

**APLICAÇÃO DE ANÁLISE DE PONTOS POR FUNÇÃO NA MENSURAÇÃO DE  
SOFTWARE COM SOA E BPM**

**VALÉRIA APARECIDA GUIMARÃES**

**FACULDADE DE TECNOLOGIA**  
**UNIVERSIDADE DE BRASÍLIA**

**UNIVERSIDADE DE BRASÍLIA**  
**FACULDADE DE TECNOLOGIA**  
**DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**APLICAÇÃO DE ANÁLISE DE PONTOS POR FUNÇÃO NA  
MENSURAÇÃO DE SOFTWARE COM SOA E BPM**

**VALÉRIA APARECIDA GUIMARÃES**

**ORIENTADOR: RICARDO STACIARINI PUTTINI**

**DISSERTAÇÃO DE MESTRADO EM ENGENHARIA ELÉTRICA**

**PUBLICAÇÃO: PPGENE.DM - 683/2017**  
**BRASÍLIA, DF, 20 de dezembro de 2017**

UNIVERSIDADE DE BRASÍLIA  
FACULDADE DE TECNOLOGIA  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

APLICAÇÃO DE ANÁLISE DE PONTOS POR FUNÇÃO NA  
MENSURAÇÃO DE SOFTWARE COM SOA E BPM

VALÉRIA APARECIDA GUIMARÃES

DISSERTAÇÃO DE MESTRADO SUBMETIDA AO DEPARTAMENTO DE ENGENHARIA ELÉTRICA DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA, COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE.

APROVADA POR:



RICARDO STACIARINI PUTTINI, Dr., ENE/UNB  
(ORIENTADOR)



RAFAEL TIMÓTEO DE SOUSA JUNIOR, Dr., ENE/UNB  
(EXAMINADOR INTERNO)



JULIANA PEREIRA DE SOUZA ZINADER, Dra., UFG  
(EXAMINADORA EXTERNA)

Brasília, 20 de dezembro de 2017.

## **FICHA CATALOGRÁFICA**

GUIMARÃES, VALERIA APARECIDA  
APLICAÇÃO DE ANÁLISE DE PONTOS POR FUNÇÃO NA MENSURAÇÃO  
DE SOFTWARE COM SOA E BPM / Valéria Aparecida Guimarães;  
Orientador Ricardo Staciarini Puttini. -- Brasília, 2017. 76 p.

Dissertação de Mestrado – Universidade de Brasília. Faculdade de Tecnologia.

Departamento de Engenharia Elétrica.

1. Análise de Pontos por Função (APF).

2. Arquitetura Orientada a Serviços (SOA)

3. Business Process Management (BPM)

I. Puttini, Ricardo Staciarini, orient.

II. Título.

## **REFERÊNCIA BIBLIOGRÁFICA**

GUIMARAES, V. A. (2017). Aplicação de Análise de Pontos por Função na Mensuração de Software com SOA e BPM. Dissertação de Mestrado em Engenharia Elétrica, Publicação PPPGEE.DM-683/2017, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 76p.

## **CESSÃO DE DIREITOS**

AUTOR: Valéria Aparecida Guimarães.

TÍTULO: Aplicação de Análise de Pontos por Função na Mensuração de Software com SOA e BPM.

GRAU: Mestre

ANO: 2017

É concedida à Universidade de Brasília permissão para reproduzir cópias desta dissertação de mestrado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte dessa dissertação de mestrado pode ser reproduzida sem autorização por escrito do autor.



Valéria Aparecida Guimarães

Rua Manoel Isidoro da Silveira, 610 apto 204 – Lagoa da Conceição  
Florianópolis - SC – Brazil – 88062-130

## **AGRADECIMENTOS**

Agradeço a todas as pessoas que me encorajaram e me ajudaram a produzir esse trabalho.

Ao professor Ricardo Puttini pela oportunidade, ensinamentos, orientações e incentivo.

A Gabriela e Juergen por ajudarem a criar as condições para realização deste trabalho.

Aos amigos da Core Consultoria, Alcione e Rafael por terem ajudado a construir os resultados desta pesquisa.

Ao Valério pelas orientações técnicas e acadêmicas.

A minha mãe e toda a minha família por estarem presentes em todos os momentos.

Ao Alexander pelo incentivo, paciência nos momentos de ansiedade e angústia e pela "co-orientação" indispensável à realização deste trabalho.

## **DEDICATÓRIA**

Dedico este trabalho ao meu irmão Carlinhos que não mediu esforços para permitir que eu alcançasse tudo o que sonhei. Saudades e gratidão eterna!

