Futuros .....	92
<b>Referências.....</b>	<b>93</b>
<b>Apêndice.....</b>	<b>100</b>

# 1 Introdução

---

Os avanços tecnológicos e a alta competitividade do mercado estão continuamente aumentando a demanda por softwares cada vez melhores e que sejam produzidos em projetos aderentes aos custos e prazos planejados (BALDASSARE et al., 2009). Mesmo diante de todos os avanços tecnológicos, realizar projetos aderentes aos seus planos ainda é um desafio para grande parte das organizações de software. Buscando aprimorar suas práticas de Engenharia de Software e, conseqüentemente, desenvolver produtos de melhor qualidade em projetos conduzidos de acordo com seus planos, as organizações têm mostrado um crescente interesse em programas de melhoria de processos, contexto no qual a medição de software tem papel fundamental (WEBER; LAYMAN, 2002; CANFORA et al., 2004; DUMKE et al., 2006).

Aumentar a qualidade, desempenho e produtividade são os objetivos chave de qualquer organização que desenvolve software (MENS; DEMEYER, 2001) e o uso de métricas está diretamente ligado ao controle desses processos. Com a crescente exigência pela melhoria da qualidade de software, através do desenvolvimento dentro dos prazos e custos estimados e atendendo as expectativas do cliente, é impossível não enxergar as técnicas de medição de software como alavanca para melhoria da qualidade do processo de desenvolvimento de software.

A mensuração de software está se consolidando como uma prática importante entre as empresas, visando suportar suas iniciativas de melhoria no processo de software, pois pode servir tanto como fonte de informação para o monitoramento da situação atual do processo de software, como para a identificação de desvios na execução do processo (FRANÇA et al., 1998).

Métricas de diferentes aspectos do desenvolvimento podem ajudar a determinar o progresso do projeto, e podem ser usadas no gerenciamento para prover base para a tomada de decisão. As métricas corretas, usadas de maneira certa, são essenciais para o sucesso de um projeto (PERKINS et al., 2003).

Pressman afirma que medição faz parte de uma série de “medicações” que pode ajudar a curar a aflição do software, ela oferece benefícios em nível estratégico, em nível de projeto e em nível técnico (PRESSMAN, 1995).

Alguns questionamentos ainda são bastante discutidos sobre as atuais práticas na área de métricas de software, como por exemplo:

- Medimos de forma adequada?
- Qual técnica de medição melhor atende a um determinado tipo de software?
- Qual técnica de medição melhor atende a um determinado tipo de projeto (Desenvolvimento, Melhoria, Aplicação)?
- Como medir os requisitos não-funcionais?
- Como obter dados de medições do processo de desenvolvimento de software?
- Como utilizar as métricas de uma forma mais efetiva no gerenciamento do processo de software?
- Como criar uma base histórica de minhas medições e quais indicadores consigo extrair dela?

As métricas são ferramentas essenciais ao gerenciamento de projetos de software. A escolha das métricas está intimamente associada às estratégias e objetivos da organização, e vai depender do estágio de maturidade em que a mesma se encontra. As métricas coletadas devem prover informações que ajudem na tomada de decisões de acordo com os objetivos e estratégias da organização. Alguns desses objetivos podem ser: melhorar a qualidade do planejamento do projeto, reduzir os custos de retrabalho no processo, melhorar a qualidade do processo de desenvolvimento, melhorar a qualidade do produto resultante, reduzir os custos de falha, aumentar a produtividade do desenvolvimento, aperfeiçoar continuamente os métodos de gestão do projeto, etc. Por essa razão, o uso de métricas tem se tornado uma grande vantagem estratégica. Inserido nesse contexto da necessidade de um processo de medição bem definido dentro do processo de desenvolvimento de software das organizações, este trabalho selecionou duas técnicas de medição funcional para serem estudadas: A Análise de Pontos de Função (APF) e o COSMIC.

Em 1986, uma Pesquisa do *Quality Assurance Institute* mostrou que PF é a melhor métrica para o estabelecimento de medições de qualidade e produtividade de projetos de software (PERRY, 1986). Em 1993, pontos de função se tornou a métrica mais utilizada

e estudada na Engenharia de Software (JONES, 1994). Atualmente, a métrica de PF continua sendo a mais utilizada na indústria de software, como métrica padrão na definição de indicadores, como insumo para derivação de estimativas de prazo, custo e esforço e no estabelecimento de contratos de software. Desde 2002 a Análise de Pontos de Função foi padronizada pela norma ISO/IEC 20.926 como uma técnica de medição funcional, aderente à norma ISO/IEC 14.143 (DEKKERS, 2003).

O COSMIC, por outro lado, é considerado por seus criadores como a segunda geração em técnica de medição funcional, sendo uma métrica mais precisa e abrangente para medir software e estimar esforço de desenvolvimento. Além de ser aplicável a vários domínios de software, é aceito em um crescente número de organizações dos setores públicos e privados em todo o mundo. A técnica foi desenhada para garantir a sua total conformidade com a norma ISO/IEC 14.143-1 (COSMIC, 2007).

No Brasil, diversas instituições públicas e privadas têm utilizado a métrica Ponto de Função devido aos diversos benefícios desta métrica, e por recomendações de órgãos reguladores para sua utilização nos contratos de serviços de desenvolvimento e manutenção de software em organizações governamentais, o que tornou a APF unânime no mercado de medição nacional (SISP, 2012). O fato das organizações já terem uma métrica consolidada no mercado e com dados de *benchmarking* para apoio a decisão, junto com o alto custo de migração para uma outra técnica de medição, inviabiliza o processo de adesão a uma nova técnica como o COSMIC.

Esse trabalho realiza uma pesquisa com as duas técnicas de medição funcional, para entender quais suas características e seu processo de medição, com o objetivo de definir uma fórmula de conversão da técnica APF para a técnica COSMIC, construída a partir do estudo de caso em projetos de desenvolvimento de softwares aplicativos de negócio, com o objetivo de diminuir o custo de migração de uma base histórica entre essas medidas, e conseqüentemente diminuir também a resistência do mercado a uma nova técnica.

## **1.1. Objetivos**

Este trabalho tem como objetivo geral contribuir com o mercado de desenvolvimento de software, apresentando duas técnicas de medição a partir de um estudo comparativo entre suas definições, regras e características, possibilitando um apoio as organizações na escolha de uma métrica padrão.

Considerando a análise de pontos de função como a técnica de medição mais utilizada pelo mercado de desenvolvimento de software, e sabendo do alto custo que as organizações teriam para migrar sua unidade de medida de software padrão para o COSMIC, o objetivo específico desse trabalho é encontrar uma fórmula de conversão da técnica APF para a técnica COSMIC a ser utilizada na migração de suas bases históricas de medição, a partir da medição de quatro projetos de desenvolvimento de software analisados nas duas técnicas estudadas.

## **1.2. Metodologia**

A metodologia escolhida para a execução desse trabalho foi o estudo de caso, considerando a necessidade de uma aplicação prática de cada técnica estudada para obter melhores resultados na pesquisa comparativa de suas características, e para a descoberta coerente de uma fórmula de conversão entre as técnicas APF e COSMIC.

Para Yin, estudo de caso é uma investigação empírica, um método que abrange tudo – planejamento, técnicas de coleta de dados e análise dos mesmos (YIN, 2010). Na perspectiva de Merriam (1988, apud André 2005), o conhecimento gerado a partir do estudo de caso é diferente do conhecimento gerado a partir de outras pesquisas porque é mais concreto, mais contextualizado, mais voltado para a interpretação do leitor e baseado em populações de referência determinadas pelo leitor.

Nos estudos de caso, a riqueza do fenômeno e a extensão do contexto da vida real exigem que sejam enfrentadas algumas dificuldades técnicas. Para solucionar tal ameaça, é necessário usar múltiplas fontes de evidência, de forma que os dados converjam de modo triangular para garantir a confiabilidade dos dados (YIN, 2010).

A escolha do método de estudo de caso ao invés do método de experimento, se dá pelo fato de que experimentos são realizados quando o pesquisador pode manipular o comportamento de forma direta, precisa, e sistemática, já em estudos de caso não há possibilidade de controle de variáveis (YIN, 2010).

O estudo de caso pode-se utilizar de evidência quantitativa. Não existe distinção em estudos de caso quando se utiliza um método quantitativo ou qualitativo. Em outras palavras, o método de estudo de caso não é apenas uma forma de “pesquisa qualitativa”, porque pode utilizar também evidências quantitativas. O método escolhido para esse trabalho utiliza evidências quantitativas (YIN, 2010).

Na pesquisa de estudos de caso podem ser incluídos tanto um estudo de caso único quanto estudos de casos múltiplos (YIN, 2010). Neste trabalho foi convencionado o estudo de caso múltiplo, considerando que um único caso não é o suficiente para alcançar o resultado esperado na pesquisa científica, e que a utilização de múltiplas fontes consolida uma análise mais precisa.

A designação do número de replicações depende da certeza que o pesquisador deseja ter sobre os resultados dos casos múltiplos (a maior certeza reside no maior número de casos). Porém, o número de replicações podem se restringir devido à disponibilidade da amostra. E caso sejam encontrados resultados não esperados, mais casos adicionais poderão ser requeridos para que sejam formuladas teorias rivais (YIN, 2010).

O processo de pesquisa desse trabalho foi dividido nas seguintes etapas:

- **Definição do tema de pesquisa:** A partir do objetivo do trabalho foi realizada uma pesquisa pelas palavras chaves associadas ao assunto de interesse.
- **Revisão da literatura:** Foram selecionados fontes na literatura e trabalhos relacionados às técnicas de medição funcional APF e COSMIC, com o objetivo de melhorar a compreensão do estudo realizado e tornar seu resultado mais confiável.
- **Definição da organização e projetos que serão utilizados na pesquisa:** Foram selecionados quatro projetos de desenvolvimento de uma única organização, descrita na Seção 3.1. Como critério para seleção dos projetos foram consideradas as seguintes características: o tipo de sistema (Softwares Aplicativos de Negócio), o tipo de projeto (Desenvolvimento), e o tamanho estimado de 300 PF. O critério de seleção teve como objetivo a aderência desses sistemas para realização das contagens nas duas técnicas de medição escolhida. O tamanho aproximado para a escolha dos sistemas foi definido a partir de uma quantidade de funcionalidades das medições considerada como suficiente para evidenciar as características da comparação entre as técnicas, e o cumprimento do prazo planejado para o trabalho.
- **Medição dos projetos selecionados:** Foram executados oito projetos de medição, onde cada um dos quatro sistemas selecionados, já entregue e homologado pelo cliente, teve seu tamanho medido a partir de um projeto

utilizando a técnica de Análise de Pontos de Função, e outro com a técnica COSMIC. Todas as medições foram executadas com base no processo, definido em acordo com as equipes de desenvolvimento dos projetos, documentado no Apêndice A.

- **Análise do resultado:** A partir dos oito projetos de medição realizados na etapa anterior foi definido uma fórmula de conversão da técnica de APF para a técnica COSMIC. Após a execução dos oito projetos de medição foi possível concluir características, problemas e benefícios das duas técnicas analisadas.

### 1.3. Contribuição

Um grande número de organizações do mercado de desenvolvimento e manutenção de software utiliza o ponto de função como medida padrão em seus processos de medição de software. Atualmente APF é a técnica mais utilizada por esse mercado.

A técnica COSMIC é reconhecida por seus criadores como a segunda geração de medição funcional de software, acreditando ser bastante promissora e com grandes vantagens em relação a outras técnicas.

Este trabalho irá contribuir com as organizações que utilizam um processo de medição em seus processos de desenvolvimento de software, apoiando essas organizações a tomarem uma decisão mais coerente com a realidade de seus projetos, na escolha de qual a melhor técnica a ser utilizada como padrão. As organizações que já utilizam pontos de função como métrica padrão em suas bases históricas de medição de software, terão a possibilidade de migrar as medições de seus projetos para a técnica COSMIC a partir da fórmula de conversão encontrada nessa pesquisa, evitando o alto custo de recontagem de todos os seus projetos para a nova técnica.

### 1.4. Estrutura do Documento

Além deste capítulo introdutório que apresenta as considerações iniciais, metodologia, objetivo e contribuição, este trabalho é composto por mais três capítulos:

- **Capítulo 2 – Referencial Teórico:** apresenta o referencial teórico sobre o assunto abordado, para o entendimento do trabalho realizado, falando sobre métricas de software e sua utilização como ferramenta de apoio a gestão de

projetos, implantação de um programa de medição, e os conceitos, definições, regras e algumas particularidades das técnicas APF e COSMIC.

- **Capítulo 3 – Estudo de caso:** apresenta a organização, os projetos selecionados para o estudo de caso, o processo definido para as medições dos projetos, suas execuções, e o resultado encontrado.
- **Capítulo 4 – Conclusão:** apresenta uma análise das limitações e ameaças à validade da pesquisa, assim como as lições aprendidas e as recomendações para trabalhos futuros, além das conclusões do trabalho.

O trabalho também contém um conjunto de apêndices relacionados aos estudos de caso.

# 2 Referencial Teórico

---

Este capítulo apresenta os resultados de uma revisão da literatura conduzida de maneira não sistemática (ad hoc), que será utilizada como fundamentação teórica para o melhor entendimento deste trabalho, de modo que tais conceitos sejam suficientes para a contextualização da pesquisa e a correta interpretação dos resultados aqui apresentados.

## 2.1. Métricas de Software

### 2.1.1. Introdução

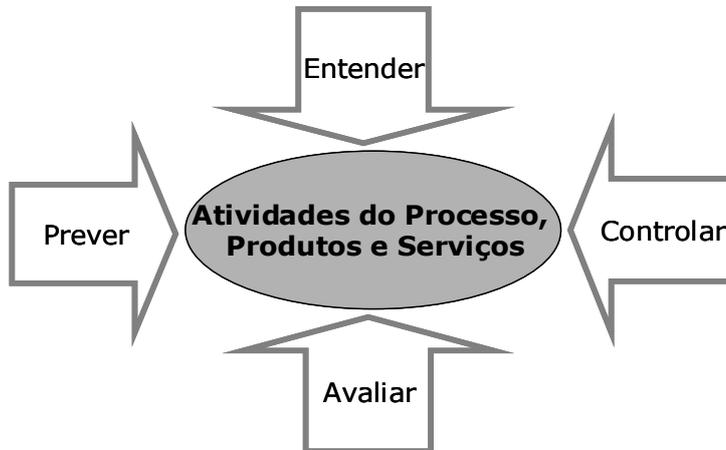
Com o avanço da Tecnologia da Informação (TI), as pessoas e as organizações necessitam cada vez mais de sistemas capazes de fornecer informações com qualidade (FERREIRA; NUNES, 2008) e à medida que o processo de desenvolvimento de Sistemas de Informação (SI) amadurece, a medição passa a desempenhar um papel cada vez mais importante no entendimento e controle das práticas e produtos relacionados a este desenvolvimento (KITCHENHAM et al., 1995).

Para alcançar níveis cada vez mais altos de qualidade, torna-se necessário melhorar o ciclo de desenvolvimento do software e, para tornar isto possível, dados quantitativos que descrevam a realidade do processo precisam ser obtidos e devidamente analisados. A medição de software auxilia no monitoramento do processo de desenvolvimento de software e da qualidade do produto, proporcionando sua avaliação contínua e possibilitando ajustes caso seja identificado algum desvio em função das tendências detectadas (ROCHA et al., 2001).

Na visão de Florac e Carleton, a medição de software contribui para melhor entendimento dos processos que fazem parte de um projeto, ajuda a estabelecer pontos importantes no projeto, contribui na análise do status do projeto de acordo com o que foi planejado e permite avaliar até que ponto o processo que está sendo executado se encontra de acordo com os padrões especificados (FLORAC; CARLETON, 1999).

Uma abordagem clássica para destacar a importância da medição de software pode ser obtida através da visão de Humphrey apud (FEITOSA 2004), que apresenta quatro

principais objetivos da medição de software, conforme Figura 2.1: entender, controlar, prever e avaliar o processo, produtos e serviços.



**Figura 2.1 - As Quatro Principais Razões para Medição de Software, segundo Watts Humphrey (FEITOSA 2004)**

**Entender:** viabiliza a aquisição de aprendizado sobre aspectos do produto ou atividade do processo que está sendo medido, facilitando o suporte à tomada de decisões.

**Avaliar:** aplica-se à verificação se os produtos ou processos envolvidos na medição atendem os critérios de aceitação, focando na melhoria de processos.

**Controlar:** facilita o gerenciamento do desenvolvimento do software, e o acompanhamento dos projetos.

**Prever:** corresponde à utilização de métricas para elaboração de estimativas, construção de médias e tendências.

A mensuração do tamanho do software na gestão de projetos está vinculada à necessidade de avaliar e medir resultados, como também de conhecer melhor o patrimônio de software, gerar indicadores para tomada de decisão, avaliar o impacto da introdução de novas tecnologias e obter e/ou melhorar estimativas de prazo, custo e recursos gerando expectativas mais realistas para o usuário (SIMÕES, 1999).

O primeiro passo para o estabelecimento de uma política de mensuração é a identificação adequada das medidas a serem coletadas. Essa definição deve ser feita com base em critérios bem fundamentados, devido ao grande número de opções possíveis e ao custo envolvido na coleta de cada informação (BORGES 2003).

Instituições como IFPUG (International Function Point Users Group), ISO (International Organization for Standardization) e COSMIC (Common Software Measurement International Consortium) desenvolveram técnicas, normas e práticas para

a avaliação quantitativa das funcionalidades de software e, entre elas, a Análise de Pontos de Função (APF) – criada em 1979 por Allan Albrecht, da IBM – tem sido a mais utilizada e referenciada em termos de contagem de requisitos funcionais (GARMUS; HERRON, 2001). Trata-se de uma medida de tamanho funcional de projetos de software que considera as funcionalidades implementadas, independentemente da metodologia e tecnologia utilizadas, sob o ponto de vista do usuário.

### **2.1.2. Métricas como Ferramenta de Apoio a Gestão de Projetos**

Quando se pode medir aquilo sobre o qual se está falando e expressá-lo em números, sabe-se alguma coisa sobre o mesmo. Mas quando não se pode medi-lo, quando não se pode expressá-lo em números, o conhecimento que se tem é de um tipo inadequado e insatisfatório (PRESSMAN, 1995). Essa definição de Pressman traduz a necessidade contínua de se medir as características de um projeto de software, para que seu gestor tenha sempre o conhecimento atualizado do que está acontecendo em todo o ciclo de vida do projeto, e fazer um gerenciamento com competência, e transparência para os envolvidos no processo.

Quando falamos de métrica, temos que ter em mente que se trata de dados (números) quantitativos que irão mostrar, em forma de indicadores, o estado atual de projetos em execução, e apoiar na realização de estimativas necessárias para uma boa gestão de novos projetos. A busca pela qualidade utilizando métricas de software deve ser aplicada tanto às pessoas que produzem o produto, quanto para o processo em que se desenvolve o produto. É muito importante que os gerentes de projetos de software tenham o hábito de sempre estar medindo seus projetos para saberem como os mesmos estão evoluindo e apresentarem de forma clara os resultados obtidos para as pessoas envolvidas (CAMPOS, 2013).

A medição tem seu papel muito importante dentro da Engenharia de Software, especialmente na gerência de projetos de software, seja qual for a metodologia a ser utilizada. A medição é analisada por Gerentes de Projetos de Software e coletada pelos engenheiros de software. Sem as métricas, temos dados subjetivos, os quais não serão de grande importância, pois como vimos anteriormente, as métricas são expressas de forma quantitativa, ou seja, em números. As métricas são produzidas em três fases: coleta de dados, cálculo dos dados e análise dos dados. É importante saber escolher a

metodologia que melhor se encaixa no projeto, trazendo resultados mais precisos, sejam eles bons ou ruins (CAMPOS, 2013).

Segundo Hazan, o processo de estimativa de projetos de software possui quatro atividades essenciais para uma boa gestão: estimar o tamanho do produto a ser desenvolvido, estimar o esforço a ser empregado para sua implementação, estimar a duração e o custo do projeto. O responsável pelas estimativas deve analisar os requisitos para garantir a qualidade do produto e então estimar o tamanho do projeto de software. O próximo passo é calcular o esforço necessário e então derivar as estimativas de prazo e custo com base nas estimativas de tamanho. Desta forma, a partir do cálculo do tamanho do projeto, é possível calcular todas as outras estimativas, de forma a identificar as necessidades de recursos financeiros e de pessoal, fazer compensações de custo, produtividade, qualidade e quantificar o impacto das mudanças no projeto de desenvolvimento (HAZAN, 2009).

As organizações estão buscando, cada vez mais métodos e processos para dimensionar o tamanho de seus projetos de software, de forma a definir esforço, custo e prazos necessários para o desenvolvimento do software, tendo em vista a facilidade de relacionamento com o cliente, o gerenciamento dos riscos do projeto e o controle do cronograma, beneficiando a gerência de projetos para manter a qualidade dos seus produtos (SANTOS; MOREIRA, 2006).

Os principais problemas que um Gerente de Projetos de Software pode enfrentar estão relacionados aos processos de estimativa dos prazos de desenvolvimento dos projetos. Influenciando estes prazos e suas estimativas, temos a composição da equipe de desenvolvimento, e o acompanhamento das etapas realizadas do projeto. Soma-se a isso, as crescentes mudanças no mercado e as novas tecnologias disponíveis para se projetar e desenvolver softwares, e temos um cenário competitivo extremamente dinâmico no qual as empresas desenvolvedoras de software devem se adaptar para continuarem competitivas e principalmente manterem sua fatia de mercado, duramente conquistadas. O papel do Gerente de Projetos de Software se torna, portanto, fundamental na sobrevivência e crescimento destas empresas no mercado (CAMPOS, 2013).

A ausência de estimativas de tamanho ou estimativas incorretas no projeto de desenvolvimento do produto de software gera, por consequência, cronogramas atrasados e elevação no custo do projeto. Sem as medições quantitativas, o gerente de projeto não

conhece a dimensão do que está sendo produzido e por consequência, não realiza uma boa gestão, por essa razão o software deve ser medido (PRESSMAN, 1995).

Um dos problemas no gerenciamento do projeto é o desvio da situação atual em relação ao caminho projetado. Para que esse desvio seja percebido, é necessário analisar as métricas definidas e coletadas, assim como os indicadores gerados pelo inter-relacionamento delas. Essa análise permite que as várias dimensões de aspectos, como cronograma, custos, qualidade, riscos ou escopo sejam monitorados e ajustados quando necessário (VAZQUEZ et al., 2010).

Com a utilização de métricas, justificar e defender as decisões tomadas fica mais fácil para o gerente de projetos. As consequências dos desvios ocorridos no projeto tendem a ser menos graves, considerando que eles foram identificados e gerenciados com antecedência. O gestor toma suas decisões não apenas com base na sua experiência individual, mas também na avaliação de indicadores que refletem uma tendência do comportamento futuro. Esta tendência deve ser derivada não só das experiências passadas no projeto, mas também de experiências semelhantes de outros projetos de dentro e fora da organização (VAZQUEZ et al., 2010).

É muito importante observar a estreita relação entre gerência de projetos e medição. Para acompanhar o andamento do projeto, é preciso medir o progresso e comparar com o estimado. Mesmo no planejamento, sobretudo quando se pretende utilizar dados de projetos anteriores, dados de métricas são muito importantes. Não é possível controlar o que não se pode medir e, principalmente, só é possível chegar a boas estimativas com base em dados históricos se dados forem coletados criteriosamente. Assim, as medidas dão visibilidade ao estado do projeto, permitindo tanto saber para onde ir no início do projeto quanto verificar se o rumo está correto, tomando ações corretivas quando necessário (VAZQUEZ et al., 2005).

### **2.1.3. Implantação de um Programa de Medição**

Existem diversas formas de implantação de um programa de medição, podendo tal implementação variar de acordo com cada organização, pelo processo de desenvolvimento de software utilizado, pela quantidade de recursos disponíveis, pela forma adotada para implantação, pelos objetivos e metas da organização, pelos indicadores desejáveis, pela técnica de medição a ser utilizada, dentre outras variantes (BASSETO, 2008).

Um programa de medição, quando implementado por uma organização, passa a fazer parte do seu contexto de desenvolvimento de software, sendo que novas atividades são atribuídas a profissionais e métricas são apuradas com base em produtos específicos gerados por este contexto de desenvolvimento (BASSETO, 2008).

As empresas necessitam de dados para tomada de decisão. E estes dados, são frequentemente baseados em resultados de medições, através de algum instrumento ou meios de medição (CANOSSA, 2005).

Alguns autores como Basili et.al e Perkins et al., enfatizam que a utilização de métricas por si só, não apresentam resultados significativos, e defendem a criação de um programa de mensuração, ou seja, uma sequência de ações coordenadas para realizar as medições em processos de software. Além disto, este tipo de programa deve ser baseado nos objetivos da organização e cuidadosamente planejado, implementado e regularmente avaliado para ser efetivo (BASILI et al., 1994; PERKINS et al., 2003).

No entanto, a implementação de um programa de medição não constitui uma tarefa simples, e tem sido o grande desafio das empresas. Quando uma organização decide por iniciar um programa de medição, várias providências devem ser tomadas: a técnica de medição deve ser padronizada, as medidas a serem coletadas devem ser selecionadas, procedimentos de medição devem ser institucionalizados e documentados, a forma de armazenamento das medidas em uma base histórica precisa ser definida, e diretrizes para análise desses dados devem ser estabelecidas (BORGES, 2003).

A seguir descreveremos, de forma sucinta e consolidada, algumas orientações e lições aprendidas citadas por algumas referências relevantes da literatura (CLARK, 2002; GOLDENSON, 2003; KAN, 2003; KAUTZ, 1999; PFLEEGER, 1993; PFLEEGER et al., 1997; RIFKIN; COX, 1991; WIEGERS, 1997; WIEGERS, 1999):

1. **Necessidade de foco desde os estados iniciais:** quanto mais cedo se começa a medir mais cedo o gerente terá um controle mais objetivo do projeto, como por exemplo, gerenciar os riscos, alocar recursos e detectar problemas.
2. **Começar com um conjunto pequeno de métricas:** um programa de medição impacta a cultura organizacional, a forma com que as pessoas trabalham, e requer recursos para ser implantado. Deve-se começar com poucas métricas, que meçam os aspectos mais críticos e relevantes do software e de seus processos de desenvolvimento. De acordo com a maturidade do processo, o programa deve evoluir a partir das necessidades e possibilidades da organização.

3. **Utilização de um conjunto de métricas coerentes:** deve-se começar o programa com poucas métricas e evoluir esta quantidade aos poucos, à medida que a cultura de medição vai evoluindo na organização. Um estudo deve ser realizado, inicialmente, a fim de se definir poucas métricas, com mais importância para organização, e que já tragam algum tipo de benefício.
4. **Utilização de um conjunto de medições rigorosamente definido:** a definição clara de cada métrica a ser utilizada é fundamental para garantir a consistência da coleta e análise dos dados. Uma definição confusa de métricas permite que as pessoas interpretem as mesmas de diversas formas, podendo gerar uma visão equivocada do processo e do produto, através de coletas, reportagens ou interpretação errônea de dados.
5. **Não utilizar métricas para motivar comportamentos esperados:** se impomos valores como meta exigida em programas de medições, a equipe sempre irá encontrar maneiras de reportar exatamente o que o programa requer. As organizações devem utilizar as medições para entender sua realidade e buscar a melhoria, e não impor regras e exigir mudanças bruscas nos dados da realidade.
6. **Automatizar a coleta e reportagem dos dados:** é recomendável a implantação de ferramentas embutidas no ambiente de desenvolvimento para a reportagem dos dados. Como exemplo temos programas automáticos para armazenamento e coleta de esforço, contadores de linhas de código automatizados, utilização de planilhas automáticas para geração dos relatórios dos dados.
7. **Motivar a Gerência:** a alta gerência deve ser motivada a perceber o valor do programa de medição como uma ferramenta de apoio gerencial, para então patrocinar os custos da implantação e institucionalização do mesmo.
8. **Estabelecer expectativas:** uma vez que medições de software podem ser utilizadas para diversos propósitos, as expectativas e objetivos do programa e de cada métrica devem ser claramente estabelecidos desde o início de sua definição. Isso se faz necessário para que se tornem explícitos os reais benefícios a serem alcançados pelo programa, evitando dessa forma, que falsas expectativas sejam criadas. Isto poderia levar o programa de medições ao seu cancelamento devido à apresentação de resultados insatisfatórios comparados com as expectativas estabelecidas.

9. **Envolvimento de todos desde os estágios iniciais:** o envolvimento das pessoas em todos os estágios de implantação de um programa de medições é essencial para motivar e passar a todos o real objetivo do programa, ganhando assim um maior número de aliados. Fazer com que membros da equipe de desenvolvimento sejam envolvidos também nas atividades de análise dos dados é importante para que a equipe seja informada a respeito de como os dados estão sendo utilizados. Através dessa transparência, mostra-se que os dados realmente estão sendo utilizados, e a equipe de desenvolvimento passa a enxergar esse valor.
10. **Educação e treinamento:** todas as pessoas envolvidas e influenciadas pelo programa de medições devem ser treinadas para a implantação do programa nos aspectos técnicos, gerenciais e culturais. Esses treinamentos são necessários para que todos entendam os reais objetivos do programa e seus papéis e responsabilidades dentro do mesmo.
11. **Ganhar confiança:** os dados coletados devem ser utilizados para a melhoria do processo, e não para a punição, porém essa utilização de dados de forma ética não é fácil, e deve ser bem planejada dentro da organização. O uso de métricas para punição ou recompensa não é apropriado, pois o medo de reportar dados reais, visto que serão utilizados para avaliação de desempenho, é um dos principais aspectos de fracasso de um programa de medições. Uma boa prática para evitar que isso aconteça no início do programa é apresentar os dados de forma anônima para os níveis gerenciais acima.
12. **Adotar uma abordagem evolutiva:** periodicamente, o programa de medição deve ser revisado no que se refere aos seus objetivos, forma de utilização e aspecto custo-benefício, de forma a estar continuamente alinhado aos objetivos e prioridades organizacionais.
13. **Descartar o que não é útil:** durante inúmeras revisões do programa, para ajustá-lo ou mesmo corrigi-lo, pode-se constatar que partes do mesmo (métricas, ferramentas e procedimentos) não são mais úteis e devem ser descartados, uma vez que não estão mais adequados à realidade da organização.
14. **Fornecer informações corretas para pessoas certas:** qualquer programa de medição vai perder o seu valor se não der suporte à alta gerência com o nível

de informação necessário para tomada de decisões. Os resultados gerados devem ser objetivos, concisos e com o nível de detalhes apropriado para a cadeia gerencial em questão. Não adianta fornecer relatórios detalhados para a alta gerência da organização. Por outro lado, para os gerentes de desenvolvimento os detalhes são apropriados na maioria das situações.

15. **Incentivar os desenvolvedores a fazer uso das métricas:** o mais importante desse aspecto é que a equipe de desenvolvimento possa, de fato, utilizar os dados para entender como suas atividades estão sendo executadas, onde estão localizados os problemas, e então definir diretrizes para melhorar todo o processo e produto gerado pelo desenvolvimento.
16. **Compreender que a adoção leva tempo:** a sistematização de técnicas de medições em qualquer organização é um processo lento, visto que seus resultados não são imediatos, e muitas vezes geram mudanças culturais na forma de trabalho das pessoas. A mudança pode levar anos, e deve ser continuamente reforçada através da quantificação de seus benefícios, para então poder sobreviver e crescer continuamente. Deve estar claro para a alta gerência a necessidade de investimento inicial para obter retorno satisfatório em médio ou longo prazo.
17. **Má interpretação dos dados:** as tendências apresentadas nos resultados das medições podem ser facilmente mal interpretadas se as pessoas responsáveis pela análise dos dados não tiverem um grande entendimento do processo de desenvolvimento completo e não conhecerem em detalhes os objetivos e o programa de medição propriamente. Uma das grandes dificuldades em analisar medições de software é entender o que está por trás da variação dos dados apresentados.

Para se obter uma boa utilização da métrica adotada se faz necessário a construção de uma base histórica das contagens dos projetos da organização. Através dessa base é possível obter estimativas mais precisas, além de poder calcular indicadores importantes para auxílio na tomada de decisão (MARETTO; JUNGER, 2008).

Como exemplo de informações que podem ser coletadas para compor os indicadores pode-se ter valores estimados e realizados para: homem hora, quantidade de pontos de função, custo, defeitos encontrados, defeitos removidos, além de características do processo de estimativa e fatos ocorridos durante a realização das atividades que estão sendo mapeadas (SIMÕES, 2004).

A utilização da base de dados do *International Software Benchmarking Standards Group* (ISBSG, 2014), contendo informações da produtividade de projetos fornecidas por empresas de desenvolvimento de software de todo o mundo, pode ser uma tarefa difícil, pois os dados obtidos podem fugir um pouco da realidade de cada organização. Assim, as empresas buscam criar os seus próprios repositórios, o que garante um auxílio na precisão das estimativas. Em geral os dados históricos dos projetos devem conter qualquer informação que possa ajudar numa nova estimativa, por exemplo: plataforma, processo de desenvolvimento, tipo de linguagem, pontos de função não ajustados ou ajustados, os valores estimados e reais para produtividade, custo e esforço (MARETTO; JUNGER, 2008).

#### **2.1.4. Padrão ISO de Medição Funcional**

Medição funcional é o termo definido para métodos que dimensionam o tamanho do software a partir das funções solicitadas pelo usuário. No começo da década de 1990 podiam ser reconhecidas diversas técnicas de medição de tamanho funcional (Functional Size Measurement - FSM), surgidos a partir de diferentes interpretações da técnica original da Análise de Pontos de Função proposto por Allan Albrecht. Com o objetivo de resolver a inconsistência existente entre essas técnicas e estabelecer uma técnica mais rigorosa de medição funcional, um grupo de usuários de métricas de software da Austrália, Reino Unido, Holanda e Estados Unidos formaram o grupo de trabalho denominado WG12 (Working Group 12), subordinado ao SC7 (Sub-Committee Seven) do JTC1 (Joint Technical Committee One) estabelecido pela ISO (International Organization for Standardization) em conjunto com o IEC (International Engineering Consortium) (VAZQUEZ et al., 2010).

Como resultado dos trabalhos do WG12, foi estabelecido um padrão internacional chamado de norma 14143 (ISO/IEC 14143). A norma ISO/IEC 14143 foi estabelecida para garantir que todas as técnicas de medição funcional sejam baseados em conceitos similares, e possam ser testados para assegurar que eles se comportam de maneira similar e da forma esperada por uma técnica de medição (VAZQUEZ et al., 2010).

Segundo VAZQUEZ et al., atualmente são cinco as técnicas de medição funcional que são aderentes ao ISO/IEC 14143 (VAZQUEZ et al., 2010):

- APF: CPM 4.3 (ISO/IEC 20926);
- NESMA: CPM 2.1 (ISO/IEC 24570);

- Mark II (ou MK II): CPM 1.3.1 (ISO/IEC 20968);
- COSMIC-FFP: Measurement Manual 2.2 (ISO/IEC 19761);
- FISMA 1.1 (ISO/IEC 29881).

Para esse trabalho foram selecionados as técnicas de Análise de Pontos de Função por ser a mais utilizada pela indústria de desenvolvimento de software, e o COSMIC-FFP que a partir de sua versão 3.0 teve seu nome simplificado para COSMIC e considerado por seus criadores como promissor por ser preciso e abrangente para medir software e estimar esforço de desenvolvimento.

## **2.2. Análise de Pontos de Função – APF**

### **2.2.1. Definições, Conceitos e Regras**

A técnica de análise de pontos de função foi desenvolvida na década de 1970 pela IBM como uma alternativa às métricas baseadas em linhas de código. Ela foi publicada em 1979 por Allan Albrecht na tentativa de reduzir as dificuldades encontradas pelos usuários das linhas de código como medida de tamanho de software (VAZQUEZ *et al.*, 2010).

Com o aumento no número de usuários utilizando a APF viu-se a necessidade de implantar um guia que interpretasse as regras originais definidas por Allan para os novos ambientes que surgissem. O crescimento de adeptos a APF motivou a criação do *International Function Point Users Group* (IFPUG), em 1986. A partir desta data, o IFPUG se tornou o responsável pela definição das regras de contagem, pelo treinamento de usuários, pela certificação dos profissionais interessados na utilização desta técnica em ambiente de trabalho e pelo fornecimento de dados históricos de produtividade da indústria de desenvolvimento de software (ANDRADE, 2004; IFPUG, 2010).

Análise de Pontos de Função é uma técnica de medição das funcionalidades fornecidas por um software do ponto de vista de seu usuário. Ponto de Função é a unidade de medida desta técnica que tem por objetivo tornar a medição independente da tecnologia utilizada para a construção de software. Ou seja, a APF busca medir o que o software faz, e não como ele foi construído. Portanto o processo de medição, ou a contagem de pontos de função, é baseada em uma avaliação padronizada dos requisitos lógicos do usuário (VAZQUEZ *et al.*, 2010).

De acordo com Vazquez e outros autores, a APF permite não só medir o tamanho do sistema em termos da funcionalidade fornecida ao usuário, mas também estimar seu

tamanho em qualquer fase do seu ciclo de vida, mesmo que os requisitos ainda não tenham sido detalhados (VAZQUEZ *et al.*, 2010).

Segundo Vazquez et al., é importante destacar que pontos de função não medem diretamente esforço, produtividade ou custo. É exclusivamente uma medida de tamanho funcional do software. Este tamanho em conjunto com outras variáveis é que pode ser usado para derivar produtividade e estimar esforço e custo (VAZQUEZ *et al.*, 2010).

Simões também comenta que é necessário um acompanhamento constante das medições para gerar os indicadores possíveis, exigindo assim esforço e disciplina nas contagens de pontos de função nas diversas fases do projeto, e principalmente criação de uma base histórica de produtividade (SIMÕES, 2004).

É importante destacar que a aplicação da técnica requer um especialista em contagem de PF (consultor de métricas) com conhecimentos em ER. O *International Function Point Users Group* (IFPUG) aplica um exame para certificar especialistas em contagem de Pontos de Função. A certificação é denominada CFPS (Certified Function Point Specialist) (HAZAN, 2005).

### **2.2.1.1. Objetivos e Benefícios**

A Análise de Pontos de Função é uma técnica para medir o tamanho funcional de um software do ponto de vista do usuário. Sendo assim, o IFPUG define como objetivos da técnica (VAZQUEZ *et al.*, 2010):

Medir a funcionalidade que o usuário solicita e recebe;

Medir o desenvolvimento e manutenção de software de forma independente da tecnologia utilizada para sua implementação;

Ser simples o suficiente para minimizar o trabalho adicional envolvido no processo de medição;

Ser uma medida consistente entre vários projetos e organizações.

A APF possibilita ao gerente de projeto uma contagem indicativa na fase inicial de desenvolvimento sem que haja um detalhamento no modelo de dados. Em seguida, na fase de projeto, a contagem passa a ser mais precisa e no término do desenvolvimento do software, há uma contagem detalhada (ANDRADE, 2004).

As organizações aplicam a APF como uma técnica para determinar o tamanho do pacote de software adquirido; para apoiar a análise da produtividade e qualidade; e para

estimar os custos, os recursos e esforços de projeto de desenvolvimento e manutenção de software (HAZAN, 2004).

É possível destacar vários benefícios da aplicação da análise de pontos de função nas organizações (VAZQUEZ *et al.*, 2010):

- É uma ferramenta para determinar o tamanho de um pacote adquirido pela contagem de todas as funções incluídas;
- Provê auxílio aos usuários na determinação dos benefícios de um pacote para sua organização, através da contagem das funções que especificamente correspondem aos seus requisitos (make or buy);
- Suporta a análise de produtividade e qualidade, seja diretamente ou em conjunto com outras métricas como esforço, defeitos e custo;
- Apoia o gerenciamento de escopo de projetos;
- Complementa o gerenciamento dos requisitos ao auxiliar na verificação da solidez e completude dos requisitos especificados;
- É um meio para estimar custo e recursos para o desenvolvimento e manutenção de software;
- É um fator de normalização para comparação de software ou para comparação da produtividade na utilização de diferentes técnicas.
- É uma ferramenta para fundamentar a negociação de contratos;

Segundo Aguiar (AGUIAR, 2003) apud (MACORATTI, 2005), dentre as principais razões para a utilização da APF como métrica estão as seguintes:

- O fato dos Pontos de Função serem mantidos por uma organização internacional sem fins lucrativos, o IFPUG, desde 1986;
- Os PF possuem suporte no Brasil através do chapter – BFPUG;
- Os PF serem padronizados pela ISO através da norma ISO/IEC 20296;
- Existir um grande acervo de informações sobre PF armazenadas em diversas organizações o que permite estudos e comparações;
- Os PF modelarem os requisitos em um nível de abstração mais alto e independente dos artefatos e poderem ser usados por organizações que usam qualquer forma de representação de requisitos;
- Os PF serem usados em contratos e licitações no Brasil em organizações governamentais e pelo mercado em geral.

### 2.2.1.2. Processo de Medição

Para conduzir uma contagem de pontos de função devem ser executadas as seguintes atividades, a fim de identificar e classificar os componentes funcionais básicos, de acordo com a Figura 2 (IFPUG, 2010):

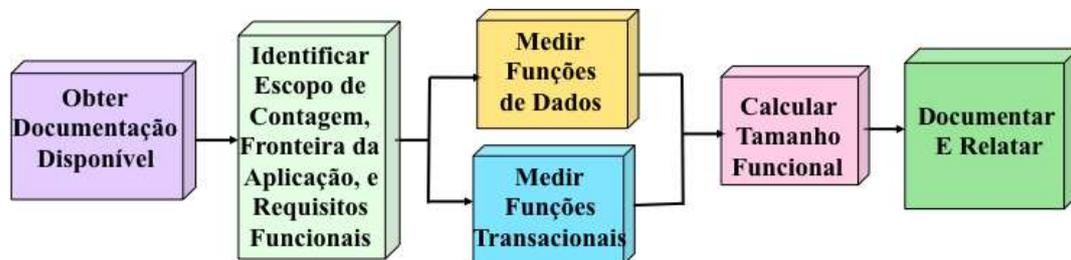


Figura 2.2 – Visão Geral do Processo de Contagem de Pontos de Função

Antes de iniciar as atividades da medição deve ser identificado qual o propósito da contagem, pois ele norteia todo o processo. O Propósito da Contagem é fornecer uma resposta a um problema de negócio. Ele determina algumas premissas para o processo de contagem; determina o tipo de contagem; estabelece o escopo da contagem; afeta o posicionamento da fronteira da aplicação; e define o nível de detalhe da contagem (VAZQUEZ *et al.*, 2010).

A APF pode ser utilizada para diversos tipos de necessidade por ser aplicada desde o início até o término do projeto. A utilização da APF gera estimativas mais precisas, o que acarreta em uma melhoria no gerenciamento de projetos de desenvolvimento de sistemas. Antes de iniciar a contagem de pontos por função, é preciso definir o tipo de contagem, o escopo e a fronteira da aplicação (HAZAN, 1999; SILVA, 2000; ANDRADE, 2004).

A determinação do tipo de contagem de pontos por função representa a função do objeto da contagem que será usado para estimar o projeto, podendo esse projeto ser de desenvolvimento, de manutenção ou uma aplicação já finalizada. A seguir detalharemos os tipos de contagem (IFPUG, 2010; VAZQUEZ *et al.*, 2010):

- **Projeto de Desenvolvimento:** A contagem de pontos de função mede as funcionalidades fornecidas aos usuários finais do software quando da sua primeira instalação, abrangendo, também, as eventuais funções de conversão de dados necessárias à implantação do sistema.

- **Projeto de Melhoria:** A contagem de pontos de função mede as funções adicionadas, modificadas ou excluídas do sistema pelo projeto, como também as eventuais funções de conversão de dados.
- **Aplicação:** A contagem de pontos de função mede as funcionalidades já entregues aos usuários em uma aplicação instalada, também chamado de *baseline*. Esse tipo de projeto fornece uma medida atualizada das funcionalidades da aplicação instalada. Ele é inicializado ao final da contagem do projeto de desenvolvimento, sendo atualizado no término de todo projeto de melhoria que altera as funcionalidades da aplicação.

As atividades a serem executadas pelo processo de medição serão descritas nas subseções dessa seção, de acordo com o manual de práticas e contagem do IFPUG (IFPUG, 2010).

#### **2.2.1.2.1. Obter a Documentação Disponível**

O primeiro passo do processo de medição consiste em buscar a documentação disponível sobre o sistema e/ou projeto que será medido. O propósito da contagem ajuda a definir quais artefatos são mais interessantes para o trabalho de medição. Em essência a documentação ideal deve (VAZQUEZ *et al.*, 2010):

- Descrever a funcionalidade entregue pelo software; ou
- Descrever a funcionalidade que é impactada pelo projeto de software medido.

Uma documentação adequada pode incluir requisitos, modelos de dados/objetos, diagramas de classe, diagramas de fluxo de dados, casos de uso, descrições procedurais, *layout* de relatórios e telas, manuais de usuário e outros artefatos do desenvolvimento de software (IFPUG, 2010).

Se não há documentação suficiente disponível, é preciso buscar o acesso aos especialistas no negócio para complementar essas falhas. Isso acarreta esforço adicional na medição (VAZQUEZ *et al.*, 2010).

A documentação de um projeto/aplicação disponibilizada para sua medição deve refletir a visão de seu usuário. Um usuário é qualquer pessoa ou coisa que se comunica ou interage com o software a qualquer momento (IFPUG, 2010).

A visão do usuário representa uma descrição formal das necessidades dos negócios, na linguagem do usuário. Os desenvolvedores traduzem a informação do usuário para

informações em linguagem técnica a fim de prover uma solução. A visão do usuário (IFPUG, 2010):

- É uma descrição das funções do negócio;
- Pode ser feita por declaração verbal pelo usuário através de seu ponto de vista;
- É aprovada pelo usuário;
- Pode ser usada para medir o tamanho funcional;
- Pode variar na forma física (ex., catálogo de transações, propostas, documento de requisitos, especificações externas, especificações detalhadas, manuais do usuário);

#### **2.2.1.2.2. Determinar o Escopo da Contagem e Fronteira da Aplicação, identificando os Requisitos Funcionais do Usuário**

O segundo passo no Processo de Medição é determinar o escopo e a fronteira da aplicação a ser contada, identificando os requisitos funcionais do usuário. O escopo da contagem define o conjunto de Requisitos Funcionais de Usuários para ser incluído na contagem de pontos de função. O escopo (IFPUG, 2010):

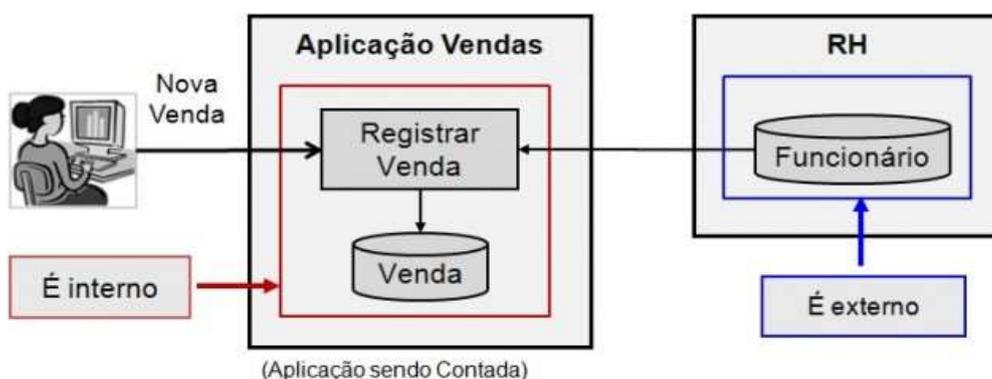
- Define o (sub)conjunto do software que está sendo medido;
- É determinado pelo propósito para a realização da contagem de pontos de função;
- Identifica quais funções serão incluídas na medida de tamanho funcional assim como fornece respostas relevantes para o propósito da contagem;
- Pode incluir mais de uma aplicação.

Para realizar uma contagem de pontos de função é necessário definir determinados contextos, entre eles identificar qual é o escopo para o tipo de contagem definida. O escopo da contagem não altera a aplicação das regras de contagem, mas influencia no tamanho da contagem e deve ser identificado depois que foi definido o tipo de contagem de pontos de função a ser realizada (CAMPOS, 2010a).

O objetivo da identificação do escopo de contagem consiste em definir quais funções serão incluídas na contagem: se abrange um ou mais sistemas, se abrange apenas uma parte do sistema e se compreenderá todas as funcionalidades, apenas as

funcionalidades em uso pelo usuário ou funcionalidades específicas (FABIAN, 2008; VAZQUEZ *et al.*, 2010).

A fronteira da aplicação pode ser considerada como uma interface conceitual, linha imaginária, que delimita a área de influencia da aplicação que está sendo analisada e o usuário, o mundo exterior. Se olharmos para essa linha imaginária, poderemos ver toda a funcionalidade que a aplicação recebe ou fornece a usuário. Para definir a fronteira é preciso analisar a aplicação sob a perspectiva da visão do usuário de negócio e a sua definição é independente de considerações técnicas ou da forma como a aplicação está implementada (CAMPOS, 2010b). A Figura 2.3 mostra a fronteira entre a aplicação de vendas e seus usuários.



**Figura 2.3 – Fronteira da Aplicação (CAMPOS, 2010b)**

O IFPUG especifica as seguintes regras para determinação da fronteira da aplicação (VAZQUEZ *et al.*, 2010):

- Sua determinação deve ser feita com base no *Ponto de Vista do Usuário*. O foco deve estar no que ele pode entender e descrever;
- A Fronteira entre aplicações deve ser baseada na separação das funções conforme estabelecido pelos processos de negócio, não em considerações tecnológicas;
- Em projetos de melhoria, a fronteira estabelecida no início do projeto deve estar de acordo com a fronteira já estabelecida para a aplicação sendo modificada.

O posicionamento da fronteira entre o software sob análise e outra aplicação do software pode ser subjetivo. É comum haver dificuldade para delinear onde uma aplicação termina e a outra se inicia. A fronteira deve ser definida a partir de uma perspectiva de negócio ao invés de se basear em considerações técnicas ou físicas. É

importante que a fronteira seja delimitada com cuidado, de forma que todos os seus cruzamentos de dados possam ser potencialmente incluídos no escopo da contagem (IFPUG, 2010).

### **2.2.1.2.3. Medir Funções de Dados**

As funções de dados representam a funcionalidade oferecida ao usuário para satisfazer requisitos de dados internos e externos à aplicação, ou seja, representam os requisitos de armazenamento de dados do usuário. Uma função de dado pode ser classificada como um arquivo lógico interno ou um arquivo de interface externo. O termo arquivo aqui não significa arquivo físico ou tabela. Nesse caso, arquivo se refere a um grupo de dados logicamente relacionados e não à implementação física destes grupos de dados (IFPUG, 2010; VAZQUEZ *et al.*, 2010).