## RESUMO

### **Aplicação de Análise de Pontos por Função na Mensuração de Software com SOA e BPM**

Uma das técnicas de medição mais utilizadas e documentadas para a obtenção do tamanho de um software é a Análise de Pontos de Função (APF). No entanto, discussões acerca da sua aplicabilidade para projetos SOA (Arquitetura Orientada a Serviço) têm sido constantes, o que justificou a escolha desse tema de investigação. Este documento apresenta os resultados de uma pesquisa cujo objetivo foi a construção de uma proposta para a medição, por meio da utilização de Pontos de Função (PF), do tamanho funcional de softwares que utilizam as abordagens de desenvolvimento de software com SOA e BPM (*Business Process Management*). A proposta foi desenvolvida a partir de revisão da literatura e de análises de dados empíricos coletados durante a execução de um projeto real. Ao final do trabalho, foi possível realizar análises comparativas entre projeções usando APF, o método proposto e dados empíricos. O resultado mostrou que a APF não é adequada para a medição do tamanho funcional de projetos SOA/BPM, ao contrário do método proposto.

**Palavras-chave:** Estimativa de Software, Pontos de Função, SOA, APF, BPM

## **ABSTRACT**

### **Function Point Analysis Adoption in the Measurement of Software based on SOA and BPM**

One of the most widely used and documented measurement techniques for obtaining software size is Function Point Analysis (FPA). However, there have been frequently discussions about its applicability to SOA (Service Oriented Architecture) projects, which justifies a choice of the research topic. This document presents the results of a research whose objective was the construction of a proposal to measure SOA and BPM (Business Process Management) softwares through the use of Function Points (PF). A proposal has been developed based on literature review and empirical data analyzes, with was collected from an execution of a real project. The research compare empirical data (from a real-world project) to projections based on FPA, as well on the proposed method. Results demonstrate that the proposed method is suitable to SOA/BPM projects, differently of FPA.

**Keywords:** Software Estimates, Function Point, FPA, SOA, BPM

# ÍNDICE

1	INTRODUÇÃO.....	1
1.1	Contextualização .....	1
1.2	Análise de Pontos por Função .....	2
1.3	Arquitetura Orientada a Serviços (SOA).....	2
1.4	Análise de Pontos de Função em Arquitetura Orientada a Serviços (SOA) .....	3
1.5	Objetivo .....	5
1.6	Objetivo Geral .....	5
1.7	Objetivos Específicos .....	5
1.8	Motivação do Pesquisador .....	6
1.9	Trabalhos Relacionados.....	6
1.10	Organização do Trabalho.....	8
2 -	METODOLOGIA DE PESQUISA .....	9
2.1	Projetos de Pesquisa .....	9
2.2	Concepções Filosóficas .....	11
2.3	Estratégia de Investigação .....	13
2.4	Métodos de Pesquisa .....	14
2.5	Aspectos Operacionais .....	15
2.6	Construção do Referencial Teórico .....	15
2.7	Pesquisa-Ação .....	16
2.7.1.1	Intervenção .....	16
2.7.1.2	Proposta e Registro.....	17
3 -	FUNDAMENTAÇÃO TEORICA .....	19
3.1	Conceitos e Definições Relacionadas a PF.....	19
3.2	Análise de pontos de função - IFPUG .....	21
3.3	Introdução.....	21
3.4	Etapas APF .....	22
3.5	Obter a Documentação Disponível.....	23
3.6	Identificar Propósito, Tipo da Contagem, Escopo e Fronteira da Aplicação .	23
3.7	Medir Funções de Dados .....	24
3.8	Medir as Funções de Transação.....	26
3.9	Calcular o Tamanho Funcional.....	29
3.10	Documentar e Reportar a Contagem .....	29

3.11	Arquitetura Orientada a Serviço .....	30
3.12	Princípios SOA .....	31
3.13	Objetivos Estratégicos SOA .....	34
3.14	Modelos de Serviço .....	36
3.15	Modelos Canônicos de Dados .....	38
4 -	INTERVENÇÃO.....	40
4.1	Introdução.....	40
4.2	Resultados.....	41
4.3	Projeto de Referência.....	41
4.4	Equipe de Projeto .....	42
4.5	Esforço Realizado.....	43
4.6	Custo Realizado.....	43
4.7	Resultado Geral .....	44
4.8	Medição do Tamanho Funcional – APF.....	44
4.9	Análise dos Resultados - APF .....	48
5 -	APRESENTAÇÃO DA PROPOSTA .....	49
5.1	Pilares .....	49
5.2	Descrição do Método APFSOA .....	50
5.3	Contar as Funções de Dados.....	50
5.4	Contar as Funções de Transação .....	52
5.5	Calcular o Tamanho Funcional.....	54
5.6	Exemplo de Serviço de Entidade.....	54
5.7	Exemplo de Serviço de Tarefa.....	57
5.8	Exemplo de Serviços de Processo .....	61
5.9	Medição do Projeto por Meio da Aplicação do APFSOA.....	63
5.10	Análise dos Resultados.....	64
6 -	CONSIDERAÇÕES FINAIS .....	66
6.1	Sobre a Condução da Pesquisa .....	66
6.2	Perspectiva Profissional <i>versus</i> Perspectiva Acadêmica.....	66
6.3	Principais Desafios .....	66
6.4	Sobre o Uso da APF em Projetos SOA .....	67
6.5	Sobre a Proposta - APFSOA .....	67
6.6	Aplicação da Proposta .....	68
6.7	Projeto Real, APF e APFSOA.....	69

6.8	Sobre o Cumprimento dos Objetivos da Pesquisa.....	69
6.9	Limitações e Trabalhos Futuros .....	70
REFERÊNCIAS BIBLIOGRÁFICAS .....		72

## LISTA DE TABELAS

Tabela 2-1: Abordagens Quantitativas e Qualitativas .....	10
Tabela 2-2: Quatro Concepções .....	12
Tabela 2-3: Metodologias de Pesquisa.....	13
Tabela 2-4: Métodos de Pesquisa.....	14
Tabela 3-1: Conceitos APF .....	19
Tabela 3-2: Complexidade das Funções de Dados.....	25
Tabela 3-3: Tamanho das Funções de Dados .....	26
Tabela 3-4: Relação entre a Intenção Primária e o Tipo de Função de Transação.....	28
Tabela 3-5: Complexidade Funcional das EE .....	28
Tabela 3-6: Complexidade Funcional das SE e CE .....	29
Tabela 3-7: Tamanho Funcional da Função de Transação .....	29
Tabela 4-1: Expoente por Tipo de Projeto .....	47

## LISTA DE FIGURAS

Figura 2-1: Elementos Envolvidos na Definição da Abordagem do Projeto de Pesquisa.....	11
Figura 2-2: Procedimentos Metodológicos .....	15
Figura 3-1: Etapas APF IFPUG .....	23
Figura 3-2: Princípios SOA.....	32
Figura 3-3: Objetivos Estratégicos de SOA .....	35
Figura 3-4: Modelo de Serviços .....	37
Figura 3-5: Modelo de Entidades/Canônico.....	39
Figura 4-1: Metodologia de Desenvolvimento SOA.....	41
Figura 5-1: Etapas contagem de PF em Projetos SOA/BPM .....	49
Figura 5-2: Exemplo Serviço de Processo .....	54
Figura 5-3: BPEL Serviço <i>processarAfastamentoDeferidoService</i> .....	59
Figura 5-4: Serviço de Processo <i>registrarAfastamentoService</i> .....	61

## LISTA DE QUADROS

Quadro 1-1: Trabalhos Relacionados .....	7
Quadro 4-1: Equipe de Projeto.....	43
Quadro 4-2: Esforço Realizado .....	43
Quadro 4-3: Custo Realizado .....	44
Quadro 4-4: Resultado Geral.....	44
Quadro 4-5: Resultado Contagem IFPUG .....	45
Quadro 4-6: Derivação de Prazo, Esforço, Equipe e Custo .....	46
Quadro 5-1: Heurísticas Funções de Dados .....	51
Quadro 5-2: Identificação Grupo lógico de Dados .....	52
Quadro 5-3: Contrato do Serviço <i>DesignacaoMagistradoService</i> .....	55
Quadro 5-4: Funções de Transação Serviço <i>designacaoMagistradoService</i> .....	56
Quadro 5-5: Funções de Dados do Serviço <i>designacaoMagistrado</i> .....	57
Quadro 5-6: Total de PFSOAs do Serviço <i>designacaoMagistrado</i> .....	57
Quadro 5-7: Contrato do Serviço <i>processarAfastamentoDeferidoService</i> .....	57
Quadro 5-8: Funções de Transação Serviço <i>processarAfastamentoDeferidoService</i> .....	60
Quadro 5-9: Funções de Dados Serviço <i>processarAfastamentoDeferidoService</i> .....	61
Quadro 5-10: PFSOA Serviço <i>processarAfastamentoDeferidoService</i> .....	61
Quadro 5-11: Funções de Transação do Serviço <i>registrarAfastamentoService</i> .....	62
Quadro 5-12: Funções de Dados do Serviço <i>registrarAfastamentoService</i> .....	63
Quadro 5-13: PFSOA Serviço <i>registrarAfastamentoService</i> .....	63
Quadro 5-14: Tamanho Funcional em PFSOA para o PROJETO DE REFERÊNCIA .....	64
Quadro 5-15: Derivação de Custo PROJETO DE REFERÊNCIA (APFSOA) .....	64
Quadro 6-1: Comparação Projeto Real, APF e APFSOA .....	69
Quadro 6-2: Objetivos Específicos da Pesquisa.....	70