Um Arquivo Lógico Interno (ALI) é um grupo de dados ou de informações de controle logicamente relacionados, reconhecido pelo usuário, mantido dentro da fronteira da aplicação que está sendo contada. A intenção primária de um ALI é armazenar dados mantidos através de um ou mais processos elementares da aplicação que está sendo contada (IFPUG, 2010).

Um Arquivo de Interface Externa (AIE) é um grupo de dados ou de informações de controle logicamente relacionados, reconhecido pelo usuário, referenciado pela aplicação que está sendo contada, porém, mantido dentro da fronteira de uma outra aplicação. A intenção primária de um AIE é armazenar dados referenciados através de um ou mais processos elementares dentro da fronteira da aplicação que está sendo contada. Isto significa que um AIE contado para uma aplicação deve ser um ALI em outra aplicação (IFPUG, 2010).

Cada ALI ou AIE deve ser classificado em relação à sua complexidade funcional (baixa, média ou alta) com base em (VAZQUEZ *et al.*, 2010):

- Quantidade de Tipos de Registros (TR/RLR). Onde tipo de registro é um subgrupo de dados, reconhecido pelo usuário, e logicamente relacionados. Na análise de pontos de função em sistemas *Web* as tabelas reconhecidas pelo usuário, onde seus campos são utilizados por alguma transação da aplicação contada, devem ser consideradas como tipos de registros de um arquivo lógico.

- Quantidade de Tipos de Dados (TD/DER). Onde tipo de dado é um campo único, reconhecido pelo usuário, não repetido, e que atravessa a fronteira da transação. Nas contagens em sistemas *Web* os campos utilizados por alguma transação da aplicação contada, e que fazem parte das tabelas que são identificadas como tipos de registros de um arquivo lógico, devem ser considerados como tipos de dados.

A complexidade dos arquivos lógicos é determinada a partir das quantidades de seus tipos de dados e tipos de registros identificados, de acordo com a Tabela 2.1.

Tipos de Registros (TR's)	Tipos de Dados (TD's)		
		< 20	20 - 50
1	Baixa	Baixa	Média
2 - 5	Baixa	Média	Alta
> 5	Média	Alta	Alta

**Tabela 2.1 – Tabela de complexidade funcional dos ALI e AIE (IFPUG, 2010)**

A contribuição dos arquivos lógicos, ou seja, o tamanho em pontos de função, deve ser calculada após a determinação de sua complexidade, baseada na Tabela 2.2.

Tipo de Função	Baixa	Média	Alta
Arquivo Lógico Interno	7 PF	10 PF	15 PF
Arquivo de Interface Externa	5 PF	7 PF	10 PF

**Tabela 2.2 – Tabela de contribuição dos pontos de função das funções do tipo dado (IFPUG, 2010)**

Os Requisitos de armazenamento, Funcionais e Não Funcionais, de uma aplicação são classificados como (IFPUG, 2010):

- **Dados de Negócio:** também podem ser chamados Dados Essenciais do Usuário (“Core User Data”) ou Objetos de Negócio. Este tipo de dado reflete a informação necessária a ser armazenada e recuperada pela área funcional abordada pela aplicação e, geralmente, representam um percentual significativo das entidades identificadas. Ex: Arquivo de Cliente, Arquivo de Fatura, Arquivo de Empregado, Arquivo de Função.
- **Dados de Referência:** são armazenados para apoiar as Regras de Negócio na manutenção de Dados de Negócio. Os Dados de Referência geralmente representam um pequeno percentual das entidades identificadas. Ex: Faixas Salariais, Taxas de Desconto, Alíquotas de Impostos, Configuração de Limites.
- **Dados de Código:** é uma implementação de requisitos técnicos e não deve influenciar o tamanho funcional da aplicação na análise dos pontos de função, não sendo contado na medição. Eles são chamados, também, de metadados e, em geral, não são especificados pelo próprio usuário, sendo identificados pelo desenvolvedor em resposta a um ou mais requisitos técnicos. Ex: Estado (Código, Nome); Tipo de Pagamento (Código, Descrição).

Existem vários tipos diferentes de Dados de Código, os quais se enquadram em três áreas gerais (IFPUG, 2010):

- **Dados de Substituição:** Este tipo de Dado de Código contém um código e um nome explicativo ou descrição. Este tipo de Dado de Código pode servir como um meio para tornar mais ágil a entrada de dados para usuários experientes, o nome / descrição explicativo para usuários menos experientes ou para listagens como em relatórios. Este tipo de Dado de Código também pode ser implementado para economizar espaço de armazenamento ou ser um resultado de normalização. Ex: Código do País, Nome do País.
- **Dados Constantes ou Estáticos:** Este tipo de Dado de Código contém uma e apenas uma ocorrência independentemente da quantidade de atributos; podem mudar, mas muito raramente. Ex: Uma entidade com dados sobre

uma organização em particular, utilizados como cabeçalho/rodapé em todos os relatórios do sistema.

- **Dados com Valores Válidos:** Este tipo de Dado de Código fornece uma lista de valores válidos para um atributo de um ou mais tipos de objetos de negócio. Devem conter dados basicamente estáticos; se não forem, podem ser Dados de Referência ou Dados de Negócio. Este tipo de Dado de Código é implementado para satisfazer requisitos como reduzir erros e aumentar a facilidade de uso pelo usuário. Ex: Nome do Estado - contém todos os valores válidos para o atributo nome do estado.

Os requisitos de armazenamento enquadrados como dado de código não devem ser considerados como arquivos lógicos. Eles são uma implementação de requisitos técnicos e não devem influenciar o tamanho funcional da aplicação. As transações que existem exclusivamente para a manutenção de dados de código não devem ser consideradas processos elementares, assim como os dados de código não devem ser contados como arquivos referenciados nos processos elementares que os leiam e/ou atualizem. Apenas os dados de negócio e de referência devem ser identificados como requisitos funcionais do usuário e contabilizados na análise de pontos de função (VAZQUEZ *et al.*, 2010).

#### **2.2.1.2.4. Medir Funções de Transações**

A função de transação representa a funcionalidade que é fornecida ao usuário para o processamento de dados por uma aplicação. Uma função de transação pode ser classificada como entrada externa, saída externa, ou consulta externa (CAMPOS, 2011; IFPUG, 2010).

Processo Elementar é a menor unidade de atividade que satisfaz todas as seguintes regras para ser reconhecido como uma função de transação (VAZQUEZ *et al.*, 2010):

- Tem significado para o negócio, ou seja, é reconhecido pelo Usuário e satisfaz um Requisito Funcional;
- Constitui uma transação completa;
- É auto-contido;
- Deixa o negócio da aplicação sendo medida em um estado consistente.

Para uma transação ser considerada como processo elementar deve existir uma garantia que nenhuma outra com os mesmos tipos de dados, arquivos referenciados, e lógicas de processamento, já foi contada no projeto (VAZQUEZ *et al.*, 2010).

O processo elementar deve ser classificado de acordo com seu objetivo principal, em uma das funções de transações abaixo (HAZAN, 2001; VAZQUEZ *et al.*, 2010):

- **Entrada Externa (EE):** são processos elementares (transações) que processam dados ou informações de controle que entram pela fronteira da aplicação. O objetivo principal de uma EE é manter um ou mais ALIs e/ou alterar o comportamento do sistema. Ex: Incluir Cliente; Alterar Funcionários; Excluir Pedido.
- **Saída Externa (SE):** são processos elementares que enviam dados ou informações de controle para fora da fronteira da aplicação. Seu objetivo é apresentar informações ao usuário que foram recuperadas através de um processamento lógico e não apenas uma simples recuperação de dados. Uma SE deve envolver cálculos ou criação de dados derivados ou também, pode manter um ALI ou alterar o comportamento do sistema. Ex: Relatório do total de vendas por funcionário, tela de Login (com criptografia).
- **Consulta Externa (CE):** assim como uma SE, é um processo elementar que envia dados (ou informações de controle) para fora da fronteira da aplicação. Porém, seu objetivo é apresentar informação para o usuário, por meio apenas de uma recuperação das informações, sem a realização de nenhum cálculo nem a criação de dados derivados. Nenhum ALI é mantido durante sua realização, nem o comportamento do sistema é alterado. Ex: Consulta de Clientes, Drop-Downs desde que recuperem dados de arquivos lógicos (ALIs e/ou AIEs). Os Drop-Downs estáticos, com valores codificados diretamente no programa-fonte, não são contados.

Os procedimentos de contagem de função de transação devem incluir os seguintes passos:

1. Identificar cada processo elementar.
2. Determinar o processo elementar único.
3. Classificar cada função de transação como Entrada Externa (EE), Saída Externa (SE) ou Consulta Externa (CE).

4. Determinar a complexidade funcional para cada função de transação e sua contribuição para o tamanho funcional.

Cada Entrada Externa, Saída Externa e Consulta Externa deve ser classificada com relação à sua complexidade funcional (Baixa, Média ou Alta) baseada em (VAZQUEZ *et al.*, 2010):

- Quantidade de Arquivos Referenciados (AR). Onde arquivo referenciado é um Arquivo Lógico Interno lido ou mantido, ou um Arquivo de Interface Externa lido, pela função do tipo transação.
- Quantidade de Tipos de Dados (TD). Onde tipo de dado é um campo único, reconhecido pelo usuário, não repetido, e que atravessa a fronteira da transação, sendo exibido para o usuário.

Determinadas as quantidades de Arquivos Referenciados e de Tipos de Dados, a classificação com relação à complexidade é fornecida pelas tabelas 2.3 e 2.4, de acordo com cada tipo de função de transação identificado (Entrada Externa e Saída Externa/Consulta Externa):

Arquivos Referenciados (ARs)	Tipos de Dados (TDs)			
		< 5	5 - 15	> 15
	< 2	Baixa	Baixa	Média
	2	Baixa	Média	Alta
> 2	Média	Alta	Alta	

**Tabela de complexidade para Entradas Externas (EE's)**

**Tabela 2.3 – Tabela de complexidade para Entradas Externas (EE's) (IFPUG, 2010)**

Arquivos Referenciados (ARs)	Tipos de Dados (TDs)			
		< 6	6 - 19	> 19
	< 2	Baixa	Baixa	Média
	2 - 3	Baixa	Média	Alta
> 3	Média	Alta	Alta	

**Tabela de complexidade para Saidas Externas (SE's) e Consultas Externas (CE's)**

**Tabela 2.4 – Tabela de complexidade para Saidas Externas (SE's) e Consultas Externas (CE's) (IFPUG, 2010)**

Após a conclusão da etapa de determinação da complexidade das funções do tipo transação, deve ser calculada sua contribuição, ou seja, seu tamanho em pontos de função, utilizando os seguintes critérios:

<b>Tipo de Função</b>	<b>Baixa</b>	<b>Média</b>	<b>Alta</b>
<b>Entrada Externa</b>	3 PF	4 PF	6 PF
<b>Saída Externa</b>	4 PF	5 PF	7 PF
<b>Consulta Externa</b>	3 PF	4 PF	6 PF

**Tabela 2.5 – Tabela de contribuição dos pontos de função das funções do tipo transação (IFPUG, 2010)**

### **2.2.1.3. Pontos de Função Não Ajustados e Ajustados**

Para a APF o tamanho funcional, também chamado de pontos de função não ajustados, representa o tamanho do software obtido pela quantificação dos requisitos funcionais do usuário, através da medição das funções de dados e de transação. Os Pontos de Função Não Ajustados (PFNA) de uma aplicação medem os requisitos solicitados e recebidos pelo usuário, sendo calculado a partir da soma das contribuições de todas as funções identificadas (IFPUG, 2010; VAZQUEZ et al., 2010).

Os pontos de função ajustados de um software são obtidos a partir da aplicação do fator de ajuste sobre os pontos de função não ajustados. O Fator de Ajuste (VAF) é baseado nas 14 características gerais do sistema (CGSs), definidas pela técnica do IFPUG, que classificam a funcionalidade geral da aplicação sendo contada. Cada característica tem descrições associadas que ajudam a determinar o nível de influência da característica. O nível de influência de cada característica varia em uma escala de 0 a 5, de sem influência até forte influência. O valor do Fator de Ajuste é definido a partir de uma fórmula que soma todos os níveis de influência encontrados para as 14

características gerais do sistema. Quando aplicado, o fator ajusta o tamanho funcional não ajustado em +/- 35% para produzir o tamanho funcional ajustado (IFPUG, 2010).

Enquanto as funções do tipo dado refletem requisitos específicos de armazenamento e as funções do tipo transação requisitos específicos de processamento, as características gerais refletem funções que afetam a aplicação de maneira geral. As 14 características gerais do sistema são (IFPUG, 2010):

1. Comunicação de Dados;
2. Processamento Distribuído;
3. Performance;
4. Configuração Intensamente Utilizada
5. Volume de Transações
6. Entrada de Dados On-Line
7. Eficiência do Usuário Final
8. Atualização On-Line
9. Processamento Complexo
10. Reusabilidade
11. Facilidade de Instalação
12. Facilidade de Operação
13. Múltiplos Locais
14. Facilidade de Mudança

O uso do fator de ajuste tornou-se opcional ao final do ano de 2002, como uma medida para aceitação dos pontos de função do IFPUG como uma técnica padrão de medição funcional aderente a norma ISO/IEC 14143, porque varias da CGSs contemplam requisitos não funcionais. Entretanto, antes mesmo do uso do fator de ajuste tornar-se opcional, uma pesquisa apoiada pelo IFPUG demonstrou que vários usuários já não o utilizavam (VAZQUEZ et al., 2010).

A norma ISO/IEC 14143 foi desenvolvida para garantir que todas as técnicas de Medição de Tamanho Funcional sejam baseadas em conceitos similares e que possam ser testadas para assegurar que elas se comportam de maneira similar e da forma esperada por uma técnica de medição, dependendo dos domínios funcionais a que se aplicam. Ao final do ano de 2002 a técnica de análise de pontos de função, conforme definida na versão 4.3.1 do manual do IFPUG, foi aprovada, sob a norma 20.926, como uma técnica de Medição de Tamanho Funcional aderente à norma ISO/IEC 14.143. (FATTO, 2012).

O Fator de Ajuste e suas 14 características gerais é um ponto tão criticado da técnica, resultado de uma grande variação na interpretação das CGSs e da constatação que algumas delas estão desatualizadas, que foi criado um grupo de trabalho do comitê de práticas de contagem do IFPUG (VAZQUEZ *et al.*, 2010), com a responsabilidade de se dedicar ao tema de medição dos requisitos não funcionais, tendo como objetivo criar uma outra técnica para medir esses requisitos independentemente da análise de pontos de função.

Atualmente a APF é a técnica de medição mais reconhecida e utilizada pelo mercado de desenvolvimento e manutenção de software, e o seu resultado final esperado é a medição dos pontos de função não ajustados.

### **2.2.2. Projetos de Melhoria Utilizando a APF**

Após a implantação, grande parte dos sistemas sofre alterações ao longo do período em que estão sendo utilizados. Seus requisitos são modificados para atender alguma necessidade do usuário, mudança de ambiente, introdução de novo hardware ou software, correção de erros e outros. Este processo de mudança do software após sua implementação é chamado de manutenção de sistemas. A manutenção de sistemas é uma realidade no mercado de software sendo considerada em estimativas como até 70% de todo o trabalho de engenharia de software (CALAZANS, 2006).

Manter sistemas propõe uma visão diferenciada do processo de desenvolvimento. Diferentemente de novo desenvolvimento, o processo de manutenção é mais complexo, especializado e com características específicas. Isso requer que outros processos, como os processos de medição, que interagem com o processo de construção do software de forma a viabilizar um melhor gerenciamento, sejam adaptados de forma a garantir o gerenciamento adequado de recursos, custos e tempo (CALAZANS, 2006).

Uma vez que uma aplicação foi desenvolvida e instalada, ela deve ser mantida (modificada) a fim de continuar satisfazendo às constantes necessidades de mudanças do negócio e do ambiente técnico. Esta manutenção inclui um grupo de atividades que são executadas durante esta fase do ciclo de vida da aplicação. Segundo o IFPUG, podemos classificar as manutenções de software como (IFPUG, 2010):

- **Manutenção Adaptativa:** Manutenção para fazer com que o software continue sendo utilizável em um ambiente alterado. Seu objetivo é fornecer as melhorias necessárias para adaptar as modificações no ambiente em que o

software deve funcionar. Medição do tamanho funcional é aplicável a um subconjunto de manutenções adaptativas. Isso inclui as funcionalidades do software adicionadas, alteradas ou excluídas bem como as funcionalidades do software fornecidas para converter dados e atender outros requisitos de conversão. (ISO/IEC 14764:2006)

- **Manutenção Corretiva:** Manutenção para corrigir falhas no hardware ou software. Modificação reativa de um software realizada depois da entrega para corrigir problemas descobertos. A modificação corrige o software para satisfazer requisitos. (ISO/IEC 14764:2006)
- **Manutenção Perfectiva:** Manutenção para melhorar a performance, facilidade de manutenção ou outros atributos do software instalado. Seu objetivo é fornecer melhorias para o usuário, melhoria da documentação do programa, e recodificação para melhorar a performance ou outros atributos do software. (ISO/IEC 14764:2006)

Apenas a manutenção evolutiva, considerada um subconjunto da manutenção adaptativa e que contempla todas as mudanças nos requisitos funcionais reconhecidos pelo usuário, deve ser medida pela APF. O projeto que tem no escopo esse tipo de manutenção é definido pelo IFPUG como Projeto de Melhoria.

O Tamanho Funcional do Projeto de Melhoria mede as modificações na aplicação instalada existente que adicionam, modificam ou excluem funções do usuário. Mudanças nas funcionalidades podem ocorrer a partir de novos requisitos, revisão de requisitos do usuário, mudanças legais/regulamentares ou novos usuários (IFPUG, 2010).

A APF considera como escopo de um projeto de melhoria as inclusões, exclusões e modificações realizadas nas funções de dados (ALI e AIE) e nas funções transacionais (EE, SE ou CE) já identificadas no projeto da aplicação instalada, ou baseline. O projeto de melhoria é calculado pela soma dos pontos de função das funções adicionadas, modificadas ou excluídas do sistema, como também as eventuais funções de conversão de dados.

Nos projetos de melhoria, segundo o IFPUG, deve ser considerado o mesmo critério de avaliação de tamanho para funções incluídas, alteradas e excluídas, mesmo sabendo que pode haver diferenças significativas de esforço para criar, alterar ou excluir uma função. A técnica de análise de pontos de função para projetos de melhoria não possui uma granularidade tão refinada em sua medição a ponto de dar um peso

adequado a cada função modificada, de acordo com o que de fato foi alterado nessa função. Essa característica torna a técnica do IFPUG para projetos de melhoria dificilmente utilizada pelo mercado de medição de software, considerando que a quantidade de pontos de função calculada pode não refletir o esforço utilizado para se manter o sistema, sendo necessário aderir a abordagens de outras entidades ligadas à medição funcional de software.

A abordagem proposta pela NESMA (Netherlands Software Metrics Users Association) para projetos de melhoria é uma alternativa para o problema exposto. A NESMA é uma associação de usuários de métricas que tem indicado alternativas de contagem, utilizando a APF, de forma a possibilitar medir de forma mais acurada alguns escopos em que a APF não tem se mostrado eficaz. Sua proposta em projetos de melhoria permite a redução do tamanho de cada função alterada considerando uma relação proporcional das alterações realizadas com o que existia anteriormente. Esta abordagem já é utilizada por uma grande maioria do mercado, que considera uma melhor equivalência do tamanho da demanda com o esforço realizado nas melhorias.

### **2.2.3. Pontos de Função no Brasil**

O uso da técnica da análise de pontos de função no Brasil começou significativamente no início da década de 1990. Porém, o interesse do mercado se consolidou apenas quando grandes contratos públicos de desenvolvimento e manutenção de sistemas começaram a ser baseados em pontos de função. Em paralelo o interesse crescente das organizações ligadas ao desenvolvimento de software por modelos de qualidade e maturidade (ISO e CMMI) contribuiu para o aumento de interesse no assunto. A criação do BFPUG (Brazilian Function Point User Group) em 1998, que é o *chapter* do IFPUG no Brasil, foi também um importante marco para essa consolidação (VAZQUEZ et al., 2010). Dos 893 profissionais certificados em pontos de função em todo o mundo, 371 são do Brasil, liderando esse número na frente da Coréia, Índia, Itália e EUA, o que comprova ainda mais essa consolidação (AGUIAR, 2014; BFPUG, 2008).

Diversas instituições públicas e privadas têm utilizado a métrica Ponto de Função nas estimativas e dimensionamento de tamanho funcional de projetos de software devido aos diversos benefícios de utilização desta métrica, destacando-se: regras de

contagem objetivas, independência da solução tecnológica utilizada e facilidade de estimativa nas fases iniciais do ciclo de vida do software (SISP, 2012).

A partir de 2010, a adoção ao uso de pontos de função acelerou no governo federal por causa da Instrução Normativa SLTI nº 4, que recomenda o uso de métricas em contratos de projetos de software, restringindo o uso da métrica de esforço homem-hora. Essa diretriz também vem sendo seguida pelos governos estaduais e municipais. O Tribunal de Contas da União (TCU) tem publicado vários acórdãos que recomendam a utilização da métrica Ponto de Função Não Ajustado em contratos de prestação de serviços de desenvolvimento e manutenção de sistemas. Além disso, a Portaria SLTI/MP nº 31, de 29 novembro de 2010, recomenda o uso da métrica Ponto de Função para os órgãos integrantes do Sistema de Administração dos Recursos de Informação e Informática (SISP), bem como a adoção do Roteiro de Métricas de Software do SISP na contratação de serviços de desenvolvimento e manutenção de soluções de software (SISP, 2012).

O Roteiro de Métricas de Software do SISP foi desenvolvido por um comitê formado pela Secretaria de Logística e Tecnologia da Informação, do Ministério do Planejamento, composto por representantes de diversas instituições públicas. O comitê foi responsável por elaborar, em forma de guia, um documento que definisse uma forma de mensuração em pontos de função dos serviços de desenvolvimento e manutenção de software que são vivenciados pelas instituições integrantes do SISP, e que são considerados pelo IFPUG como não mensuráveis, com isso obedecendo as recomendações dos órgãos de controle do governo brasileiro em utilizar APF nos contratos de todos os serviços de desenvolvimento e manutenção de software. O roteiro aborda uma solução onde a função impactada é analisada de acordo com a técnica do IFPUG, e depois de calculados seus pontos de função não ajustados deve ser aplicado um fator de impacto que foi definido para cada tipo de serviço, de acordo com um estudo estatístico realizado pelo comitê, baseado no esforço para se executar cada serviço (SISP, 2012).

Como no Brasil a técnica de análise de pontos de função além de ser utilizada para estimar dados para o planejamento dos projetos de software, também é utilizada como unidade monetária para pagamento desses projetos, as organizações buscaram utilizar abordagens paralelas junto a técnica do IFPUG em seus projetos de manutenção, para se obter uma relação mais justa com o esforço realizado, e conseguir mensurar todos os tipos de manutenções (SISP, 2012).

É importante ressaltar que a métrica Ponto de Função foi concebida como uma medida de tamanho funcional para projetos de desenvolvimento e de melhoria (manutenção evolutiva) de software. No entanto, os projetos de software não estão limitados a projetos de desenvolvimento e de melhoria. Desta forma, torna-se essencial a definição de métricas para dimensionar o tamanho de outros tipos de projetos de manutenção, os quais são itens não mensuráveis pelo CPM (SISP, 2012).

#### **2.2.4. APF na Contratação de Software**

A dinâmica do mercado atual impõe às organizações um ritmo de inovação e excelência cada vez mais agressivo. Como resultado a esta pressão as empresas buscam focar esforços nas áreas estratégicas e relacionadas aos seus negócios principais. Conseqüentemente as demais atividades tornam-se passíveis de terceirização, inclusive o desenvolvimento de software. Neste caso podem existir vários níveis de terceirização: da codificação do software até ao *outsourcing* de toda a equipe de sistemas (VAZQUEZ et al., 2004).

O fato de parte ou todo trabalho envolvido no desenvolvimento e manutenção de sistemas estar terceirizado não elimina a necessidade de medição, muito pelo contrário. Tanto no corpo de conhecimento em gerência de projetos (PMBOK) do *Project Management Institute* (PMI) quanto no *Capability Maturity Model* (CMM) do *Software Engineering Institute* (SEI) a aplicação de métricas é um aspecto determinante no relacionamento entre a empresa que contrata e a empresa contratada (SIMÕES, 2004).

Todos os profissionais, representando organizações nos papéis de contratantes ou contratadas em projetos de desenvolvimento de software, podem relatar pelo menos uma experiência em que o relacionamento entre as partes foi prejudicado em função do projeto exceder as expectativas iniciais de custo, escopo ou prazo. O erro na escolha da melhor forma de contratação do fornecedor pode agravar ainda mais a situação desse relacionamento (VAZQUEZ et al., 2010).

A seguir são apresentadas três modalidades de contratação de desenvolvimento de software mais comuns: homem/hora, preço global fixo e preço unitário; com suas vantagens e desvantagens do ponto de vista do cliente e como a técnica da análise de pontos de função pode ser utilizada para obter benefícios em cada caso (VAZQUEZ et al., 2004):

- **Modalidade de Contratação Homem/Hora:** Na contratação de homem/hora, também conhecida como *body shopping* ou *time and material*, a empresa contrata profissionais para alocação no desenvolvimento do software, usualmente conjuntamente com sua equipe própria, nem sempre com apenas um fornecedor de mão de obra e utilização de sua própria infraestrutura logística. A remuneração da empresa contratada é calculada com base no nível de qualificação e experiência dos profissionais alocados, nas horas apuradas e em outras eventuais despesas.

Um contrato dessa modalidade é de simples administração e apresenta uma grande flexibilidade tanto para a empresa contratante quanto para a contratada, considerando que para mudança dos requisitos, não é necessária uma nova renegociação de contrato com o prestador de serviço, porém, um aumento de escopo acarretará em um aumento de esforço (horas) e também de preço, uma vez que a gerência do escopo e dos requisitos é responsabilidade direta da contratante.

O aspecto mais crítico dessa modalidade de contratação é que a empresa contratante é responsável por gerenciar todo o serviço, inclusive a produtividade dos contratados. Isto exige um nível de competência que pode não estar disponível internamente. Além disto a remuneração do prestador de serviço não está vinculada aos resultados produzidos, mas apenas na quantidade de horas apuradas. Não há nenhum estímulo ao prestador de serviço para a manutenção ou o aumento dos níveis de produtividade e qualidade.

Neste modelo pode-se aplicar a técnica da análise de pontos de função para monitorar a produtividade da equipe a partir dos dados de esforço (horas) e resultados (pontos de função do produto entregue), e indicadores de qualidade através dos dados de quantidade de defeitos por pontos de função entregue.

- **Modalidade de Contratação Preço Global Fixo (ou Projeto Fechado):** Esta modalidade privilegia a abordagem de projeto, com um início e fim bem definidos. Exige maior nível de organização tanto por parte da empresa contratante quanto do fornecedor. Quanto melhor definidos estiverem os requisitos, menor a chance de atritos entre cliente e fornecedor.

Um fator que complica a utilização desta abordagem é assumir que os requisitos não mudam após o início do projeto. Assim como o mundo é dinâmico, os requisitos também o são. Quanto maior a duração do projeto, mais provável que haja alteração nos requisitos. E é difícil estimar quanto estas alterações afetam o orçamento proposto originalmente pelo fornecedor. Porém, é quase certo que seja necessária uma nova negociação. Se este for o caso, dificilmente o cliente irá obter as mesmas condições originais; pois dependendo do momento em que o projeto esteja não haverá concorrência e tão pouco uma unidade para comparar o preço originalmente cobrado com o preço cobrado pelas novas características solicitadas.

Em contratação de projetos fechados o fornecedor para absorver os riscos de requisitos mal definidos ou de crescimento de escopo durante o ciclo de vida de desenvolvimento do projeto, acrescenta margens de segurança no valor da proposta comercial.

Uma aplicação da análise de pontos de função nestes casos é sua utilização como um fator de normalização. A empresa pode dimensionar o projeto originalmente contratado e, com base no resultado da contagem dos pontos de função, pode calcular o valor unitário cobrado pela empresa contratada. O serviço adicional para o qual se solicitou uma proposta pode também ser dimensionado em pontos de função e aplicado o valor unitário calculado para o projeto original. Se o preço resultante for significativamente diferente do valor cobrado pelo fornecedor, a empresa contratante pode questionar quais são as causas para esta discrepância.

Outra aplicação é como ferramenta na gestão de escopo do projeto. Pode-se medir o tamanho funcional do projeto em cada fase do seu desenvolvimento e obter de maneira objetiva como e quanto o escopo está variando.

- **Modalidade de Contratação Preço Unitário:** Neste modelo é definida uma remuneração para o fornecedor sobre elementos do projeto. Este elemento pode assumir várias formas: tela, relatório, tabela, caso de uso, linha de código, *stored* procedure ou ponto de função. Em tese é um modelo que procura equilibrar as deficiências da contratação por homem/hora e por preço global fixo. Neste caso, a produtividade passa a ser atribuição do fornecedor, pois há o risco de prejuízo caso haja atraso na produção dos elementos. Por outro lado, no caso de um aumento de escopo dos requisitos,

mais elementos deverão ser construídos para o projeto e o fornecedor será remunerado por isso, aumentando o custo do projeto para o cliente.

O grande desafio dessa abordagem é encontrar um elemento que possa ser reconhecido de maneira inequívoca, uniforme e consistente por ambos cliente e fornecedor e que também possua uma natureza não excessivamente técnica, daí que é inserido como uma solução a esses desafios a análise de pontos de função como uma ferramenta para medir projetos de software.

A aplicação da análise de pontos de função como solução alternativa aos desafios expostos apresenta uma série de vantagens. Sua utilização é bastante semelhante ao caso onde telas, arquivos e relatórios são utilizados como base para a medição do serviço. Contudo, em pelo menos dois aspectos ela facilita o relacionamento entre as partes.

- ✓ Pontos de Função Como Unidade Padrão: A análise de pontos de função é uma técnica padrão de medição funcional. Centenas de empresas e profissionais de todo o mundo participam das ações do IFPUG, que é a organização responsável pela manutenção e evolução da técnica. Estas ações procuram garantir a consistência e a uniformidade da aplicação da técnica. Experiências de governos e empresas de todo mundo relatam casos de sucesso em sua aplicação.
- ✓ Pontos de Função Facilitando a Comunicação: O vocabulário da análise de pontos de função utiliza terminologia e define objetos de contagem que independem da tecnologia utilizada para o desenvolvimento do software. A eliminação destas tecnicidades facilita a compreensão entre as partes e é um importante fator de alavancagem da comunicação entre elas. Não é relevante para a análise de pontos de função se determinado agrupamento lógico de dados foi implementado em dois, três ou quantos arquivos forem convenientes. Não é relevante se no espaço de uma tela não couberam todos os campos necessários para a conclusão de um processo elementar do negócio, o que é relevante é a perspectiva do negócio como entendido e validado pelo usuário.

Segundo Aguiar e Baklizky, o dimensionamento de elementos funcionais nos serviços de desenvolvimento e manutenção de software é o método mais adequado de contratação, considerando que ele é orientado para resultados, pode ser entendido e

validado pelo cliente, as medidas funcionais de tamanho podem ser padronizadas, e pode ser objeto de benchmark (AGUIAR; BAKLIZKY, 2010).

A causa raiz na disputa entre contratantes e contratados está associada aos termos contratuais ambíguos ou não entendidos. Contratos e avaliações de desenvolvimento de software baseado na métrica de Pontos de Função podem minimizar as chances de conflito e litigação (JONES, 2001).

Para Vazquez et al., a APF, por usar objetos de contagem que independem da tecnologia utilizada, eliminando tecnicidades e facilitando a compreensão entre as partes, torna-se um importante fator para melhoria da comunicação, facilitando também a comunicação efetiva com todos que possam ser afetados pelo sistema, o que é excelente em termos de melhoria da governança (VAZQUEZ et al., 2010).

Apesar de a adoção dos contratos baseados no valor unitário de pontos de função representar um importante avanço para resguardar os interesses de ambas as partes, há necessidade de cuidados na formulação das cláusulas que vão regular a prestação dos serviços, considerando a complexidade dos processos de estimativa e medição. É recomendável o estabelecimento de um anexo ao contrato, esclarecendo a perspectiva do contratante quanto a alguns aspectos da técnica que podem causar diferenças nas interpretações, evitando problemas durante a execução (ARAUJO, 2008).

É conveniente lembrar que nenhuma técnica elimina a necessidade de cuidado na definição do objeto. Como observa o ISBSG, nenhuma cláusula com penalidades será capaz de cuidar de especificação de requisitos incompleta, problemas de escopo e arquitetura errada, já que será, em geral, impossível estabelecer quem é o responsável pelas falhas (ISBSG, 2013).

Os contratos de desenvolvimento e manutenção de sistemas baseados em pontos de função continuam em constante crescimento no Brasil por organizações governamentais desde a década de 1990. Os *cases* de sucesso e a grande fatia de mercado que é representada pelos órgãos do governo fez a técnica ganhar, também, uma popularidade entre as empresas privadas. Nesse tipo de contratação, o pagamento da empresa contratada é realizado por meio do dimensionamento do produto entregue, utilizando a valor do ponto de função estabelecido no contrato (HAZAN, 2010).

A análise de pontos de função tem sido muito utilizada como unidade monetária (R\$/PF) nesses contratos de desenvolvimento e manutenção de sistemas pelas organizações governamentais brasileiras. Este tipo de contrato permite o melhor balanceamento de riscos entre contratante e contratada, sendo recomendado pela

Instrução Normativa - IN04 e pelos Órgãos de Controle do Governo Brasileiro (HAZAN, 2010).

O valor do ponto de função não ajustado, utilizado como unidade monetária em um grande número de contratos de desenvolvimento e manutenção de software no Brasil, deve ser definido de acordo com a complexidade de negócio dos sistemas e de seus requisitos não funcionais acordados na negociação contratual, considerando que essas características influenciam diretamente na produtividade da equipe e conseqüentemente nos custos dos projetos.

## **2.3. Common Software Measurement International Consortium – COSMIC**

### **2.3.1. Definições, Conceitos e Regras**

Inicialmente chamado ‘*Full Function Point - FFP*’, em sua versão 1.0, foi proposto em 1997 por um grupo de pesquisadores da universidade de Quebec, com o objetivo de ser uma extensão da APF, para capturar o tamanho funcional de aplicações em tempo real e de aplicações técnicas e de sistema. Os testes de campo mostraram que o FFP era também adequado para medir o tamanho funcional de aplicações do tipo MIS (softwares aplicativos de negócio), levando a resultado semelhantes (OLIGNAY et. al., 1998).

Em 1998, o grupo de FFP somou seus esforços ao trabalho de um grupo de especialistas em métricas funcionais chamado COSMIC - *Common Software Measurement International Consortium*, propondo os princípios de uma segunda geração de técnicas para a medição de tamanho funcional. Estes esforços resultaram no primeiro teste de campo disponível ao público da técnica de medição COSMIC-FFP versão 2.0, publicado em outubro de 1999. Da versão 2.0 em diante, a técnica também foi desenhado para garantir a sua total conformidade com a norma ISO / IEC 14143-1: 1998 (e, posteriormente, ISO / IEC 14143-1: 2007). A partir da versão 3.0, o nome da técnica foi simplificado de ‘COSMIC-FFP’ para ‘COSMIC’ (COSMIC, 2007).

COSMIC, *Common Software Measurement International Consortium* é uma iniciativa voluntária de um grupo internacional de especialistas em medição de software, praticantes e acadêmicos, da Ásia/Pacífico, Europa e América do Norte. Os objetivos originais do projeto COSMIC eram desenvolver, testar, trazer e buscar a aceitação do mercado para uma nova técnica de medição de tamanho com o objetivo de apoiar a estimativa e medições de desempenho. Tais objetivos foram agora atingidos e a

técnica é aceita em um crescente número de organizações dos setores públicos e privados em todo o mundo (COSMIC, 2007).

Depois que os princípios da técnica COSMIC foram estabelecidos em 1999, testes de campo foram conduzidos com sucesso, em 2000, envolvendo diversas organizações e instituições acadêmicas internacionais. Em 2001 foi iniciado o processo de desenvolvimento de um Padrão Internacional para a técnica COSMIC. O padrão foi aprovado em dezembro de 2002 e publicado pela ISO no início de 2003 como ISO/IEC 19761 (COSMIC, 2007).

O COSMIC é uma técnica padrão para a medição do tamanho funcional de software, aderente aos domínios funcionais normalmente denominados de software para “aplicações de negócio” (ou ‘MIS’), software ‘real-time’ e combinações dessas duas categorias (COSMIC, 2007).

Na visão proposta pelo COSMIC o software é parte de um produto ou serviço concebido para satisfazer os requisitos funcionais do usuário. Em uma perspectiva de alto nível os requisitos funcionais do usuário podem ser alocados ao hardware, ao software, ou uma combinação de ambos. Quando alocados ao software eles podem ser distribuídos a pedaços de software em diferentes níveis de abstração, cooperando para fornecer a funcionalidade necessária ao produto ou serviço do requisito (XUNMEI et. al., 2006).

A técnica COSMIC não tenta medir todos os aspectos possíveis da funcionalidade que poderiam ser considerados como contribuintes do ‘tamanho’ do software. Nem a influência da complexidade sobre o tamanho do software, nem a influência do número de atributos de dados por movimentação de dados são capturados por esta técnica de medição (COSMIC, 2007).

A técnica de medição COSMIC envolve a aplicação de um conjunto de modelos, princípios, regras e processos aos Requisitos Funcionais do Usuário de um dado pedaço de software. O resultado é um número, o valor de uma quantidade, representando o tamanho funcional do pedaço de software (COSMIC, 2009).

Para apoiar seu processo de medição, a técnica define seus conceitos, definições e regras a partir de um conjunto de princípios estabelecidos em dois modelos de software COSMIC – o ‘Modelo de Contexto de Software’ e o ‘Modelo Genérico de Software’ (COSMIC, 2009).

### **2.3.1.1. O Modelo de Contexto de Software COSMIC**

Um pedaço de software a ser medido com a técnica COSMIC deve ser cuidadosamente definido no escopo da medição, e tal definição deve levar em conta, em seu contexto, qualquer outro software e/ou hardware com o qual o mesmo interaja. Este Modelo de Contexto de Software apresenta os seguintes princípios e conceitos necessários a esta definição (COSMIC, 2009):

- a) O Software é delimitado pelo hardware;
- b) O software é normalmente estruturado em camadas;
- c) Uma camada pode conter um ou mais pedaços de software ‘pares’ distintos e qualquer pedaço de software pode ser composto de componentes pares distintos;
- d) Qualquer pedaço de software a ser medido deverá ser definido por seu escopo de medição, o qual deve estar integralmente contido em uma única camada;
- e) O escopo de um pedaço de software a ser medido depende do propósito da medição;
- f) Os usuários funcionais de um pedaço de software devem ser identificados a partir dos requisitos funcionais do usuário do pedaço de software a ser medido, como fontes e/ou destinos pretendidos para os dados;
- g) Um pedaço de software interage com os seus usuários funcionais por meio de movimentações de dados através de uma fronteira, e o pedaço de software pode mover dados de e para o armazenamento persistente dentro da fronteira;
- h) Os RFU do software podem ser expressos a diferentes níveis de granularidade;
- i) O nível de granularidade no qual as medições devem ser normalmente efetuadas é o dos processos funcionais;
- j) Se não for possível medir no nível de granularidade dos processos funcionais, nesse caso os RFU do software devem ser medidos através de uma abordagem de aproximação e escalonados para o nível de granularidade dos processos funcionais.

### **2.3.1.2. O Modelo Genérico de Software COSMIC**

Uma vez interpretados os RFU do software a ser medido em termos do Modelo de Contexto de Software, aplicamos o Modelo Genérico de Software aos RFU, para identificar os componentes da funcionalidade que serão medidos. Este Modelo Genérico

de Software assume que os seguintes princípios gerais são verdadeiros para qualquer software que possa ser medido com a técnica (COSMIC, 2009):

- a) O software recebe dados de entrada dos seus usuários funcionais, gerando saídas e/ou outros resultados para os usuários funcionais;
- b) Os requisitos dos usuários funcionais de um pedaço de software a ser medido podem ser mapeados para processos funcionais distintos;
- c) Cada processo funcional consiste de subprocessos;
- d) Um subprocesso pode ser uma movimentação de dados ou uma manipulação de dados;
- e) Cada processo funcional é disparado por um movimento de dados do tipo Entry proveniente de um usuário funcional, o qual informa ao processo funcional que o usuário funcional identificou um evento;
- f) Uma movimentação de dados movimenta um único grupo de dados;
- g) Um grupo de dados consiste de um único conjunto de atributos de dados que descreve um único objeto de interesse;
- h) Há quatro tipos de movimentação de dados. Uma Entry movimenta um grupo de dados para dentro do software, a partir de um usuário funcional. Uma Exit movimenta um grupo de dados para fora do software, em direção a um usuário funcional. Um Write movimenta um grupo de dados do software para o armazenamento persistente. Um Read movimenta um grupo de dados do armazenamento persistente para o software;
- i) Um processo funcional deve incluir pelo menos uma movimentação de dados Entry e uma movimentação de dados Write ou Exit, ou seja, deve incluir no mínimo duas movimentações de dados;
- j) Como uma aproximação para fins de medição, os subprocessos de manipulação de dados não são medidos separadamente; assume-se que a funcionalidade de qualquer manipulação de dados será considerada na movimentação de dados com a qual a mesma esteja associada.

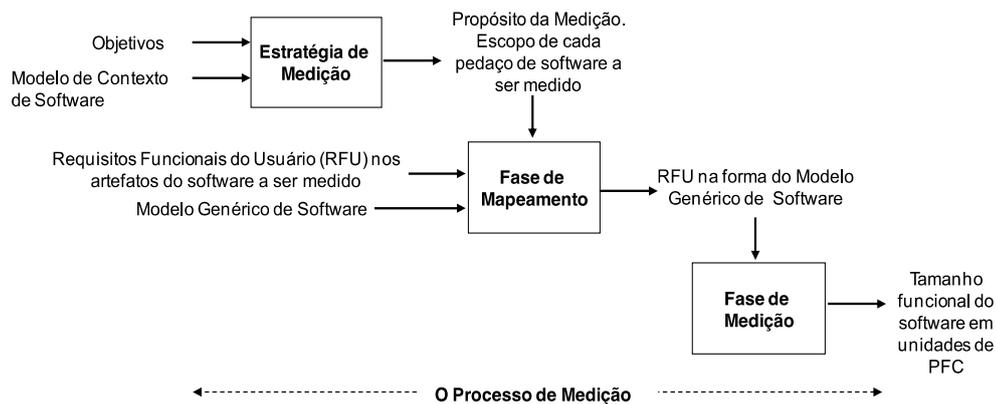
### **2.3.1.3. Processo de Medição**

O processo geral de medição COSMIC consiste de três fases (COSMIC, 2009):

- A Estratégia de Medição, na qual o Modelo de Contexto de Software é aplicado ao software a ser medido;

- A Fase de Mapeamento, na qual o Modelo Genérico de Software é aplicado ao software a ser medido;
- A Fase de Medição, na qual as medições de tamanho propriamente ditas são obtidas;

A relação entre essas três fases do processo de medição COSMIC é apresentada na Figura 2.4 (COSMIC, 2007).



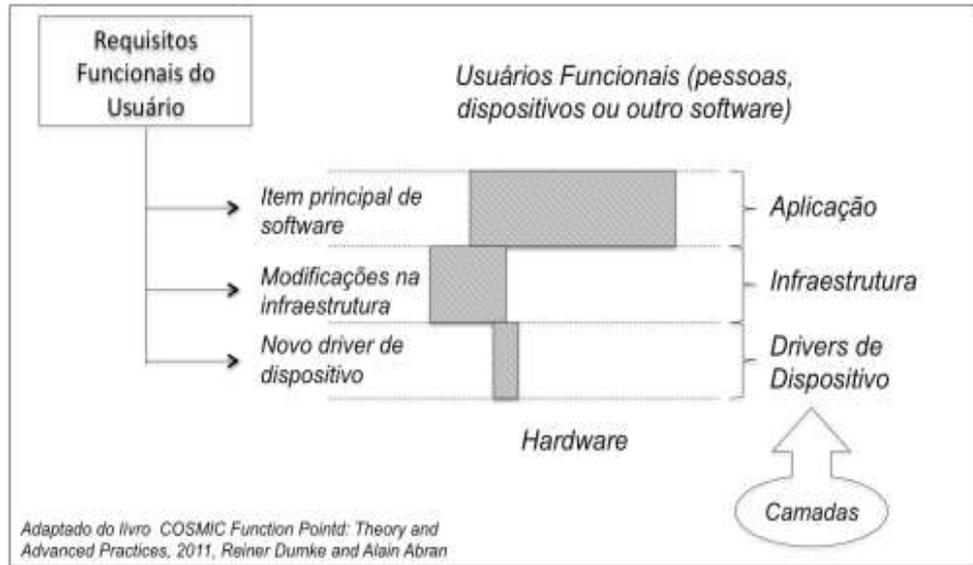
**Figura 2.4 – Estrutura da Técnica COSMIC**

### 2.3.1.3.1. Estratégia de Medição

Nessa fase são determinados os seguintes parâmetros-chave da medição de tamanho funcional de software que devem ser considerados antes de iniciar a medição (COSMIC, 2009):

- **Propósito da medição:** é uma afirmação que define por que uma medição é requerida e para quê o resultado será utilizado. O propósito ajuda o medidor a determinar o escopo a ser medido, os usuários funcionais, a fase do ciclo de vida do software que será realizada a medição, e a exatidão requerida para a medição. Estes dois últimos pontos determinarão o nível de granularidade no qual os RFU serão medidos.
- **Escopo da medição:** é o conjunto de Requisitos Funcionais do Usuário a serem incluídos em um exercício específico de medição funcional de tamanho. Para o COSMIC, o termo 'escopo' relaciona-se a um pedaço individual de software cujo tamanho deve ser medido separadamente. Como regras do escopo, podemos dizer que ele deve ser derivado do propósito da medição, e que não deve contemplar mais do que uma camada do software a ser medido. Entende-se por 'camada' uma partição resultante da divisão

funcional de um sistema de software que, juntamente com o hardware, forma um sistema computacional completo. A Figura 2.5 descreve bem como os requisitos de um escopo deve ser definidos para uma camada específica (DUMKE; ABRAN, 2011);



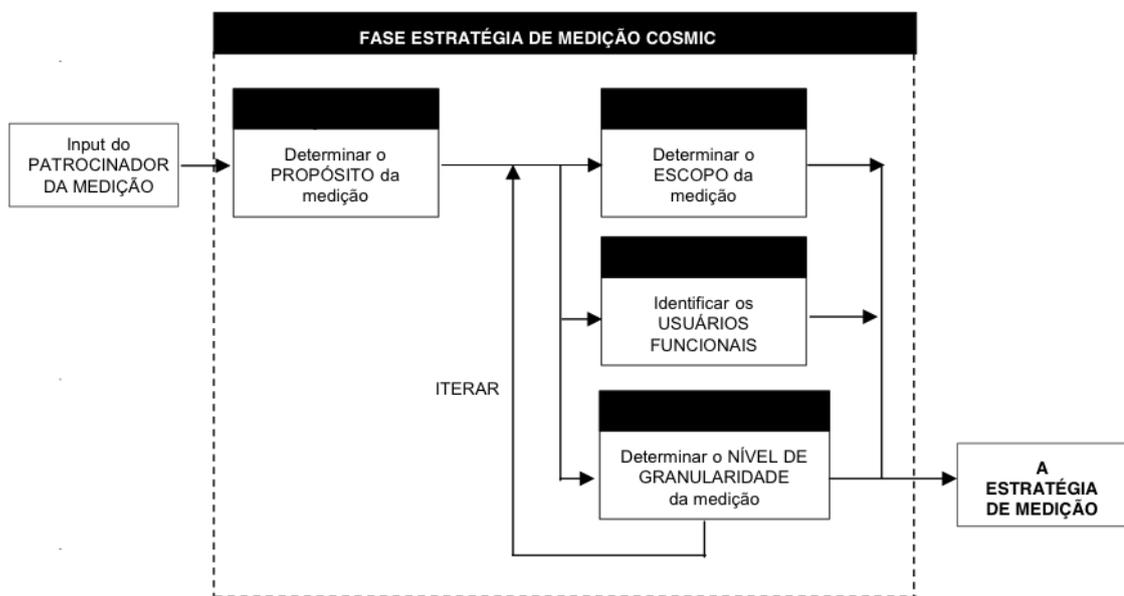
**Figura 2.5 – Requisitos Funcionais do Usuário por Camadas**

- **Usuários funcionais:** é uma fonte e/ou um destino pretendido para os dados nos Requisitos Funcionais do Usuário de um pedaço de software. Uma vez identificados os usuários funcionais é imediata a identificação da fronteira, já que a mesma situa-se entre o pedaço de software medido e seus usuários funcionais. É ignorado qualquer outro hardware ou software naquele espaço intermediário. Fronteira é definida como uma interface conceitual entre o software sendo medido e os seus usuários funcionais. A fronteira de um pedaço de software é o limite conceitual entre o referido pedaço e o ambiente no qual o mesmo opera, conforme percebido externamente segundo a perspectiva de seus usuários funcionais. A fronteira permite que o medidor distinga, sem ambiguidade, o que está incluído dentro do software medido daquilo que é parte do ambiente operacional do referido software;
- **Nível de granularidade:** Qualquer nível de expansão da descrição de um pedaço de software (p.ex., uma declaração de seus requisitos, ou uma descrição da estrutura do pedaço de software) tal que cada evolução desse nível sua descrição estará mais detalhada e uniforme. Os medidores devem estar cientes de que, quando os requisitos evoluem cedo na vida de um

projeto de software, a qualquer momento partes da funcionalidade requerida pelo mesmo terão sido documentadas em diferentes níveis de granularidade. Na prática, a documentação do software contendo requisitos funcionais do usuário descreve a funcionalidade em vários níveis de granularidade, especialmente quando a documentação ainda está evoluindo em cada fase do ciclo de vida do desenvolvimento.