## ACRÔNIMOS

AIE	Arquivo de Interface Externa
ALI	Arquivo Lógico Interno
ALR	Arquivo Lógico Referenciado
APF	Análise de Pontos por Função
APFSOA	Análise de Pontos de Função SOA
BPEL	Business Process Execution Language
BPM	Business Process Management
BPMN	Business Process Modeling Notation
BPMN-AP	Business Process Modeling Notation Activity Points
BPMS	Business Process Management System
CE	Consulta Externa
COSMIC	Common Software Measurement International Consortium
CPM	Counting Practices Manual
DER	Dado Elementar Referenciado
EE	Entrada Externa
IFPUG	International Function Point Users Group
ISO	International Organization for Standardization
NESMA	Netherlands Software Metrics Users Association
PA	Pesquisa-Ação
PF	Pontos de Função
PDSOA	Pontos de Função SOA
QTD	Quantidade
RLR	Registro Lógico Referenciado
SE	Saída Externa
SOA	Service-Oriented Architecture
WSDL	Web Service Description Language
XML	Extensible Markup Language
XSD	XML Schema Definition

# 1 INTRODUÇÃO

## 1.1 CONTEXTUALIZAÇÃO

Estimar o tamanho de um software é indispensável para se obter uma estimativa confiável de custos e prazos para seu desenvolvimento (WILKIE, 2011), mas realizar adequadamente esse tipo de estimativa é apontado pela literatura como um dos maiores desafios enfrentados por gestores dessa área (PRESSMAN, 2011; WILKIE, 2011; KAUR, 2016). É o que mostra, por exemplo, Roetzheim (2000), que conduziu, durante 18 anos, uma extensa pesquisa que concluiu que as maiores causas de fracassos de projetos de software estão relacionadas a uma pobre estimativa de custo e cronograma de execução, e não a problemas técnicos, políticos ou de desenvolvimento da equipe.

Apesar de não ser uma ciência exata, devido à influência de variáveis humanas, técnicas, ambientais e políticas (SOARES, 2013), técnicas de medição (e métricas) de software podem contribuir, significativamente, com a geração de estimativas mais assertivas, o que diminui a discrepância entre o que foi estimado e o que foi realizado, bem como permite detectar tendências e antecipar problemas, proporcionando um controle de custo mais eficiente; reduzindo riscos; melhorando a qualidade e garantindo que os objetivos de negócio sejam alcançados (FLORAC,1997; PRESSMAN, 2011; TICHENOR, 2013; SOARES, 2013). Contribui, ainda, para uma melhor compreensão a respeito dos processos que compõem um projeto e ressalta os pontos mais importantes (FLORAC,1997), além de permitir um maior controle sobre o andamento do projeto-- ao proporcionar a realização de um planejamento mais próximo da realidade (SOARES, 2013). Segundo (PRESSMAN, 2011 p.583):

“A medição nos permite obter o entendimento do processo e projeto, dando-nos um mecanismo para avaliação objetiva. Medições contribuem na determinação de estimativas, controle de qualidade, produtividade e controle do projeto”.

Uma das técnicas de medição mais utilizadas e documentadas para a obtenção do tamanho funcional de um software é a Análise de Pontos de Função (APF) (da sigla em inglês FPA, *Function Point Analysis*) (IFPUG, 2010; WILKIE, 2011; SOARES, 2013). Este método foi publicado pela primeira vez em meados da década de 1970, por Allan Albrecht

(ALBRECHT, 1984). Desde sua criação, esse método vem sendo aprimorado por organizações como *IBM CIS & A Guideline 313, AD/M Productivity Measurement and Estimate Validation, International Function Points Group - IFPUG* (IFPUG, 2010), *Common Software Measurement International Consortium - COSMIC* (COSMIC, 2015) e pela *Netherlands Software Metrics Users Association - NESMA* (NESMA, 2015). Dentre estes, ganhou maior relevância o método desenvolvido pelo IFPUG.

Desde a proposta inicial, apresentada em 1986, o IFPUG trabalha fortemente no aprimoramento contínuo, buscando a evolução do método, o qual é conhecido atualmente como APF (IFPUG, 2010) e se encontra consolidado e é amplamente utilizado em todo o mundo (IFPUG, 2010; CALAZANS, 2011; SISP, 2015).