A consideração cuidadosa dos quatro elementos do processo pertinente à estratégia de medição antes de começar uma medição deve garantir que o tamanho resultante possa ser interpretado adequadamente. A Figura 2.6 ilustra como deve ser determinado os quatro elementos abaixo (COSMIC, 2009):

1. Estabelecer o propósito da medição;
2. Definir o escopo total do pedaço de software a ser medido e escopo dos pedaços a serem medidos separadamente, considerando as camadas e componentes pares da arquitetura de software;
3. Estabelecer os usuários funcionais do pedaço de software a ser medido;
4. Estabelecer o nível de granularidade dos artefatos de software a serem medidos e como, se necessário, escalonar os tamanhos para o nível padrão de granularidade dos processos funcionais.

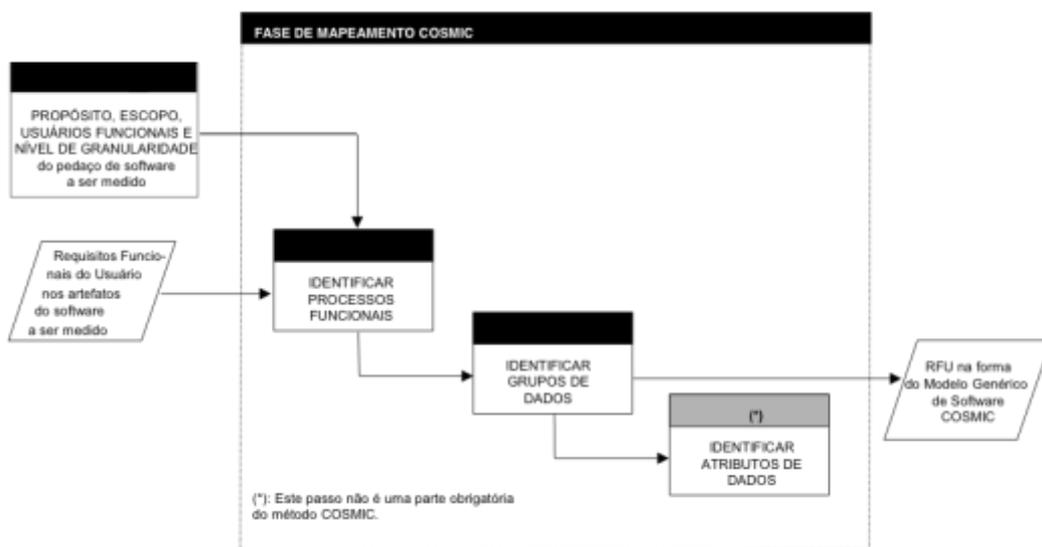


**Figura 2.6 – Processo de Determinação de uma Estratégia de Medição**

### 2.3.1.3.2. Fase de Mapeamento

Esta fase determina as regras e o método para o processo de mapeamento. O Modelo Genérico de Software COSMIC deve ser aplicado aos requisitos funcionais do usuário de cada pedaço de software de um escopo de medição definido. Aplicar o Modelo Genérico de Software COSMIC diz respeito à identificação do conjunto de eventos disparadores percebidos por cada um dos usuários funcionais identificados nos RFU, seguido pela identificação dos correspondentes processos funcionais, objetos de interesse, grupos de dados e movimentações de dados que devem ser providos como resposta a esses eventos (COSMIC, 2009).

O método geral para mapear software para o Modelo Genérico de Software COSMIC é resumido na Figura 2.7 (COSMIC, 2009):



**Figura 2.7 – Método geral do processo de mapeamento de software COSMIC**

As atividades a serem executadas na fase de mapeamento, descritas na Figura 2.6 são determinadas a partir do conceito de cada um dos elementos abaixo (COSMIC, 2009):

- **Processo Funcional:** é um componente elementar de um conjunto de Requisitos Funcionais do Usuário compreendendo um conjunto de movimentações de dados único, coeso e independentemente executável. É disparado por uma movimentação de dados (uma Entry) de um usuário funcional que informa ao pedaço de software que o usuário funcional identificou um evento disparador. Estará concluído quando tiver executado tudo o que é requerido para ser feito em resposta ao evento disparador, que é um evento que faz com que um usuário funcional do pedaço de software

inicie um ou mais processos funcionais. Um processo funcional deve estabelecer as seguintes regras:

- ✓ Deve ser derivado de pelo menos um Requisito Funcional do Usuário identificável dentro do escopo acordado;
  - ✓ Deve ser executado quando um evento disparador identificável ocorre;
  - ✓ Pode ser disparado por mais de um evento. Um evento específico pode disparar um ou mais processos funcionais que executem em paralelo;
  - ✓ Deve compreender pelo menos duas movimentações de dados: uma *Entry*, mais uma *Exit* ou um *Write*;
  - ✓ Deve pertencer completamente ao escopo de medição de um pedaço de software em uma e somente uma camada;
  - ✓ No contexto de software ‘*real-time*’, um processo funcional deve ser considerado concluído quando o mesmo já fez tudo o que é requerido como resposta ao evento disparador e espera até receber a próxima *Entry* disparadora;
- **Grupo de Dados:** é um conjunto de atributos de dados distinto, não vazio, não ordenado e não redundante, onde cada atributo de dado incluído descreve um aspecto complementar do mesmo objeto de interesse. Objeto de Interesse é uma entidade identificada pelo ponto de vista dos Requisitos Funcionais do usuário, onde o software deve processar e/ou gravar dados no armazenamento persistente. O armazenamento persistente é o armazenamento que permite a um processo funcional armazenar um grupo de dados além da vida do processo funcional, e/ou de onde um processo funcional pode recuperar um grupo de dados armazenado por outro processo funcional, por uma ocorrência anterior do mesmo processo funcional, ou por algum outro processo. Uma vez identificado, cada grupo de dados candidato deve obedecer aos seguintes princípios:
    - ✓ Cada grupo de dados identificado deve ser único e distinguível através de sua coleção única de atributos de dados;
    - ✓ Cada grupo de dados deve ser diretamente relacionado a um objeto de interesse nos Requisitos Funcionais do Usuário;
    - ✓ Um grupo de dados deve ser materializado dentro do sistema de computador que suporta o software.

- **Atributo de Dados:** é a menor parcela de informação, dentro de um grupo de dados identificado, que possui um significado segundo a perspectiva dos Requisitos Funcionais do Usuário do software. Na versão atual da técnica, não é obrigatório identificar os atributos de dados. Contudo, pode ser útil analisar e identificar os atributos de dados no processo de distinguir grupos de dados e objetos de interesse.

### 2.3.1.3.3. Fase de Medição

A fase de medição é definida por conceitos e regras que apoiam os processos de medição geral da técnica, onde a partir dos Requisitos Funcionais do Usuário é definido o tamanho funcional do software medido, conforme exibido na Figura 2.8 (COSMIC, 2009):

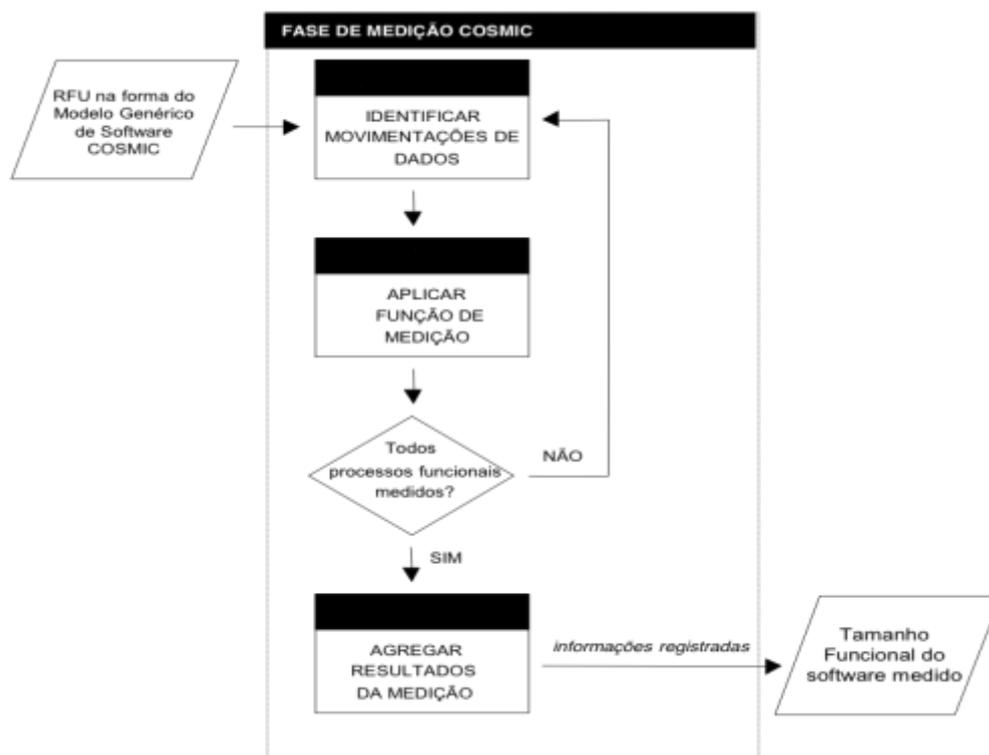


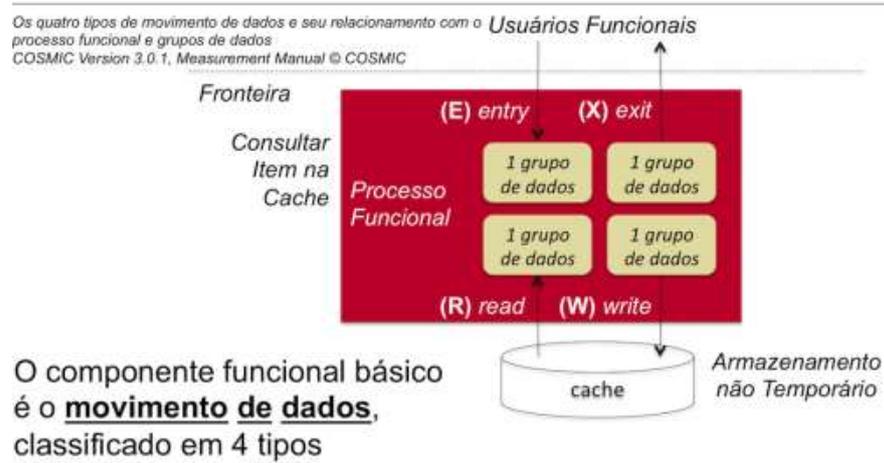
Figura 2.8 – Método geral do processo de medição de software COSMIC

O processo da fase de medição é definido pelas atividades exibidas na Figura 2.8, onde as definições e princípios de seus elementos são os seguintes (COSMIC, 2009):

- **Movimentação de Dados:** é um componente funcional básico que movimenta um único tipo de grupo de dados. Há quatro tipos de movimentações de dados: *Entry*, *Exit*, *Read* e *Write*. Para fins de medição,

considera-se que todos os tipos de movimentação de dados inclui certas manipulações de dados associadas. É considerada como manipulação de dados qualquer coisa que aconteça aos dados que não seja a movimentação dos mesmos para dentro ou para fora de um processo funcional, ou entre um processo funcional e o armazenamento persistente. Todas as manipulações de dados em um processo funcional devem estar associadas aos quatro tipos de movimentações de dados. Os tipos de movimentação de dados são definidos como:

- ✓ *Entry (E)*: é uma movimentação de dados que movimenta um grupo de dados de um usuário funcional através da fronteira para dentro do processo funcional no qual o mesmo é requerido;
- ✓ *Exit (X)*: é uma movimentação de dados que movimenta um grupo de dados de um processo funcional através da fronteira na direção do usuário funcional que o requisita;
- ✓ *Read (R)*: é uma movimentação de dados que movimenta um grupo de dados do armazenamento persistente para dentro do processo funcional que dele necessita;
- ✓ *Write (W)*: é uma movimentação de dados que movimenta um grupo de dados do interior de um processo funcional para o armazenamento persistente.



**Figura 2.9 – Movimentação de Dados**

- **Função de Medição:** A função de medição COSMIC é uma função matemática que atribui um valor ao seu argumento com base no padrão de medição COSMIC. O argumento da função de medição é a movimentação

de dados. O padrão de medição COSMIC, 1 PFC (Ponto de Função COSMIC) é definido como o tamanho de uma movimentação de dados. De acordo com a função de medição, cada instância identificada de uma movimentação de dados (Entry, Exit, Read ou Write) que precisa ser incluída, alterada ou excluída, recebe um tamanho numérico de 1 PFC;

- **Resultado da Medição:** consiste em agregar os resultados da função de medição em um único valor do tamanho funcional, aplicando a função de medição a todas as movimentações de dados. Este passo é realizado de acordo com as seguintes regras:

- ✓ Para qualquer processo funcional, os tamanhos funcionais das movimentações de dados individuais devem ser agregados em um único valor de tamanho funcional em unidades de PFC, somando-os aritmeticamente:

$$\text{Tamanho (processo funcional)} = \Sigma \text{tamanho(Entries)} + \Sigma \text{tamanho(Exits)} + \Sigma \text{tamanho(Reads)} + \Sigma \text{tamanho(Writes)};$$

- ✓ O tamanho de um pedaço de software em um escopo definido deve ser obtido pela agregação dos tamanhos dos processos funcionais referentes ao pedaço de software;
- ✓ Os tamanhos de pedaços de software dentro de camadas podem ser somados somente se forem medidos no mesmo nível de granularidade de processo funcional de seu RFU;
- ✓ Os tamanhos de pedaços de software dentro de qualquer camada, ou de diferentes camadas devem ser somados somente se isso fizer sentido para o propósito da medição.

### 2.3.2. Projetos de Melhoria Utilizando COSMIC

Uma mudança funcional no software existente é interpretada na técnica COSMIC como qualquer combinação de inclusões de novas movimentações de dados, ou de alterações ou exclusões de movimentações de dados existentes em um processo funcional e reconhecidas pelo usuário. O termo ‘melhoria’ é frequentemente utilizado para o que a técnica chama de ‘mudança funcional’ (COSMIC, 2009).

Para qualquer processo funcional, o tamanho funcional das alterações em seus Requisitos Funcionais do Usuário deve ser agregado a partir dos tamanhos das

movimentações de dados que tiverem sido incluídas, alteradas ou excluídas no processo funcional, produzindo como resultado o tamanho da mudança em unidades de PFC, de acordo com a seguinte fórmula (COSMIC, 2009):

- Tamanho (Mudança(processo funcional)) =  $\Sigma$  tamanho (movimentações de dados incluídas) +  $\Sigma$  tamanho (movimentações de dados alteradas) +  $\Sigma$  tamanho (movimentações de dados excluídas).

O tamanho da mudança de um pedaço de software de um escopo definido deve ser obtido agregando-se os tamanhos de todas as mudanças em todos os processos funcionais correspondentes ao pedaço. Os tamanhos de mudanças em pedaços de software dentro de qualquer camada, ou de diferentes camadas devem ser somados somente se isso fizer sentido para o propósito da medição. (COSMIC, 2009).

Depois da alteração funcional de um pedaço de software, seu novo tamanho total é igual ao tamanho original mais o tamanho funcional de todas as movimentações de dados incluídas, menos o tamanho funcional de todas as movimentações de dados excluídas. As movimentações de dados alteradas não exercem influência sobre o tamanho do pedaço de software, pois existem antes e depois das modificações serem realizadas (COSMIC, 2009).

### 2.3.3. Aderência a Alguns Tipos de Sistema

A métrica COSMIC foi desenvolvida para trabalhar com o tamanho funcional de diversos tipos de software, e mensurar o tamanho baseada nas funcionalidades entregues para o usuário, possuindo uma visão de usuário mais abrangente que as outras métricas (CALAZANS, 2003).

A técnica COSMIC para medição de tamanho funcional foi projetado para ser aplicável à funcionalidade de software dos seguintes domínios (COSMIC, 2007):

- **Software aplicativo de negócio:** normalmente necessário para apoiar a administração de negócios tais como: bancos, seguros, contabilidade, pessoal, compras, distribuição ou manufatura. Tal tipo de software é muitas vezes caracterizado como 'rico em dados', pois é amplamente dominado pela necessidade de gerenciar uma grande quantidade de dados sobre eventos do mundo real.
- **Software de tempo-real:** com a finalidade de acompanhar ou controlar eventos acontecendo no mundo real. Exemplos seriam: software para

ligações telefônicas e troca de mensagens, software embarcado em dispositivos para controlar máquinas tais como eletrodomésticos, elevadores, motores de automóveis e aeronaves, controle de processos e aquisição automática de dados, assim como software contido no sistema operacional de computadores.

- Híbridos dos tipos de software acima, tais como sistemas em tempo-real para reservas de companhias aéreas e hotéis, por exemplo.

A técnica de medição COSMIC não foi projetada para levar em conta a funcionalidade de software intensivo em matemática, isto é, software caracterizado por algoritmos matemáticos complexos, ou outras regras especializadas e complexas, tais como sistemas especialistas, software para simulação, software para autoaprendizagem, sistemas para previsão do tempo, etc., ou que processe variáveis contínuas, tais como sons de áudio ou imagens de vídeo, por exemplo: software para jogos de computador, instrumentos musicais, etc. (COSMIC, 2007; KAVANAGH, 2003).

#### **2.3.4. Vantagens da aplicação da técnica**

A técnica COSMIC Full Function Points é uma das abordagens mais atuais com relação à mensuração de tamanho de software. O COSMIC surgiu como uma alternativa de mensuração mais exata, com independência de domínio e propondo diferentes medidas para diferentes propósitos, considerando a visão do usuário e do desenvolvedor (CALAZANS, 2003).

A métrica COSMIC possui várias características que a diferenciam das métricas já existentes, tais como (CALAZANS, 2003):

- A possibilidade de aplicar a visão do usuário final e do desenvolvedor. Todas as métricas desconsideram a visão do desenvolvedor e em muitos processos a visão do desenvolvedor é responsável por uma mensuração mais acurada do produto; □
- A identificação de camadas. Considerando que a maioria das tecnologias existentes trabalha com este paradigma. No COSMIC todas as camadas da aplicação podem ser medidas, e não apenas a camada do usuário;
- A flexibilidade. No COSMIC o tamanho das transações é definido de acordo com a quantidade de movimentos de dados, quanto mais

movimentos de dados maior será a transação. Utilizando a abordagem FPA uma EE – entrada externa - pode ter no máximo de 3 a 6 pontos;

- A aplicabilidade de forma mais simples e menos ambígua. Serão analisados apenas os artefatos que definem os processos funcionais, não tendo necessidade de ser analisado o modelo de dados. Considerada objetiva, simples de aprender e fácil usar.

Com base no manual de medição do COSMIC, considera-se que em projetos de melhoria devem ser analisadas as partes das funcionalidades que foram incluídas, alteradas, e excluídas, ou seja, deve ser contado apenas o tamanho das movimentações de dados, que foram mantidas, das transações do software medido. Na APF, esse tipo de projeto é medido com a soma do tamanho de todas as funcionalidades impactadas pela melhoria, e não apenas do que foi alterado nessas funcionalidades. Essa diferença faz com que o COSMIC seja mais eficiente, nas estimativas de esforço para execução dos projetos de melhoria, do que a técnica de análise de pontos de função (COSMIC, 2009; IFPUG, 2010).

## **2.4. Considerações Finais do Capítulo**

Esse capítulo apresentou uma revisão da literatura onde foi abordada a importância das métricas de software em organizações de desenvolvimento e manutenções de software. Ficou evidenciado que os projetos nesse tipo de organização, quando apoiados em sua retaguarda por processos de medição, podem ter como resultado o aumento da qualidade de sua gestão e uma maior satisfação dos clientes. Dentro desse contexto, foram escolhidas duas técnicas de medição de software, exibindo seus conceitos e regras, para que essas referências contribuam para o entendimento do estudo de caso realizado nesse trabalho, e na decisão da técnica a ser selecionada como padrão em uma organização, a partir de suas características, problemas e benefícios.

Considerando a APF como a técnica de medição predominante no mercado brasileiro de desenvolvimento de software, foi mostrado como ela está posicionada no cenário nacional e diante dos tipos de contratação praticados pela terceirização dos serviços de desenvolvimento e manutenção de software como podemos utilizá-la para apoiar em uma melhor gestão desses contratos. Também foram apresentadas as abordagens do IFPUG e da NESMA para projetos de melhoria, e como a proposta da

NESMA possibilita medir de forma mais acurada alguns escopos em que a APF não tem se mostrado eficaz.

Foi apresentado o COSMIC e os benefícios de sua utilização, sua aderência a alguns tipos de sistemas, e a vantagem de utilizar a técnica em projetos de melhoria considerando que em seu escopo são medidas apenas as movimentações impactadas das funcionalidades alteradas.

Na Tabela 2.6 é exibida uma comparação entre as duas técnicas estudadas de algumas das principais características identificadas nesse capítulo.

<b>Característica</b>	<b>APF</b>	<b>COSMIC</b>
Tipos de sistemas aderentes	Software aplicativo de negócio.	Software aplicativo de negócio, Software de tempo-real, e híbridos dos dois tipos.
Medição das transações	Funcionalidades medidas a partir de sua complexidade, definida pela quantidade de tipos de dados e arquivos referenciados.	Funcionalidades medidas a partir da quantidade de movimentações (Entry, Exit, Read e Write).
Projetos de melhoria	É contabilizado na medição o tamanho total da funcionalidade alterada, mesmo que apenas um tipo de dado seja alterado.	É contabilizado na medição apenas a quantidade de movimentações impactadas pela mudança da funcionalidade alterada.
Utilização da técnica pelo mercado brasileiro de medição de software	Técnica de medição de software mais utilizada nas instituições públicas e privadas. Modelo de contratação que utiliza PF como unidade monetária.	Não existe registro de organização utilizando essa técnica como padrão de medição no Brasil.

**Tabela 2.6 – Tabela de características da APF e do COSMIC**

# 3 Estudo de Caso

---

Este capítulo apresenta informações sobre o estudo de caso realizado nessa pesquisa, descrevendo a organização que desenvolveu os softwares medidos, o processo de medição definido para execução de cada projeto, a comparação entre as medições realizadas nas duas técnicas estudadas nesse trabalho, e a demonstração de como foi construída a fórmula de conversão da técnica APF para a técnica COSMIC .

## 3.1. Organização selecionada dos estudos de caso

A organização escolhida para ser aplicado o estudo de caso é uma empresa do setor de tecnologia da informação, considerada de médio porte e com uma quantidade aproximada de 300 colaboradores. Fundada no ano 2005, ela fornece os serviços de Fábrica de Software, Fábrica de Testes, Desenvolvimento e Manutenção sob demanda, Conversão de Legados, Outsourcing e Consultoria.

O crescimento dos serviços prestados levou a organização a investir em melhorias na qualidade de seus processos internos. Tais melhorias foram alcançadas através da adoção e implantação de modelos específicos para qualidade, principalmente na área de desenvolvimento de software. No início deste trabalho, em 2013, a organização possuía certificações ISO 9001:2008 (ISO, 2008), CMMI nível três (SEI, 2010) e o MPS.BR nível C (SOFTEX, 2011).

Nos últimos anos a empresa expandiu continuamente seus negócios, diversificando o foco dos serviços de fábrica de software com investimento nas áreas de consultoria, fábrica de testes e educação. Em 2013 ganhou seu primeiro contrato público de serviços de medição de software, após três anos da criação de sua área de métricas. A métrica padrão utilizada pela empresa é a Análise de Pontos de Função.

## 3.2. Aplicabilidade da Pesquisa

As vantagens da utilização do COSMIC sobre algumas técnicas de medição funcional de software, especialmente sobre a APF, considerada a mais utilizada pela indústria de software, vem despertando interesse em sua adoção como métrica padrão

de algumas organizações em todo o mundo por ser avaliada como precisa e abrangente para medir software e estimar esforço de desenvolvimento. Porém, o autor desse trabalho realizou uma pesquisa informal na conferência anual de medição de software no ano de 2013 – ISMA 8 - *International Software Measurement & Analysis Conference*, entre as maiores empresas atuantes no Brasil em medição de software, e constatou que ainda não existe relatos na comunidade de medição de alguma organização utilizando o COSMIC como técnica padrão.

A adoção em massa da APF nas organizações do setor público no Brasil, decorrente das recomendações de sua utilização pelos órgãos reguladores, e os “cases” de sucesso nessas empresas, fez fortalecer a técnica, também, entre as empresas nacionais da iniciativa privada. A prática da utilização de pontos de função como unidade monetária na contratação de serviços de desenvolvimento e manutenção de software, entre as empresas brasileiras, tornou sua utilização diferenciada na comparação com organizações de outros países, que utilizam a APF para geração de indicadores de gestão e planejamento de seus projetos.

Considerando as vantagens relatadas nesse trabalho da técnica COSMIC, acreditando ser uma técnica de medição promissor, e visando quebrar a resistência criada pelo mercado brasileiro a mudança para uma outra técnica diferente da APF, pelo alto custo de uma migração e a dúvida da aceitação pelas empresas que dele fazem parte, o estudo de caso desse trabalho tem como objetivo realizar contagens detalhadas de softwares aplicativos de negócio nas duas técnicas estudadas, para a construção de uma fórmula de conversão da técnica APF para o COSMIC, buscando facilitar o processo de migração das bases históricas de medição, evitando o custo de uma nova medição de todos os projetos da organização, e com isso contribuindo para uma maior aceitação da técnica COSMIC pelo mercado de desenvolvimento de software.

### **3.3. Coleta de Dados**

Como ponto de partida para os insumos a serem coletados foi definido, conforme acordado com a organização estudada, um processo a ser seguido para as medições dos projetos selecionados, demonstrando as atividades e interações a serem executadas junto as equipes desses projetos, conforme apresentado no Apêndice A. As equipes de desenvolvimento dos projetos tiveram como responsabilidades nesse processo fornecer os insumos necessários para as contagens, apoiar a equipe de métricas, composta apenas

pelo autor desse trabalho, na resolução das dúvidas quanto ao negócio das aplicações e da documentação de projeto disponibilizada para medição, e validar as contagens realizadas.

Foram selecionados quatro projetos de desenvolvimento de dois diferentes clientes da organização escolhida para realização do estudo de caso, todos documentados conforme as práticas do IBM *Rational Unified Process* (RUP, 2013) por exigências contratuais dos clientes. A atuação de mercado desses clientes é um na área governamental e outro no setor elétrico nacional. Para a escolha desses projetos foram estabelecidos os seguintes critérios:

- **Tipo de Sistema:** foram selecionados softwares aplicativos de negócio, considerando a aplicabilidade da técnica de APF apenas para esse tipo de sistema. Nas medições realizadas com a técnica COSMIC apenas a camada de aplicação foi considerada no escopo, seguindo o conceito que essa técnica mede pedaços de software, e apenas essa camada é aplicável em uma comparação com o escopo de um projeto medido em APF.
- **Tipo de Medição:** Dos três tipos possíveis (Projeto de Desenvolvimento, Projeto de Melhoria e Aplicação), foram escolhidos 4 projetos de desenvolvimento, considerando esse ser a grande maioria dos projetos executados pela organização selecionada e pela indústria de software, e por não ter sido disponibilizado nenhum projeto de melhoria já executado para ser utilizado como amostra nesse estudo de caso.
- **Tamanho do Projeto:** Foram selecionados projetos da organização escolhida, já entregues e homologados pelos clientes, e que sua medição estimada realizada a partir da técnica da NESMA (NESMA, 2013) tivesse como resultado tamanho próximo a 300 pontos de função. O objetivo desse tamanho estabelecido era cumprir o prazo planejado para o trabalho, com evidências suficientes para o objetivo esperado.

Nas tabelas das seções 3.3.1, 3.3.2, 3.3.3 e 3.3.4 são descritas as informações das duas medições de cada projeto de software selecionado, identificando os processos elementares para a técnica APF e os processos funcionais para o COSMIC, com seus tamanhos determinados a partir das regras estabelecidas em cada técnica. Com base na prática utilizada pelo mercado de medição e recomendações dos órgãos reguladores brasileiros, foram considerados apenas os pontos de função não ajustados das funções

identificadas nas medições em APF. As características dos projetos e a análise das medições também são definidas em cada uma dessas seções.

### 3.3.1. Sistema de Gestão de Aquisições

O esforço gasto para desenvolver o sistema de gestão de aquisições, considerando todas as disciplinas de desenvolvimento de software, foi de 3466 horas, sendo registrado uma produtividade de 10,8 horas por ponto de função para o tamanho de 321 PFs na medição pela técnica APF, e 10,6 horas por CFP para o tamanho medido de 326 CFPs em COSMIC, para o desenvolvimento do sistema na linguagem de programação Java. Essa produtividade foi considerada melhor que a média da produtividade dos projetos desenvolvidos na mesma linguagem de programação pela empresa estudada, que tem sua base histórica de medições padronizada em pontos de função. A complexidade para o desenvolvimento desse sistema foi classificada como baixa, considerando a boa produtividade alcançada para uma equipe pouco experiente.

As tabelas 3.1a, 3.1b, 3.1c e 3.1d exibem as medições do sistema de gestão de aquisições nas duas técnicas estudadas, com as análises de cada funcionalidade identificada como processo elementar para a APF e como processo funcional para o COSMIC, de acordo com suas regras para determinação de tamanho.

Processo Elementar / Funcional	APF			COSMIC				
	Tipo de Transação	Complexidade	Tamanho em PF	Entries	Reads	Writes	Exits	Tamanho em CFP
Listar Parâmetros	CE	Baixa	3	1	1		1	3
Alterar Parâmetro	EE	Baixa	3	1	1	1	2	5
Pesquisar Padrão de Documento	CE	Baixa	3	1	2		3	6
Consultar Padrão de Documento	CE	Baixa	3	1	1		1	3
Incluir Padrão de Documento	EE	Baixa	3	1		1	1	3
Alterar Padrão de Documento	EE	Média	4	1	3	1	3	8
Excluir Padrão de Documento	EE	Média	4	1	2	1	2	6
Exportar Padrão de Documento	CE	Baixa	3	1	1		1	3
Pesquisar Sugestão de Preenchimento	CE	Baixa	3	1	2		3	6
Incluir Sugestão de Preenchimento	EE	Média	4	1	2	1	2	6

Tabela 3.1a – Tabela de comparação entre as medições

Processo Elementar / Funcional	APF			COSMIC				
	Tipo de Transação	Complexidade	Tamanho em PF	Entries	Reads	Writes	Exits	Tamanho em CFP
Consultar Sugestão de Preenchimento	CE	Média	4	1	2		1	4
Alterar Sugestão de Preenchimento	EE	Média	4	1	2	1	2	6
Excluir Sugestão de Preenchimento	EE	Média	4	1	2	1	2	6
Pesquisar Documento / Link de Apoio	CE	Baixa	3	1	3		4	8
Incluir Documento / Link de Apoio	EE	Baixa	3	1	2	1	3	7
Alterar Documento / Link de Apoio	EE	Baixa	3	1	3	1	4	9
Consultar Documento / Link de Apoio	CE	Baixa	3	1	1		1	3
Excluir Documento / Link de Apoio	EE	Baixa	3	1	1	1	2	5
Incluir Comentário em Documento / Link de Apoio	EE	Baixa	3	1		1	1	3
Excluir Comentário em Documento / Link de Apoio	EE	Baixa	3	1		1	1	3
Pesquisar Termo de Referência	CE	Baixa	3	1	2		3	6
Incluir Termo de Referência	EE	Alta	6	1	2	1	2	6
Finalizar Termo de Referência	EE	Média	4	1		1	1	3
Criar Versão do Termo de Referência	EE	Média	4	1		1	1	3
Imprimir Termo de Referência	CE	Média	4	1	2		1	4
Alterar Termo de Referência	EE	Alta	6	1	2	1	2	6
Consultar Termo de Referência	CE	Baixa	3	1	1		1	3
Excluir Termo de Referência	EE	Média	4	1	1	1	2	5
Encaminhar Termo de Referência	EE	Alta	6	1	1	1	1	4
Cancelar Encaminhamento Termo de Referência	EE	Alta	6	1	1	1	1	4
Assinar Termo de Referência	EE	Média	4	1		1	1	3
Reabrir Termo de Referência	EE	Média	4	1		1	1	3
Cancelar Termo de Referência	EE	Baixa	3	1		1	1	3
Publicar Termo de Referência	EE	Média	4	1		1	1	3
Controlar Versão - Pesquisar	CE	Baixa	3	1	1		2	4
Consultar Versão	NA	NA	NA	1	1		1	3
Comparar Versão	CE	Média	4	1	1		2	4
Pesquisar Adesão a Ata	CE	Baixa	3	1	2		3	6
Incluir Adesão a Ata	EE	Alta	6	1	2	1	2	6
Finalizar Adesão a Ata	EE	Média	4	1		1	1	3
Criar Versão de Adesão a Ata	EE	Média	4	1		1	1	3
Imprimir Adesão a Ata	CE	Média	4	1	2		1	4
Alterar Adesão a Ata	EE	Alta	6	1	2	1	2	6

Tabela 3.1b – Tabela de comparação entre as medições

Processo Elementar / Funcional	APF			COSMIC				
	Tipo de Transação	Complexidade	Tamanho em PF	Entries	Reads	Writes	Exits	Tamanho em CFP
Consultar Adesão a Ata	CE	Baixa	3	1	1		1	3
Excluir Adesão a Ata	EE	Média	4	1	1	1	2	5
Encaminhar Adesão a Ata	EE	Alta	6	1	1	1	1	4
Cancelar Encaminhamento Adesão a Ata	EE	Alta	6	1	1	1	1	4
Assinar Adesão a Ata	EE	Média	4	1		1	1	3
Reabrir Adesão a Ata	EE	Média	4	1		1	1	3
Cancelar Adesão a Ata	EE	Baixa	3	1		1	1	3
Publicar Adesão a Ata	EE	Média	4	1		1	1	3
Pesquisar Consultoria	CE	Média	4	1	3		2	6
Registrar Consultoria	EE	Alta	6	1	1	1	1	4
Consultar Consultoria	CE	Alta	6	1	2		1	4
Excluir Solicitação de Consultoria	EE	Alta	6	1	1	1	1	4
Gerar Relatório de Desempenho	SE	Média	5	1	3		2	6
Consultar Órgão ( Drop-down )	CE	Baixa	3	NA	NA	NA	NA	NA
Receber Parecer de Análise do Documento	EE	Baixa	3	1		1	1	3
Disponibilizar Dados dos Documentos Cadastrados	CE	Média	4	1	2		1	4
Selecionar Material / Serviço	CE	Baixa	3	1	4		4	9
Consultar Grupo (Drop-down)	CE	Baixa	3	NA	NA	NA	NA	NA
Consultar Classe (Drop-down)	CE	Baixa	3	NA	NA	NA	NA	NA
Selecionar Item de Material / Serviço	CE	Baixa	3	1	1		1	3
Exportar TR Assinado	NA	NA	NA	1	2		1	4
Exportar Adesão Assinada	NA	NA	NA	1	2		1	4
Login (Configurar Perfil do Usuário)	SE	Baixa	4	1	1		1	3
Finalizar Consultoria	EE	Alta	6	1		2	1	4
Restaurar Versão	EE	Média	4	1		2	1	4
Pesquisar Campos do Padrão (Dado de Código)	NA	NA	NA	1	1		2	4
Incluir Campos do Padrão (Dado de Código)	NA	NA	NA	1		1	1	3
Consultar Campos do Padrão (Dado de Código)	NA	NA	NA	1	1		1	3
Alterar Campos do Padrão (Dado de Código)	NA	NA	NA	1	1	1	2	5
Excluir Campos do Padrão (Dado de Código)	NA	NA	NA	1	1	1	1	4
Pesquisar Tipo de Documento (Dado de Código)	NA	NA	NA	1	1		2	4

Tabela 3.1c – Tabela de comparação entre as medições

Processo Elementar / Funcional	APF			COSMIC				
	Tipo de Transação	Complexidade	Tamanho em PF	Entries	Reads	Writes	Exits	Tamanho em CFP
Incluir Tipo de Documento (Dado de Código)	NA	NA	NA	1		1	1	3
Consultar Tipo de Documento (Dado de Código)	NA	NA	NA	1	1		1	3
Alterar Tipo de Documento (Dado de Código)	NA	NA	NA	1	1	1	2	5
Excluir Tipo de Documento (Dado de Código)	NA	NA	NA	1	1	1	1	4
Total das Funções de Dados (ALLs e AIEs)			63	NA				NA
<b>Total PF</b>			<b>321</b>	<b>Total CFP</b>				<b>326</b>

**Tabela 3.1d – Tabela de comparação entre as medições**

Apesar das transações medidas em COSMIC poderem chegar em um tamanho bem maior que a APF segundo suas regras, para esse sistema ficou evidenciado uma aproximação do tamanho das transações identificadas como processo entre as duas técnicas, percebendo em sua maioria uma variação de tamanho entre três e seis CFPs na medição COSMIC, ficando bem próximo do intervalo definido para as funções de transação pela técnica do IFPUG entre três e sete. Ainda assim, a superioridade de tamanho da maioria das transações do COSMIC é refletida no tamanho final, onde mesmo a APF considerando em sua técnica as funções de dados, que nesse caso resultou em 63 PF, o tamanho total em COSMIC ainda foi superior.

Outro ponto a ser destacado na comparação entre as técnicas é que existem transações que são consideradas como processo em uma e não são medidas na outra. Foram identificadas dez transações a mais pelo COSMIC, conforme a seguinte análise:

- Três transações (Consultar Versão; Exportar TR Assinado; Exportar Adesão Assinadas) medidas como processo funcional pela técnica COSMIC, não foram classificadas como processos elementares pela APF por serem consideradas com os mesmos tipos de dados, arquivos referenciados e lógicas de processamento de outras transações, assim não quebrando a unicidade entre elas.

- Dez transações identificadas como dado de código, não sendo classificadas como processo elementar segundo as regras da APF, foram mensuradas como processo funcional pelo COSMIC.
- Três transações do tipo drop-down foram identificadas na medição pela técnica de APF como processos elementares, e rejeitadas como processo funcional pela técnica do COSMIC por serem consideradas como movimentações de READ e EXIT da transação que está contido.

### **3.3.2. Sistema de Controle de Viagens**

O desenvolvimento do sistema de controle de viagens foi finalizado com um total de 3084 horas registradas de esforço da equipe, para todas as fases do projeto. Com esse esforço a produtividade encontrada foi de 9,7 horas por PF, considerando o tamanho do sistema medido na técnica APF de 318 PFs. A produtividade encontrada a partir do tamanho medido em COSMIC foi de 12,2 horas por CFP, considerando a divisão do esforço realizado em horas pelo tamanho encontrado de 252 CFPs. O sistema também foi desenvolvido na linguagem de programação Java, definida em contrato com o mesmo cliente do sistema de gestão de aquisições. A complexidade para o desenvolvimento desse sistema também foi classificada como baixa, comparando sua produtividade com a média da organização estudada de 12 horas por ponto de função desenvolvido. Além da baixa complexidade da maioria dos requisitos e do nível de maturidade heterogêneo da equipe, outro fator que justifica uma produtividade melhor que a do sistema de gestão de aquisições foi que alguns desenvolvedores da equipe do sistema de gestão de aquisições fizeram parte da equipe desse projeto, trazendo suas experiências da utilização de um framework proprietário do cliente, nunca utilizado na empresa antes do sistema de gestão de aquisições.

As Tabelas 3.2a e 3.2b, exibem as medições do sistema de controle de viagens nas duas técnicas estudadas, com as análises de cada funcionalidade identificada como processo elementar para a APF e como processo funcional para o COSMIC, de acordo com suas regras para determinação de tamanho.

Processo Elementar / Funcional	APF			COSMIC				
	Tipo de Transação	Complexidade	Tamanho em PF	Entries	Reads	Writes	Exits	Tamanho em CFP
Incluir Classe de Função	EE	Média	4	1	1	1	1	4
Pesquisar Classe de Função	CE	Baixa	3	1	2		2	5
Detalhar Classe de Função	CE	Baixa	3	1	2		1	4
Alterar Classe de Função	EE	Média	4	1	2	1	2	6
Ativar Classe de Função	EE	Média	4	1	1	1	1	4
Excluir Classe de Função	EE	Baixa	3	1	1	1	2	5
Incluir Classe de Destino	EE	Baixa	3	1		1	1	3
Pesquisar Classe de Destino	CE	Baixa	3	1	1		2	4
Detalhar Classe de Destino	SE	Baixa	4	1	1		1	3
Alterar Classe de Destino	EE	Baixa	3	1	1	1	2	5
Ativar Classe de Destino	EE	Baixa	3	1		1	1	3
Excluir Classe de Função	EE	Baixa	3	1	1	1	2	5
Incluir Diária Nacional	EE	Baixa	3	1	1	1	2	5
Pesquisar Diária Nacional	SE	Baixa	4	1	2		3	6
Detalhar Diária Nacional	SE	Baixa	4	1	1		1	3
Alterar Diária Nacional	EE	Baixa	3	1	2	1	3	7
Listar Parâmetros do Sistema	CE	Baixa	3	1	1		1	3
Alterar Parâmetro do Sistema	EE	Baixa	3	1	1	1	2	5
Pesquisar Setores	CE	Baixa	3	1	1		1	3
Incluir Vinculo do Usuário	EE	Baixa	3	1	1	1	2	5
Pesquisar Vinculo do Usuário	CE	Baixa	3	1	2		2	5
Alterar Vinculo do Usuário	EE	Baixa	3	1	2	1	3	7
Pesquisar Viagem	SE	Baixa	4	1	1		2	4
Incluir Viagem	EE	Alta	6	1	2	1	2	6
Alterar Viagem pelo Solicitante	EE	Alta	6	1	3	1	3	8
Consultar Viagem	SE	Média	5	1	2		1	4
Incluir Viajante	EE	Alta	6	1	3	1	3	8
Alterar Viajante	EE	Alta	6	1	4	1	4	10
Consultar Viajante	SE	Alta	7	1	4		1	6
Listar Viajante	CE	Média	4	1	2		1	4
Excluir Viajante	EE	Alta	6	1	4	1	2	8
Replicar Viagem	EE	Alta	6	1	1	1	1	4
Confirmar Viagem	EE	Alta	6	1	1	1	1	4
Excluir Viagem	EE	Alta	6	1	2	1	2	6
Pesquisar Viagem para Acompanhamento	CE	Baixa	3	1	1		2	4
Consultar para Acompanhar Viagem	CE	Baixa	3	NA	NA	NA	NA	NA
Pesquisar Viagem para Analisar	CE	Baixa	3	1	1		2	4
Analisar Viagem	CE	Alta	6	1	1	1	1	4

Tabela 3.2a – Tabela de comparação entre as medições

Processo Elementar / Funcional	APF			COSMIC				Tamanho em CFP
	Tipo de Transação	Complexidade	Tamanho em PF	Entries	Reads	Writes	Exits	
Login (Configurar Perfil do Usuário)	SE	Baixa	4	1	1		1	3
Listar Prestações de Conta Pendente do Viajante (Minhas Viagens)	CE	Média	4	1	2		2	5
Listar Prestações de Conta Pendente dos Convidados (Convidados)	CE	Média	4	1	2		2	5
Prestar Contas da Viagem pelo Servidor ou Terceiro	EE	Média	4	1	1	1	1	4
Prestar Contas da Viagem pelo Convidado	EE	Média	4	1	1	1	1	4
Analisar Prestações de Contas - Listagem	CE	Média	4	1	2		1	4
Analisar Prestações de Contas - Registrar Analise	EE	Alta	6	1	2	1	1	5
Listagem para Emitir Solicitação de Viagem	SE	Baixa	4	1	1		2	4
Gerar PDF para Emitir Solicitação de Viagem	SE	Média	5	1	1		1	3
Listagem para Prestação de Contas do Viajante	SE	Baixa	4	1	1		2	4
Gerar PDF para Prestação de Contas do Viajante	SE	Média	5	1	1		1	3
Gerar Relatório Cadastro de Viagem	SE	Média	5	1	2		2	5
Listar Viajantes para Relatórios	CE	Média	4	1	2		2	5
Gerar Relatório de Cadastro de Viajantes	SE	Alta	7	1	2		1	4
Gerar Relatório de Demonstrativo de Concessão de Diárias	SE	Alta	7	1	2		2	5
Gerar Relatório de Resumo de Concessão de Diárias	SE	Média	5	1	2		2	5
Consulta Funções (Drop-down)	CE	Baixa	3	NA	NA	NA	NA	NA
Consulta Setores (Drop-down)	CE	Baixa	3	NA	NA	NA	NA	NA
Consulta Usuários (Drop-down)	CE	Baixa	3	NA	NA	NA	NA	NA
Consulta de Setor Gestor (Drop- down)	CE	Baixa	3	NA	NA	NA	NA	NA
Consulta Classe de Função (Drop- down)	CE	Baixa	3	NA	NA	NA	NA	NA
Consulta Classe de Destino (Drop- down)	CE	Baixa	3	NA	NA	NA	NA	NA
Total das Funções de Dados (ALIs e AIEs)			69	NA				NA
<b>Total de PF</b>			<b>318</b>	<b>Total CFP</b>				<b>252</b>

**Tabela 3.2b – Tabela de comparação entre as medições**

Na medição realizada nesse sistema utilizando a abordagem da APF, foram identificadas 32 transações de complexidade baixa, 15 médias e 13 altas. Quando medido a partir das regras do COSMIC, é percebido na maioria das funcionalidades uma aproximação da quantidade de movimentos identificados, com o tamanho da transação já medida pela APF. Esses casos ficam mais evidentes quando a complexidade das transações medidas pela APF são classificadas como baixa ou média. As quatro transações medidas no COSMIC com um tamanho maior que o limite das medidas pela APF que é de 7 PFs, foram classificadas de complexidade alta pela análise de pontos de função.

Como conclusão dessa comparação foi percebido que apesar das transações serem medidas com um tamanho aproximado para a maioria dos processos identificados nas duas técnicas, a medição em APF teve seu tamanho total maior que a medição em COSMIC, como consequência da medição das funções de dados mensuradas apenas pela técnica do IFPUG, pelas 6 consultas do tipo drop-down e pela consulta implícita da funcionalidade ‘Acompanhar Viagem’ que não são conceituadas como processo funcional pelo COSMIC, e que juntas contabilizam 90 PFs. A diferença do tamanho total extraído pela APF subtraído desses 90 PFs resulta em um total menor que o tamanho encontrado a partir do COSMIC, comprovando que na média das demais funcionalidades a quantidade de movimentações encontradas pelo COSMIC foi maior que a contribuição dessas mesmas funcionalidades medidas pela APF.

A consulta implícita da funcionalidade ‘Acompanhar Viagem’ que foi identificada como processo elementar, segundo a técnica de APF, não foi classificada como processo funcional pela medição em COSMIC, por não existir evento disparador para essa função, contrariando a regra de identificação de processos funcionais. As consultas do tipo drop-down também não cumprem as regras de processos funcionais e foram consideradas como movimentações dos processos funcionais em que o atributo faz parte.

### **3.3.3. Sistema de Gerenciamento de Limites Elétricos (GERLIM)**

Foram utilizadas pela equipe do projeto o total de 6929 horas para o desenvolvimento do sistema de gerenciamento de limites elétricos, derivando em uma produtividade de 19,4 horas por pontos de função, a partir do tamanho encontrado de 358 PFs, e uma produtividade para a medição em COSMIC de 21,9 horas por CFP para

um tamanho de 317 CFPs. O sistema foi desenvolvido na linguagem de programação dotNet, onde a base histórica da organização estudada aponta para uma produtividade pior que a do JAVA em 2,1 horas por pontos de função, devido a maior experiência da organização com a última linguagem. A produtividade registrada para esse sistema foi pior em 5,3 horas por PF da média encontrada na base histórica da organização para sistemas desenvolvidos na linguagem dotNet. Esse resultado é consequência do nível de detalhamento da documentação exigida pelo cliente, e da grande complexidade de negócio de seus requisitos, já sendo previsto em contrato essa produtividade.

As Tabelas 3.3a, 3.3b e 3.3c exibem as medições do sistema de gerenciamento de limites elétricos nas duas técnicas estudadas, com as análises de cada funcionalidade identificada como processo elementar para a APF e como processo funcional para o COSMIC, de acordo com suas regras para determinação de tamanho.

Processo Elementar / Funcional	APF			COSMIC				
	Tipo de Transação	Complexidade	Tamanho em PF	Entries	Reads	Writes	Exits	Tamanho em CFP
Pesquisar Minutas Distribuídas - Normatização	SE	Média	5	1	3		4	8
Visualizar Histórico da Minuta Distribuída	CE	Média	4	1	2		1	4
Visualizar Comentários	CE	Baixa	3	1	1		1	3
Responder Comentário	EE	Baixa	3	1		1		2
Alterar Situação do Comentário	EE	Baixa	3	1		1		2
Pesquisar Minutas Distribuídas – Agentes	SE	Média	5	1	2		1	4
Consultar Mensagem de Distribuição	CE	Baixa	3	1	1		1	3
Incluir Comentário	EE	Baixa	3	1		1		2
Pesquisar Evento de Notificação	CE	Baixa	3	1	1		2	4
Alterar Situação do Evento	EE	Baixa	3	1		1	1	3
Alterar Visibilidade do Evento	EE	Baixa	3	1		1	1	3
Pesquisar Registro Notificação	CE	Alta	6	1	6		5	12
Marcar Situação de Leitura	EE	Baixa	3	1		1	1	3
Pesquisar Preferências de Notificação do Usuário	CE	Média	4	1	2		2	5
Incluir Preferências de Notificação - Documento Normativo	EE	Alta	6	1	4	1	3	9

**Tabela 3.3a – Tabela de comparação entre as medições**

Processo Elementar / Funcional	APF			COSMIC				
	Tipo de Transação	Complexidade	Tamanho em PF	Entries	Reads	Writes	Exits	Tamanho em CFP
Incluir Preferências de Notificação – Equipamento	EE	Alta	6	1	8	1	8	18
Incluir Preferências de Notificação – Estação	EE	Alta	6	1	7	1	7	16
Consultar para Alterar Preferências de Notificação - Documento Normativo	CE	Baixa	3	NA	NA	NA	NA	NA
Consultar para Alterar Preferências de Notificação - Equipamento	CE	Baixa	3	NA	NA	NA	NA	NA
Consultar para Alterar Preferências de Notificação - Estação	CE	Baixa	3	NA	NA	NA	NA	NA
Alterar Preferências de Notificação - Documento Normativo	EE	Alta	6	1	6	1	4	12
Alterar Preferências de Notificação – Equipamento	EE	Alta	6	1	11	1	9	22
Alterar Preferências de Notificação – Estação	EE	Alta	6	1	10	1	8	20
Excluir Preferências de Notificação do Usuário	EE	Baixa	3	1	11	1	2	15
Criar Documento Normativo	EE	Baixa	3	1	1	1	2	5
Listar Revisões do Documento Normativo	CE	Baixa	3	1	1		1	3
Pesquisar Documentos Normativos	CE	Média	4	1	3		3	7
Visualizar Documento Normativo	CE	Baixa	3	1	1		1	3
Excluir Minuta	EE	Média	4	1	2	1	1	5
Excluir Programação	EE	Baixa	3	1	1	1	1	4
Cancelar Documento Vigente	EE	Baixa	3	1		1	1	3
Criar Minuta	EE	Alta	6	1	2	1	1	5
Listar Destinatários do SIGOP	CE	Baixa	3	1	1		1	3
Consultar para Alterar Minuta	CE	Alta	6	NA	NA	NA	NA	NA
Alterar Minuta	EE	Alta	6	1	6	1	2	10
Listar Limites Cadastrados do Equipamento	CE	Média	4	1	2		1	4
Pesquisar Equipamentos Cadastrados	CE	Alta	6	1	5		6	12
Consultar para Distribuir Minuta	CE	Média	4	NA	NA	NA	NA	NA
Distribuir Minuta	EE	Média	4	1	3	2	2	8
Consultar para Redistribuir Minuta	CE	Média	4	NA	NA	NA	NA	NA
Redistribuir Minuta	CE	Média	4	1	4	2	2	9
Consultar para Programar Documento	CE	Média	4	NA	NA	NA	NA	NA

Tabela 3.3b – Tabela de comparação entre as medições

Processo Elementar / Funcional	APF			COSMIC				
	Tipo de Transação	Complexidade	Tamanho em PF	Entries	Reads	Writes	Exits	Tamanho em CFP
Distribuir e Vigenciar Documento	EE	Média	4	1	3	2	2	8
Programar Vigência do Documento	EE	Média	4	1	3	2	2	8
Consultar para Redistribuir Programação	CE	Média	4	NA	NA	NA	NA	NA
Redistribuir Programação	EE	Média	4	1	4	2	2	9
Cancelar Programação do Documento	EE	Baixa	3	1	1	2	1	5
Modificar Modelo - Download	CE	Baixa	3	1	1		1	3
Modificar Modelo - Upload	EE	Baixa	3	1		1	1	3
Gerar Documento	EE	Baixa	3	1		1	1	3
Pesquisar Limtes de Equipamento	CE	Média	4	1	7		7	15
Incluir Limites de Equipamento	EE	Média	4	1		1	1	3
Consultar Limites de Equipamento	CE	Média	4	1	1		1	3
Alterar Limites de Equipamento	EE	Média	4	1		1	1	3
Excluir Limites de Equipamento	EE	Baixa	3	1		1	1	3
Consultar Centro Responsável (Drop-down)	CE	Baixa	3	NA	NA	NA	NA	NA
Consultar Centro de Interesse (Drop-down)	CE	Baixa	3	NA	NA	NA	NA	NA
Consultar Documento - Preferências (Drop-down)	CE	Baixa	3	NA	NA	NA	NA	NA
Consultar Estação - Preferências (Drop-down)	CE	Baixa	3	NA	NA	NA	NA	NA
Consultar Equipamento - Preferências (Drop-down)	CE	Baixa	3	NA	NA	NA	NA	NA
Consultar Área Elétrica (Drop-down)	CE	Baixa	3	NA	NA	NA	NA	NA
Consultar Documento (Drop-down)	CE	Baixa	3	NA	NA	NA	NA	NA
Consultar Sigla Instalação (Drop- down)	CE	Baixa	3	NA	NA	NA	NA	NA
Consultar Nome Curto Instalação (Drop-down)	CE	Baixa	3	NA	NA	NA	NA	NA
Consultar Estação (Drop-down)	CE	Baixa	3	NA	NA	NA	NA	NA
Total das Funções de Dados (ALIs e AIEs)			108	NA				NA
<b>Total de PF</b>			<b>358</b>	<b>Total CFP</b>				<b>317</b>

**Tabela 3.3c – Tabela de comparação entre as medições**

Essa medição foi caracterizada pela grande quantidade de processos elementares contados na técnica APF, e que não foram classificados como processos funcionais pelo COSMIC. Também foi interessante perceber em algumas transações a vantagem do COSMIC de ter como parâmetro para seu dimensionamento a quantidade de

movimentações, fazendo com que o tamanho de uma mesma funcionalidade medida em COSMIC tenha a possibilidade de ser bem maior que o tamanho máximo de uma função de transação contado pela APF, onde sua maior transação é a saída externa complexa com uma contribuição de 7 PFs. O fato da medição em COSMIC nesse sistema ter transações maiores, e em algumas situações com tamanho mais de três vezes maior do que o contado pela técnica de APF, ajudou a não ter uma grande diferença entre as duas contagens, considerando que a medição a partir da APF contou como processo elementar oito transações do tipo consulta implícita e dez consultas do tipo drop-down, que não foram classificadas como processo funcional pelo COSMIC. Ainda como favorável a técnica APF, as funções de dados que não são medidas pelo COSMIC, foram medidas em 108 PFs, correspondente a aproximadamente 30% do tamanho total do sistema nessa unidade de medida.

#### **3.3.4. Sistema de Gerenciamento de Informações do Programa Mensal de Operação do Setor Elétrico (SGI-PMO)**

Este sistema teve um esforço reportado pela equipe de desenvolvimento de 5989 horas, resultando em uma produtividade de 15,2 horas por PF para a medição em análise de pontos de função totalizada em 395 PFs, e de 17,0 horas por CFP para a medição realizada a partir da técnica do COSMIC com 352 CFPs.

A reutilização de alguns componentes já desenvolvidos pela equipe do sistema de gerenciamento de limites elétricos fez com que seu desenvolvimento se tornasse mais produtivo que o primeiro desenvolvido em dotNet para o mesmo cliente. A composição de uma equipe com alguns dos profissionais mais experientes da empresa também contribuiu para uma boa produtividade, não ficando tão diferenciada da média da organização, e suprimindo dificuldades do projeto como complexidade de negócio dos seus requisitos e uma quantidade excessiva de documentação exigida em contrato pelo cliente, deixando o processo bem mais “pesado” do que o padronizado pela organização estudada.

As Tabelas 3.4a, 3.4b e 3.4c exibe as medições do sistema de gerenciamento de informações do programa mensal de operação do setor elétrico nas duas técnicas estudadas, com as análises de cada funcionalidade identificada como processo elementar para a APF e como processo funcional para o COSMIC, de acordo com suas regras para determinação de tamanho.

Processo Elementar / Funcional	APF			COSMIC				
	Tipo de Transação	Complexidade	Tamanho em PF	Entries	Reads	Writes	Exits	Tamanho em CFP
Pesquisar PMO	CE	Baixa	3	1	1		1	3
Incluir PMO	EE	Média	4	1		2	1	4
Alterar PMO	EE	Média	4	1	1	2	2	6
Excluir PMO	EE	Média	4	1	2	2	1	6
Abrir Estudo	EE	Alta	6	1	2	2	2	7
Resetar Gabarito	EE	Alta	6	1	1	3	2	7
Pesquisar Gabarito	CE	Alta	6	1	6		4	11
Incluir Gabarito de Reservatório	EE	Alta	6	1	3	2	4	10
Editar Gabarito de Reservatório - consulta implícita	CE	Média	4	NA	NA	NA	NA	NA
Editar Gabarito de Reservatório	EE	Alta	6	1	4	2	3	10
Incluir Gabarito de Usina	EE	Alta	6	1	3	2	4	10
Editar Gabarito de Usina - consulta implícita	CE	Média	4	NA	NA	NA	NA	NA
Editar Gabarito de Usina	EE	Alta	6	1	4	2	3	10
Incluir Gabarito de Unidade Geradora	EE	Alta	6	1	4	2	5	12
Editar Gabarito de Unidade Geradora- consulta implícita	CE	Média	4	NA	NA	NA	NA	NA
Editar Gabarito de Unidade Geradora	EE	Alta	6	1	4	2	4	11
Incluir Gabarito de Subsistema	EE	Alta	6	1	3	2	4	10
Editar Gabarito de Subsistema - consulta implícita	CE	Baixa	3	NA	NA	NA	NA	NA
Editar Gabarito de Subsistema	EE	Alta	6	1	3	2	3	9
Listar Agentes Participantes de Gabarito	CE	Baixa	3	1	2		1	4
Listar Reservatórios Participantes de Gabarito	CE	Baixa	3	1	2		1	4
Listar Usinas e Unidades Geradoras Participantes de Gabarito	CE	Baixa	3	1	3		1	5
Pesquisar Configuração de Coletas de Dados do Estudo	CE	Alta	6	1	4		4	9
Abrir Coleta do Estudo	EE	Alta	6	1	2	2	1	6
Fechar Coleta do Estudo	EE	Média	4	1		2	1	4
Capturar Coleta do Estudo	EE	Média	4	1		2	1	4
Aprovar Coleta do Estudo	EE	Média	4	1		2	1	4
Rejeitar Coleta do Estudo	EE	Alta	6	1	1	2	1	5
Visualizar Coletas Estruturadas do Estudo	CE	Alta	6	1	8		1	10

Tabela 3.4a – Tabela de comparação entre as medições

Processo Elementar / Funcional	APF			COSMIC				
	Tipo de Transação	Complexidade	Tamanho em PF	Entries	Reads	Writes	Exits	Tamanho em CFP
Consultar Configuração para Incluir Dado de Coleta do Insumo Estruturado	CE	Alta	6	NA	NA	NA	NA	NA
Incluir Dado de Coleta do Insumo Estruturado	EE	Alta	6	1	2	2	1	6
Alterar Dado de Coleta do Insumo Estruturado	EE	Alta	6	1	2	2	2	7
Recuperar Dados da Coleta Anterior do Insumo Estruturado	CE	Alta	6	1	8		1	10
Enviar e Salvar Dado de Coleta do Insumo Estruturado	EE	Alta	6	1	2	2	1	6
Enviar Coletas do Estudo	EE	Alta	6	1	1	2	1	5
Visualizar Coletas de Manutenção do Estudo	CE	Alta	6	1	7		1	9
Incluir Coleta de Manutenção	EE	Média	4	1	2	2	3	8
Alterar Coleta de Manutenção	EE	Média	4	1	1	2	2	6
Excluir Coleta de Manutenção	EE	Baixa	3	1		2	1	4
Listar Manutenções do SGI	CE	Alta	6	1	4		1	6
Visualizar Coleta Não Estruturadas do Estudo	CE	Alta	6	1	5		1	7
Incluir Coleta Não Estruturada do Estudo	EE	Média	4	1		2	1	4
Alterar Coleta Não Estruturada do Estudo	EE	Média	4	1	1	2	2	6
Realizar Download de Arquivo Coleta Não Estruturada	CE	Baixa	3	1	1		1	3
Listar Arquivos dos Insumos Não Estruturados - Geração de Blocos/CCEE/Publicação Resultados	CE	Média	4	1	3		1	5
Gerar Blocos do Montador	SE	Alta	7	1	7	1	1	10
Realizar Download de Arquivo do Estudo	CE	Baixa	3	1	1		1	3
Excluir Arquivo Blocos Montador	EE	Média	4	1		2	1	4
Convergir Arquivo com CCEE	EE	Alta	6	1	1	2	1	5
Publicar Resultado	EE	Média	4	1		2	1	4
Reprocessar PMO	EE	Média	4	1		2	1	4
Pesquisar Insumos	CE	Baixa	3	1	1		1	3
Incluir Insumo Estruturado	EE	Alta	6	1		2	1	4
Consultar para Alterar Insumo Estruturado	CE	Baixa	3	NA	NA	NA	NA	NA
Alterar Insumo Estruturado	EE	Média	4	1	1	2	2	6
Excluir Insumo Estruturado	EE	Baixa	3	1		2	1	4

Tabela 3.4b – Tabela de comparação entre as medições

Processo Elementar / Funcional	APF			COSMIC				
	Tipo de Transação	Complexidade	Tamanho em PF	Entries	Reads	Writes	Exits	Tamanho em CFP
Incluir Insumo Não Estruturado	EE	Média	4	1		2	1	4
Consultar para Alterar Insumo Não Estruturado	CE	Baixa	3	NA	NA	NA	NA	NA
Alterar Insumo Não Estruturado	EE	Média	4	1	1	2	2	6
Excluir Insumo Não Estruturado	EE	Baixa	3	1		2	1	4
Pesquisar Log Auditoria	CE	Baixa	3	1	1		1	3
Detalhar Log Auditoria	CE	Baixa	3	1	1		1	3
Visualizar Arquivos para Convergência PLD	CE	Média	4	1	2		1	4
Convergir PLD de Estudo	EE	Média	4	1		2	1	4
Não Convergir PLD de Estudo	EE	Média	4	1		2	1	4
Listar Estudos Cadastrados ( Drop-down )	CE	Baixa	3	NA	NA	NA	NA	NA
Listar Agentes Cadastrados ( Drop-down )	CE	Baixa	3	NA	NA	NA	NA	NA
Listar Insumos cadastrados ( Drop-down )	CE	Baixa	3	NA	NA	NA	NA	NA
Listar Reservatórios ( Drop-down )	CE	Baixa	3	NA	NA	NA	NA	NA
Listar Usinas ( Drop-down )	CE	Baixa	3	NA	NA	NA	NA	NA
Listar Unidades Geradoras ( Drop-down )	CE	Baixa	3	NA	NA	NA	NA	NA
Listar Insumos por categoria (reservatório, usina, unidade geradora) ( Drop-down )	CE	Baixa	3	NA	NA	NA	NA	NA
Total das Funções de Dados (ALIs e AIEs)			74	NA				NA
<b>Total de PF</b>			<b>395</b>	<b>Total CFP</b>				<b>352</b>

**Tabela 3.4c – Tabela de comparação entre as medições**

Da comparação entre as medições realizadas no sistema de gerenciamento de informações do programa mensal de operação do setor elétrico, temos como relevante dessa análise a evidência de 14 transações do tipo consulta implícita e consulta drop-down, consideradas como processos elementares na análise de pontos de função, e que não foram classificadas como processo funcional aplicando as regras do COSMIC. Foi identificado também nessa comparação que das 72 transações consideradas como processo elementar na medição pela APF, apenas quatro tiveram seu tamanho medido com um valor maior do que quando mensurado pela técnica COSMIC, constatando a possibilidade da técnica COSMIC ser mais justa quando comparadas as estimativas de

esforço entre duas transações classificadas com a mesma complexidade pela técnica de APF.

Ficou constatado que nas medições utilizando a técnica COSMIC não existe possibilidades de tamanhos pré-definidos para as transações, e cada uma dessas transações tem seu tamanho definido unicamente a partir de suas movimentações. Essa vantagem fica mais clara quando precisamos estimar com um mesmo prazo o desenvolvimento de duas transações consideradas como complexas segundo regras da APF, e verificamos na conclusão do desenvolvimento que a duração de uma delas foi o dobro de horas da outra. Percebe-se que estimar esforço por movimentação encontrada em uma transação, utilizando a técnica COSMIC, é mais assertivo que julgar o tempo para seu desenvolvimento pela classificação de seu tipo e complexidade como na APF.

### 3.4. Formação da Fórmula de Conversão

Foi encontrado um fator de conversão entre as técnicas APF e COSMIC definido a partir da média aritmética dos valores encontrados na divisão do resultado da medição em COSMIC pelo resultado da medição em APF, para os quatro projetos estudados, de acordo com a Tabela 3.5. Esse fator é utilizado como um dos parâmetro da fórmula de conversão definida como resultado do estudo de caso dessa pesquisa.

<b>Sistema</b>	<b>APF</b>	<b>CFP</b>	<b>Fator de Conversão</b>
Aquisições	321	326	1,02
Viagens	318	252	0,79
GERLIM	358	317	0,89
SGI-PMO	395	352	0,89
<b>TOTAL</b>	1392	1247	<b>0,90</b>

**Tabela 3.5 – Tabela de formação do fator de conversão**

Este fator de conversão deve ser considerado como indicativo, precisando ser fortalecido em estudos posteriores, a partir de experimentos em uma quantidade maior de projetos. Estudos de caso com medições em projetos de melhoria e projetos de tamanhos maiores, serão necessários para sua consolidação.

Como resultado do estudo de caso foi definida a fórmula  $Y(\text{CFP}) = 0,90 * (\text{UFP})$ , onde ‘Y(CFP)’ é o resultado da conversão na unidade de medida COSMIC Function Point, 0,90 é o fator de conversão encontrado nessa pesquisa, e ‘UFP’ é o tamanho do sistema a ser convertido, medido em pontos de função não ajustados conforme a técnica de análise de pontos de função.

### **3.5. Considerações Finais do Capítulo**

Este capítulo apresentou a empresa onde foi realizado o estudo de caso, as informações dos projetos de software medidos para um melhor entendimento do estudo, as medições desses projetos nas técnicas APF e COSMIC, e uma análise comparativa das características identificadas nessas medições, possibilitando através da prática a percepção de algumas particularidades de cada técnica. A pesquisa realizada nesse capítulo teve como resultado a fórmula  $Y(\text{CFP}) = 0,90 * (\text{UFP})$ , definida a partir do fator de conversão encontrado.

# 4 Conclusão

---

Nesse capítulo serão discutidas as considerações finais sobre este trabalho, suas limitações, ameaças à validade, e possibilidades de trabalhos futuros.

## 4.1. Considerações Finais

Apesar do conteúdo abordado no capítulo de referencial teórico ser suficiente para perceber algumas das diferenças entre os conceitos e regras das duas técnicas estudadas, os oito projetos de medição executados nos quatro sistemas escolhidos para o estudo de caso, além de terem sido os insumos para a fórmula de conversão encontrada, permitiram a análise de algumas características do processo de medição identificadas entre a APF e o COSMIC.

Uma das principais atividades dentro do processo de medição das duas técnicas estudadas é classificar as transações em processos elementares na APF, e em processos funcionais para o COSMIC. A partir das medições executadas nos quatro projetos de desenvolvimento do estudo de caso foi verificado que o conceito para definir um processo elementar na APF é equivalente as regras para identificar um processo funcional no COSMIC, observando que em apenas algumas poucas exceções, relatadas durante a análise de cada projeto do capítulo 3, não existiu essa equivalência, como as transações consideradas como dado de código que não são consideradas no escopo da APF e que devem ser medidas como processo funcional no COSMIC, e as transações do tipo consulta drop-down e consulta implícita que são contadas como processo elementar na APF e não são medidas pelo COSMIC.

Esse trabalho considera como característica diferenciada da técnica COSMIC, dentro dos conceitos de medição funcional, a abordagem de que o tamanho de uma transação é medido a partir da quantidade de suas movimentações de dados, não existindo como na APF possíveis tamanhos pré-definidos de acordo com os parâmetros classificadores das complexidades baixa, média e alta. Conforme as regras da APF uma transação pode ter no máximo sete pontos de função, independentemente se ela referenciar quatro ou cinquenta grupos de dados na execução de suas lógicas de processamento. Nas regras do COSMIC o tamanho de uma transação é impactado por

essa quantidade de grupos de dados referenciados, onde cada uma dessas referências é uma movimentação do tipo *Read*. O COSMIC consegue diferenciar duas transações já consideradas como complexas, porém com uma grande diferença de esforço necessário para suas construções.

Foi identificada uma maior facilidade para o aprendizado da técnica COSMIC, característica defendida por seus criadores. Porém deve ser considerado o conhecimento já adquirido da técnica APF pelo autor desse trabalho, que utiliza alguns conceitos similares, como propósito, escopo, fronteira e usuário, e outros bem próximos como grupo de dados e arquivos lógicos, processo elementar e processo funcional.

As vantagens da APF são a consolidação da técnica no mercado brasileiro com profissionais certificados e experientes em sua utilização, grupos de discussão para apoio a esclarecimento de dúvidas e boas práticas estabelecidas pelo mercado, uma boa quantidade de material para estudo, e fontes para extração dos dados de benchmarking da produtividade do desenvolvimento de software, como o *International Software Benchmarking Standards Group* (ISBSG, 2013). Já nas medições utilizando o COSMIC, para as dúvidas identificadas na pesquisa, houve uma grande dificuldade de encontrar profissionais com experiência na técnica e capacidade de esclarecimentos dessas dúvidas, e ainda uma quantidade limitada de material para estudo que complementa o manual e documentos oficiais.

A comparação dos esforços gastos entre as duas medições de cada projeto, utilizando as técnicas estudadas, foi dificultada pelo processo estabelecido para o estudo de caso desse trabalho, onde a segunda medição tem sua produtividade beneficiada pela ausência da etapa de entendimento dos requisitos de negócio do software, já realizada na primeira medição. Porém, foi verificada uma maior eficiência para analisar e documentar as medições quando utilizada a técnica COSMIC, considerando que nessa técnica apenas as transações são medidas, não necessitando utilizar o modelo de dados como artefato para medir os agrupamentos de dados. Também foi identificada uma maior agilidade para definir o tamanho das funcionalidades no COSMIC a partir da quantidade de suas movimentações, enquanto que pelas regras da APF o tamanho de uma função de transação é determinado por sua complexidade, classificada a partir da quantidade de tipos de dados e arquivos referenciados.

É importante lembrar que a fórmula de conversão encontrada nesse trabalho deve ser utilizada como indicativo, considerando a quantidade de experimentos realizados, e

as características das medições que foram limitadas ao perfil dos projetos escolhidos na pesquisa.

Por fim, a importância do conteúdo abordado nesse trabalho não se restringe ao resultado encontrado da fórmula de conversão e as características identificadas na comparação entre as duas técnicas estudadas, mas também a contribuição para a abertura de uma discussão acadêmica sobre a necessidade da medição para o sucesso dos projetos de software e quais as melhores técnicas existentes para essas medições, assim como a possibilidade de migração entre as unidades de medida de software mais utilizadas nas bases históricas das organizações de forma simples e com menor custo.

## **4.2. Limitações e Ameaças à Validade**

As medições foram realizadas apenas em projetos de desenvolvimento, como consequência da falta de projetos de melhoria disponibilizados pela organização estudada. Estudos de casos em projetos de melhoria seriam interessantes para demonstrar o citado no capítulo de referencial teórico, que nas medições nesse tipo de projeto utilizando regras da APF, a inclusão de um campo como melhoria de uma funcionalidade tem o mesmo tamanho que todo o seu desenvolvimento, o que dificulta a estimativa de esforço com essa unidade de medida. Na abordagem do COSMIC para projetos de melhoria apenas as movimentações impactadas pelas alterações nas funcionalidades modificadas são consideradas na medição, alcançando uma maior precisão do que de fato foi alterado, e facilitando uma comparação do esforço para realizar a melhoria com o de desenvolver toda a funcionalidade.

Apesar do COSMIC ter sido projetado para medir o tamanho funcional de qualquer componente de software multicamadas, seja em software aplicativo de negócio, software de tempo-real e híbridos dos dois tipos, a pesquisa foi realizada apenas na camada de aplicação dos softwares aplicativos de negócio, considerando ser a única camada e tipo de sistema em que a APF é aplicável, possibilitando a comparação, e portanto não classificando a ausência de experimentos nesses outros tipos de softwares e camadas como limitação do trabalho.

Quando esse trabalho estava em sua fase final, foi encontrado um artigo (ABRAN et al., 2005) que aborda o mesmo assunto pesquisado, onde dois dos seus autores fazem parte do grupo criador do COSMIC, e apresentam uma análise e validação sobre três estudos realizados sobre a conversão de projetos medidos em APF para o COSMIC.

Desses três estudos analisados por Abran et al., um realiza experimentos em quatro sistemas menores que 100 PF medidos em projetos de desenvolvimento, outro em onze projetos de tamanhos variados utilizando a técnica da NESMA, e o último em seis projetos de desenvolvimento com tamanho médio entre eles de 400 PF. Os três estudos analisados utilizaram o método estatístico de regressão linear para definição de suas fórmulas de conversão. Apesar de reconhecer a validade dos três estudos analisados, os autores do artigo consideram que os fatores de conversão podem sofrer uma variação de acordo com as particularidades de um conjunto de projetos estudados, porém devem ser muito próximos para grande parte dos projetos. O artigo é finalizado com o posicionamento de que mais pesquisas são necessárias para fortalecer a precisão desses fatores.

A Tabela 4.1 mostra uma análise comparativa entre as conversões dos projetos medidos nesse trabalho utilizando sua fórmula encontrada e a fórmula definida no estudo da Seção 4 do artigo do Abran et al. (ABRAN et al., 2005). Foi utilizado como critério de seleção das fórmulas analisadas por Abran e outros autores, o estudo com dados coletados mais recentemente e utilizando versões dos manuais de medição mais atuais, onde a média de tamanho de seus projetos mais se aproximasse com a média de tamanho dos projetos do estudo de caso desse trabalho, e com medições também em projetos de desenvolvimento. A primeira coluna da tabela é formada pelos nomes dos sistemas medidos, a segunda pelas medições realizadas utilizando a técnica APF com pontos de função não ajustados, a terceira pelas medições utilizando o COSMIC, a quarta é formada pela conversão utilizando a fórmula encontrada nesse trabalho, e na quinta coluna pela conversão utilizando a fórmula selecionada no artigo do Abran et al., onde nas duas fórmulas o 'Y(CFP)' é o resultado da conversão e o 'UFP' é o valor encontrado na segunda coluna.

<b>Sistema</b>	<b>APF (UFP)</b>	<b>CFP</b>	<b>Fórmula desse Trabalho: <math>Y(CFP) = 0,90 * (UFP)</math></b>	<b>Fórmula do Artigo do Abran et al.: <math>Y(CFP) = 0.84 * (UFP) + 18</math></b>
Aquisições	321	326	288,9	287,6
Viagens	318	252	286,2	285,1
GERLIM	358	317	322,2	318,7
SGI-PMO	395	352	355,5	349,8

**Tabela 4.1 – Tabela comparativa dos resultados das fórmulas de conversão**

Como resposta da comparação entre as duas fórmulas foram encontrados resultados bem próximos, com variação menor que 2% entre as conversões de cada projeto, o que fortalece a validade do resultado encontrado nesse trabalho.

### **4.3. Trabalhos Futuros**

Como sugestão de trabalho futuro acredita-se que a fórmula de conversão encontrada precisa ser fortalecida com uma maior quantidade de experimentos, com medições de tamanhos diversos e em projetos de desenvolvimento, aplicação e melhoria, abrangendo particularidades de cada um desses cenários.

Sugere-se também o estudo do método estatístico de regressão linear e a sua aplicação nesses novos experimentos para fortalecimento da fórmula de conversão encontrada e corroborado pelo artigo encontrado na etapa final deste trabalho.

# Referências

- (ABRAN et al., 2005) Abran, A.; Desharnais, J.M.; Azziz, F. **Measurement Convertibility: From Function Points to COSMIC-FFP**, International Workshop on Software Measurement – IWSM, 2005 .
- (AGUIAR, 2003) Aguiar, M. **Pontos de Função ou Pontos por Caso de Uso? Como Estimar Projetos Orientados a Objetos**, 2003. Disponível em: <[http://www.bfpug.com.br/Artigos/UCP/Aguiar-Pontos\\_de\\_Funcao\\_ou\\_Pontos\\_por\\_Caso\\_de\\_Uso.pdf](http://www.bfpug.com.br/Artigos/UCP/Aguiar-Pontos_de_Funcao_ou_Pontos_por_Caso_de_Uso.pdf)>. Acesso em: 31 Outubro 2011.
- (AGUIAR; BAKLIZKY, 2010) Aguiar, M.; Baklizky, D. **Modelos de Negócio Baseados em Pontos de Função**. ISMA 5 - Conferência Internacional de Medição e Análise de Software do IFPUG, São Paulo, 2010.
- (AGUIAR, 2014) Aguiar, M. **IFPUG Director of International Affairs, private communication**, January 2014.
- (ANDRADE, 2004) Andrade, E. L. P., **Pontos de Casos de Uso e Pontos de Função na Gestão de Estimativa de Tamanho de Projetos de Software Orientados a Objeto**. Proposta de Tese de Mestrado defendida em 2004 na Universidade Católica de Brasília, Brasília, Brasil.
- (ANDRADE; OLIVEIRA, 2004) Andrade, E.; Oliveira, K. **Uso Combinado de Análise de Pontos de Função e Casos de Uso na Gestão de Estimativa de Tamanho de Projetos de Software Orientado a Objetos**. Simpósio Brasileiro de Qualidade de Software, 3. Brasília, 2004.
- (ANDRÉ, 2005) André, M. E. D. A. **Estudo de Caso em Pesquisa e avaliação educacional**. Brasília: Liber Livro Editora, 2005.
- (ARAUJO, 2008) Araujo, O., **Definição de uma Metodologia para Auditoria de Preços em Contratos de Projetos de Software baseada na Análise de Pontos de Função**, Monografia (Pós-graduação) – Universidade Estácio de Sá, Rio de Janeiro, 2008.
- (BALDASSARE et al., 2009) Baldassare, M. T.; Boffoli, N.; Bruno, G.; Caivano, D. **Statistically Based Process Monitoring: Lessons from the Trench**, Lecture Notes in Computer Science, v. 5543, 2009.
- (BASILI et al., 1994) Basili V. R., Caldiera, G. e Rombach, H. D. **The Goal Question Metric Paradigm, Encyclopedia of Software Engineering**. New York: John Riley and Sons, 1994.
- (BASSETO, 2008) Basseto, N. **Uma abordagem para um programa de medição baseado em análise de pontos de função**, Instituto Brasileiro de Tecnologia Avançada - IBTA, São Paulo, 2008.

(BFPUG, 2008). Brazilian Function Point Users Group, **Número de CFPS por País**. Disponível em: [www.bfpug.com.br](http://www.bfpug.com.br), Acesso em: 29 Janeiro 2014.

(BORGES, 2003) Borges, E.P. **Um modelo de medição para processos de desenvolvimento de software**, Dissertação (Mestrado) - UFMG, Belo Horizonte, p. 154, 2003.

(CALAZANS, 2003) Calazans, A. T. S. **A utilização do método Cosmic Full Function Points para o cálculo de estimativas de tamanho de software**, Developers Magazine, 2003.

(CALAZANS, 2006) Calazans, A. T. S. **Definição de um modelo para mensurar projetos de manutenção de sistemas**, EQPS - Encontro de qualidade e produtividade de software, 2006.

(CAMPOS, 2010) Campos, C. **Escopo da Contagem**, 2010. Disponível em: <http://carloscamposinfo.com/mundoapf/?p=192>>. Acesso em: 28 Janeiro 2014.

(CAMPOS, 2010b) Campos, C. **Fronteira da aplicação**, 2010. Disponível em: <http://carloscamposinfo.com/cjec/?p=194>>. Acesso em: 29 Janeiro 2014.

(CAMPOS, 2011) Campos, C. **Tipos de Função**, 2011. Disponível em: <http://carloscamposinfo.com/mundoapf/?p=197>>. Acesso em: 03 Fevereiro 2014.

(CAMPOS, 2013) Campos, F. M. **Métricas de Software Como Ferramenta de Apoio ao Gerenciamento de Projetos de Software**. Disponível em: <http://www.linhadecodigo.com.br/artigo/453/metricas-de-software-como-ferramenta-de-apoio-ao-gerenciamento-de-projetos-de-software.aspx>>. Acesso em: 03 Outubro 2013.

(CANFORA et al., 2004) Canfora, G.; Garcia, F.; Piattini, M.; Ruiz, F.; Visaggio, C. A. **A Family of Experiments to Validate Metrics for Software Process Models**, The Journal of Systems and Software, v. 77, n. 2, 2004.

(CANOSSA, 2005) Canossa S. **Gerenciando o Sistema de Medição**. Disponível em: <http://internal.dstm.com.ar/sites/mmnew/bib/notas/Medicao.pdf>>. Acesso em: 22 maio 2005.

(CLARK, 2002) Clark B., **Eight Secrets of Software Measurement**, IEEE Software, Outubro de 2002.

(COSMIC, 2007) Common Software Measurement International Consortium, **O Método COSMIC para Medição de Tamanho Funcional: Visão Geral do Método**, Setembro de 2007.

(COSMIC, 2009) Common Software Measurement International Consortium, **O Método COSMIC para Medição de Tamanho Funcional: Manual de Medição**, Maio de 2009.

(DEKKERS, 2003) Dekkers, C. **Measuring the logical or function a Size of Software Projects and Software Application**. Spotlight Software, ISO Bulletin May 2003.

(DUMKE et al., 2006) Dumke, R. R.; Braungarten, R.; Blazey, M.; Hegewald, H.; Reitz, D.; Drichter, K. **Software Process Measurement and Control - A Measurement based Point of View of Software Processes**, Technical Report, Dept. of Computer Science, University of Magdeburg, Germany, 2006.

(DUMKE; ABRAN, 2011) Dumke, R.; Abran, A., **COSMIC Function Points: Theory and Advanced Practices**, 2011.

(FABIAN, 2008) Fabian, C. M. **SICLA-APF Ferramenta de Análise de Pontos Por Função**, 2008. Disponível em: <<http://www.fattocs.com.br/livro-apf/citacao/CarlosMarceloFabian-2008.pdf>>. Acesso em: 28 Janeiro 2014.

(FATTO, 2012) Fatto CS. **Perguntas e Respostas sobre Análise de Pontos de Função** Disponível em: < <http://fattocs.com/pt/faq-apf.html>>. Acesso em: 05 Fevereiro 2012.

(FEITOSA, 2004) Feitosa, C. M. A. C. **Definição de um Processo de Medição e Análise com base nos Requisitos do CMMI**, Tese de Mestrado em Ciências da Computação - área de concentração Eng. de Software. Universidade Federal de Pernambuco, 2004.

(FERNANDES, 1995) Fernandes, A. A. **Gerência de Software Através de Métricas**. São Paulo: Atlas, 1995.

(FERREIRA; NUNES, 2008) Ferreira, S. B. L.; Nunes, R. R. **e-Usabilidade**. São Paulo: LTC, 2008.

(FLORAC; CARLETON, 1999) Florac, W. A.; Carleton, A. D. **Measuring the Software Process: Statistical Process Control for Software Process Improvement**. Addison-Wesley, MA, 1999.

(FRANÇA et al., 1998) França, L.P.A.; Von Staa, A.; Lucena, C.J.P. **Medição de Software para Pequenas Empresas: uma Solução Baseada na Web**. Rio de Janeiro: PUC-RJ, 1998.

(GARMUS; HERRON, 2001) Garmus, D.; Herron D. **Function Points Analysis – Measurement Practices for Successful Software Projects**, Estados Unidos: Addison Wesley, p. 363, 2001.

(GOLDENSON et al., 2003) Goldenson D. R., Jarzombek J., Rout T., **Measurement and Analysis in Capability Maturity Model Integration Models and Software Process Improvement**, CrossTalk The jornal of Defense of Software Engineering, 2003.

(HAZAN, 2001) Hazan, C. **Análise de Pontos de Função**, 2001. Disponível em: <[www.inf.ufes.br/~falbo/download/aulas/es-g/2005-1/APF.pdf](http://www.inf.ufes.br/~falbo/download/aulas/es-g/2005-1/APF.pdf)>. Acesso em: 02 Fevereiro 2014.

(HAZAN, 2004) Hazan, C.: **Método para Aplicação da Contagem de Pontos de Função como Suporte a Engenharia de Requisitos**. Proposta de Tese de Doutorado defendida em Dezembro/2004 na PUC-Rio, Rio de Janeiro, RJ, Brasil

(HAZAN, 2005) Hazan, C.; Berry, D.M.; Leite, J.S.P. **É possível substituir processos de engenharia de requisitos por contagem de pontos de função?** In: INTERNATIONAL WORKSHOP ON REQUIREMENTS ENGINEERING (WER), 8th, 2005, Porto, Portugal. Anais... (Porto: s.n., 2005?). p. 197-208. Disponível em: <[http://wer.inf.puc-rio.br/WERpapers/artigos/artigos\\_WER05/claudia\\_hazan.pdf](http://wer.inf.puc-rio.br/WERpapers/artigos/artigos_WER05/claudia_hazan.pdf)>.

(HAZAN, 2009) Hazan, C. **Análise de Pontos de Função - Uma Aplicação nas Estimativas de Tamanho de Projetos de Software**. Engenharia de Software Magazine, 2009. ISSN 1983127-7.

(HAZAN, 2010) Hazan, C. **Como evitar armadilhas em contratos de fábricas de software**. Engenharia de Software, Revista do TCU, 2010.

(HETZEL, 1993) Hetzel, B. **Making Software Measurement Work: Building an Effective Measurement Program**. John Wiley and Sons: LTD, 1993.

(IFPUG, 2010). **Function Point Counting Practices. Manual: Release 4.3.1**. International Function Point Users Group, [www.ifpug.org](http://www.ifpug.org), jan. 2010.

(ISBSG, 2013). **Software Development Projects in Government - performance, practices and predictions**. International Software Benchmarking Standards Group. Disponível em: <<http://www.bfpug.com.br/Artigos/ISBSG/Govt%20Sector%20Report%20Rel%201.3%20%201912031.pdf>>. Acesso em 06/09/2013.

(ISBSG, 2014) **International Software Benchmarking Standards Group**. Disponível em: <<http://www.isbsg.org/>>. Acesso em 21/02/2014.

(ISO 9001, 2008) NBR ISO 9001 **Normas de Gestão de Qualidade e Garantia da Qualidade – Diretrizes para aplicação da NBR 2008**.

(JONES, 1994) Jones, C. **Function points**. Computer, v. 27, n. 8, p. 66-67, Aug. 1994.

(JONES, 2001) Jones, C. **Conflict and Litigation Between Software Clients and Developers**, Software Productivity Research, versão 10 – Abril de 2001.

(KAN, 2003) Kan, S. H, **Metrics and Models in Software Quality Engineering**, Addison Wesley, 2003.

(KAUTZ, 1999) Kautz, K., **Making Sense of Measurement for Small Organizations**, IEEE Software.1999.

(KAVANAGH, 2003) **Kavanagh, J., Cosmic-FFP offers an alternative to function point analysis method**. Computer Weekly, Julho 2003. Disponível em: <<http://www.computerweekly.com/feature/Cosmic-FFP-offers-an-alternative-to-function-point-analysis-method>>. Acesso em: 10/01/2014.

(KITCHENHAM *et al.*, 1995) Kitchenham, B.; Pfleeger, S.; Fenton, N. **Towards a Framework for Software Measurement Validation**. IEEE Transactions on Software Engineering. 21 (12), pp. 929- 943, 1995.

(MACORATTI, 2005) Macoratti, J. C. **Análise de Pontos por Função - O Processo de contagem**, 2005. Disponível em: <[http://www.macoratti.net/apf\\_pcta.htm](http://www.macoratti.net/apf_pcta.htm)>. Acesso em: 10/02/2012.

(MARETTO; JUNGER, 2008) Maretto, C.; Junger W. **Ferramenta de Estimativas de Projetos de Software Baseada em Pontos de Função**. Centro Federal de Educação Tecnológica do Espírito Santo. 2008.

(MENS; DEMEYER, 2001) Mens, T.; Demeyer, S. **Future Trends in Software Evolution Metrics**. In: International Workshop on Principles of Software Evolution - IWPSE, 4., Vienna, Austria, 2001.

(NESMA, 2013) **Netherlands Software Metrics Association**. Disponível em : <<http://www.nesma.nl/section/nesma/>> Acesso em : 26/03/2013.

(OLIGNAY *et al.*, 1998) Olignay, S.; Abran, A.; Desharnais, J.-M.; Morris, P. , **Functional Size of Real-Time Software: Overview of Field Tests**, in Proceedings of the 13th International Forum on COCOMO and Software Cost Modeling, Los Angeles, CA, Outubro 1998.

(PERKINS *et al.*, 2003) Perkins T.; Peterson R.; Smith, L. **Back to the Basics: Measurement and Metrics**. USA: Department of Defense and USA Army, 2003. Disponível em: <[www.stsc.af.mil/crosstalk](http://www.stsc.af.mil/crosstalk)> Acesso em: out. 2005.

(PERRY, 1986) Perry, W.E. **The best measures for measuring data processing quality and productivity**. Quality Assurance Institute Technical Report. 1986.

(PFLEEGER, 1993) Pfleeger S., **Lessons Learned in Building a Corporate Metrics Program**, IEEE Software, 1993.

(PFLEEGER *et al.*, 1997) Pfleeger S. L., Jeffery R., Curtis B., Kitchenham B., **Status Report on Software Measurement**, IEEE Software, 1997.

(PRESSMAN, 1995) Pressman R. S. **Engenharia de Software**. São Paulo, Makron Books, 1995.

(ROCHA *et al.*, 2001) Rocha, A; Maldonado, J; Weber, K. **Qualidade de Software Teoria e Prática**. Prentice Hall, 2001.

(RIFKIN; COX, 1991) Rifkin S., Cox C., **Measurement in Practice**, Software Engineering Institute Technical Report CMU/SEI-91-TR-016, 1991.

(RUP, 2013) IBM Rational Unified Process. Disponível em: <[http://pt.wikipedia.org/wiki/IBM\\_Rational\\_Unified\\_Process](http://pt.wikipedia.org/wiki/IBM_Rational_Unified_Process)>, Acesso em: 11 Novembro 2013.

(SANTANA; GUSMÃO, 2010) Santana, C.; Gusmão, C. **Uso de Análise de Pontos de Função em Ambientes Ágeis**. Engenharia de Software Magazine, 2010. ISSN 1983127-7.

(SANTOS; MOREIRA, 2006) Santos, M.; Moreira, A. **Análise de Pontos por Função (APF): Nível de utilização nas empresas que produzem software, localizadas em Salvador/Bahia**. Monografia de conclusão do curso de (Graduação) - Faculdade Helio Rocha, 2006.

(SEI, 2010) *SOFTWARE ENGINEERING INSTITUTE. CMMI for Development (CMMI-DEV), Version 1.3, Technical Report CMU/SEI-2010-TR-033*. Pittsburgh, PA: *Software Engineering Institute, Carnegie Mellon University*, 2010.

(SILVA, 2000) Silva, I. J. de M. **Delphi 5 – Análise de Pontos por Função**. Rio de Janeiro, Book Express, 2000.

(SIMÕES, 1999) Simões, C. **Sistemática de Métricas, qualidade e produtividade**. Developers' Magazine, Brasil, 1999. 7p.

(SIMÕES, 2004) Simões, C. **Implantação de uma Sistemática de Métricas: A Difícil Arte de Estimar Tempo para Implementação de Sistemas de Informações**. Trabalho de Conclusão de Pós-Graduação, Universidade Candido Mendes, Rio de Janeiro, RJ, 2004.

(SISP, 2012). **Roteiro de Métricas de Software do SISP: Versão 2.0**. Ministério do Planejamento, Orçamento e Gestão: Secretaria de Logística e Tecnologia da Informação, Brasília, 2012.

(SOFTEX, 2011) SOFTEX: MPS.BR – **Melhoria de Processo do Software Brasileiro** – Guia Geral:2001, 2011.

(VAZQUEZ et al., 2004) Vazquez, C. E.; Simões, G. S.; Albert, R. M. **Gestão de Contratos de Desenvolvimento de Software com a Análise de Pontos de Função**. Congresso Nacional da Sucesu, 2004.

(VAZQUEZ et al., 2005) Vazquez, C. E.; Simões, G. S.; Albert, R. M. **Análise de Pontos de Função**. 3a edição, São Paulo, Érica, 2005.

(VAZQUEZ et al., 2010) Vazquez, C. E.; Simões, G. S.; Albert, R. M. **Análise de Pontos de Função**. 10a edição, São Paulo, Érica, 2010.

(WEBER; LAYMAN, 2002) Weber, C.; Layman, B. **Measurement Maturity and the CMMI: How Measurement Practices Evolve as Process Mature**, American Society for Quality Magazines & Journals, v. 4, n. 3, 2002

(WIEGERS, 1997) Wiegers, K. E. **Software Metrics: Ten Traps To Avoid**, Software Process Impact - [www.processimpact.com](http://www.processimpact.com), 1997.

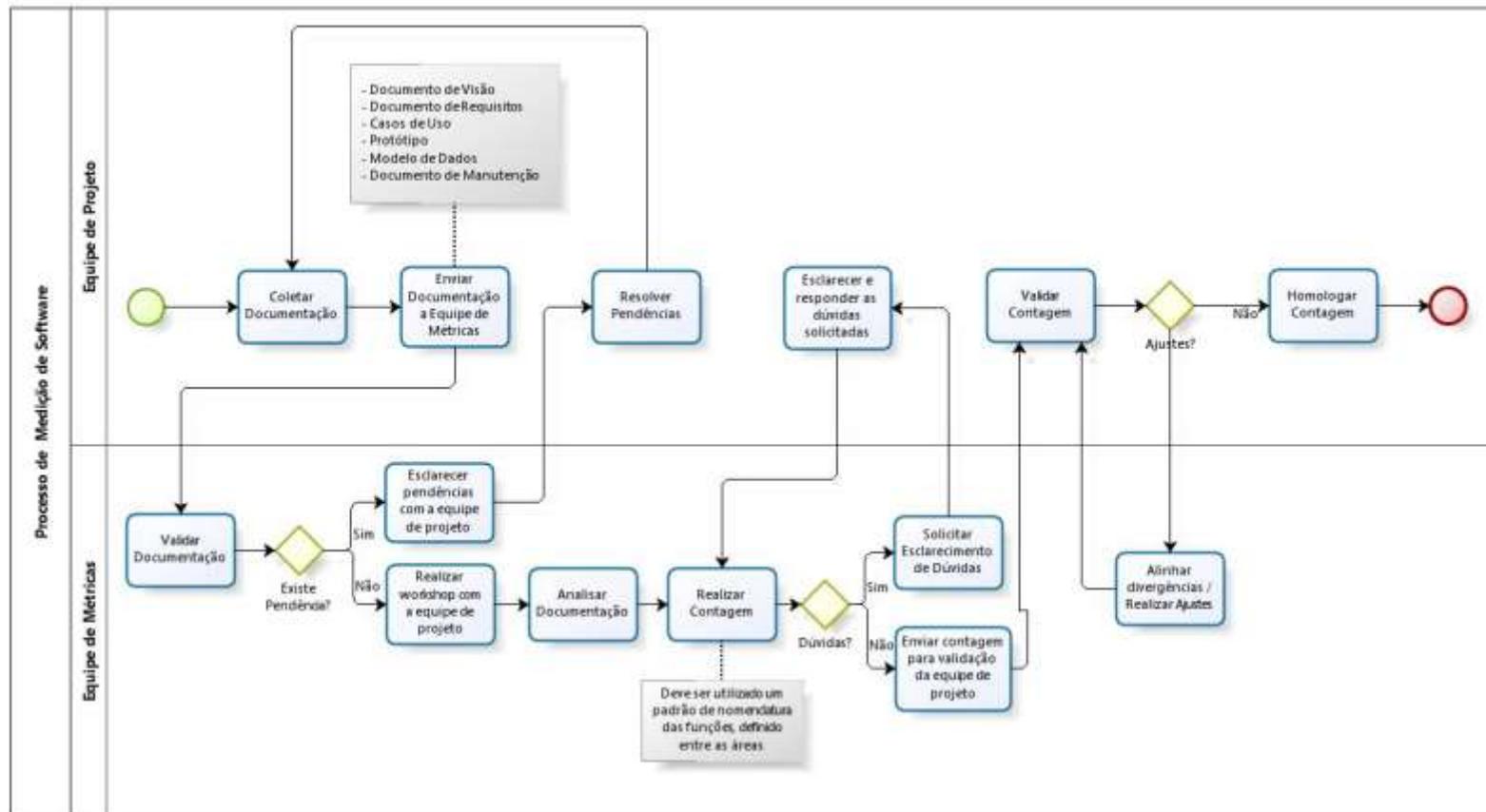
(WIEGERS, 1999) Wiegers, K. E. **A Software Metrics Primer**, Software Process Impact, 1999.

(XUNMEI et. al., 2006) Xunmei, G.; Guoxin, S.; Hong, Z. **The Comparison between FPA and COSMIC-FFP**, Software Measurement European Forum, 2006.

(YIN, 2010) Yin, R., Estudo de Caso: Planejamento e Métodos. Porto Alegre: Bookman, 2010.

# Apêndice

## Processo de Medição Definido para o Estudo de Caso



## **Descrição das Atividades do Processo de Medição**

Coletar Documentação: A documentação dos requisitos da aplicação a ser medida, especificada pela equipe do projeto e homologada pelo cliente, é coletada com o objetivo de ser enviada a equipe de métricas.

Enviar Documentação a Equipe de Métricas: Os documentos são disponibilizados através de diretório compartilhado ou e-mail, a serem utilizados como artefatos para a medição do software.

Validar Documentação: A equipe de métricas homologa se toda documentação necessária para medição do software foi disponibilizada pela equipe do projeto, e se foi elaborada com os requisitos mínimos para uma análise precisa.

Esclarecer Pendências com a Equipe de Projeto: Caso exista alguma pendência de documentação, após análise pela equipe de métricas, a equipe do projeto é acionada por e-mail com relatório das necessidades identificadas.

Resolver Pendências: As pendências identificadas são alinhadas entre as duas equipes envolvidas no processo, e os documentos ajustados ou não disponibilizados anteriormente são enviados a equipe de métricas.

Realizar Workshop com a Equipe de Projeto: Após validação da documentação enviada, a equipe do projeto realiza uma apresentação das funções do software a ser medido.

Analisar Documentação: A documentação homologada é analisada pela equipe de métricas para definição do tamanho funcional de cada função do software identificada.

Realizar Contagem: Medição das funções identificadas a partir dos documentos de requisitos funcionais disponibilizados, com análise e registro da documentação de contagem nas duas técnicas estudadas nesse trabalho.

Solicitar Esclarecimento de Dúvidas: As dúvidas geradas durante a análise de medição são enviadas por e-mail para equipe do projeto, ou esclarecidas de forma presencial.

Esclarecer e Responder as Dúvidas Solicitadas: Esclarecimento das dúvidas solicitadas pela equipe de métricas, através das respostas aos e-mails enviados ou de forma presencial.

Enviar Contagem para Validação da Equipe de Projeto: Após finalizada a documentação das contagens nas técnicas APF e COSMIC, essas são enviadas a equipe de projeto, para validação junto a equipe de métricas, se as funções foram identificadas corretamente.

Validar Contagem: Realização de uma reunião com as duas equipes participantes do processo, para apresentação das contagens realizadas nas duas técnicas estudadas nesse trabalho, e validação de cada função identificada.

Alinhar Divergências / Realizar Ajustes: Durante a reunião de validação é gerado um relatório com todos os ajustes a serem realizados. Os ajustes identificados para as contagens nas duas técnicas estudadas são realizados pela equipe de métricas, e enviados para homologação da equipe do projeto.

Homologar Contagem: Homologação pela equipe do projeto, das contagens executadas nas duas técnicas estudadas nesse trabalho, com todos os ajustes realizados.