## **1.2 ANÁLISE DE PONTOS POR FUNÇÃO**

APF é uma técnica para a medição de tamanho funcional de software, quando avaliado do ponto de vista do usuário. O método utiliza Ponto de Função (PF) como unidade de medida (IFPUG, 2010) e o IFPUG publica e mantém um conjunto de padrões e práticas de contagem, estruturadas no Manual de Práticas de Contagem (no inglês, *CPM – Counting Practices Manual*) (IFPUG, 2010). A medição do tamanho não-funcional não é contemplada pela APF, o que não significa que não possa ser medido, mas apenas que deve ser tratado com uma medida separada, o qual utiliza métodos próprios para medição (IFPUG, 2010).

APF é frequentemente aplicada para: (i) estimar esforço, custo e prazo requeridos para o desenvolvimento, manutenção e melhoria do software; (ii) fornecer suporte à análise de qualidade e produtividade; (iii) fornecer um fator de normalização para a comparação de tamanho funcional de software (IFPUG, 2010). Sua utilização apresenta diversos benefícios, dos quais destacam-se: regras de contagem objetivas, independência da solução tecnológica e linguagem de programação e possibilidade de geração de estimativa já nas fases iniciais do ciclo de vida do software (SISP, 2015).

## **1.3 ARQUITETURA ORIENTADA A SERVIÇOS (SOA)**

SOA é uma abordagem de arquitetura para desenvolvimento de software cujo princípio fundamental é a construção de serviços reutilizáveis e interoperáveis (ERL, 2005). De acordo com ERL (2005), Serviço é unidade fundamental da lógica orientada a serviço (lógica

da solução), ou seja, as funcionalidades da aplicação são disponibilizadas na forma de serviços.

Uma lógica é considerada orientada a serviço quando alguns princípios, conhecidos como princípios SOA, são aplicados em uma extensão significativa da solução (ERL, 2005). De acordo com Erl (2008) são oito estes princípios: (i) padronização de contrato; (ii) baixo acoplamento de serviços; (iii) abstração de serviço; (iv) capacidade de reuso de serviço; (v) autonomia de serviço; (vi) independência de estado de serviço; (vii) visibilidade de serviço (capacidade de descoberta); e (viii) capacidade de composição de serviço.

SOA busca, por meio da aplicação de tais princípios, aumentar o alinhamento entre negócio e TI; a federalização (o acesso aos serviços é padronizado de forma a unificar a visão dos seus consumidores); a diversidade de fornecedores; o retorno do investimento (ROI) e a agilidade organizacional, bem como reduzir o peso da TI na organização (ERL, 2008). SOA permite construir aplicações que respondam mais rapidamente às demandas de negócio, uma vez que serviços são construídos de maneira padronizada, sendo capazes de interoperar (comunicar) uns com os outros de maneira padrão e transparente, independentemente de plataforma, fornecedores ou tecnologias que em foram construídos ou são executados.

#### **1.4 ANÁLISE DE PONTOS DE FUNÇÃO EM ARQUITETURA ORIENTADA A SERVIÇOS (SOA)**

Apesar do uso frequente da APF como técnica para dimensionamento do tamanho funcional de softwares (IFPUG, 2010; CALAZANS, 2011; SISP, 2015) e do esforço contínuo para aprimoramento e evolução do método (LINDSKOOG, 2009; IFPUG, 2010), há cenários que sua adoção direta não é evidente.

Discussões acerca da aplicabilidade do método para projetos SOA têm sido constantes e muitos desafios ainda precisam ser vencidos para definição de uma proposta para medição do tamanho de projetos SOA por meio da utilização de PF (GENCEL, 2008; LINDSKOOG, 2009; GOMES, 2012). Esta dificuldade decorre de diversos fatores, dentro os quais se destacam:

- A APF trata apenas os requisitos funcionais, sendo que os requisitos não-funcionais devem ser dimensionados utilizando uma técnica diferente de medição (IFPUG, 2010). Do ponto de vista dos usuários finais, os requisitos funcionais pertencem

apenas à camada de apresentação, limitando a análise de pontos de função a essa camada. Projetos SOA possuem, normalmente, uma quantidade significativa de componentes não-funcionais e uma análise restrita à camada de apresentação pode levar a um resultado que não representa todo o trabalho realizado na construção do projeto (LINDSKOOG, 2009).

- Serviços e aplicações são construídos e executam sobre uma plataforma (*middleware*) de mediação de mensageria (barramento) e de orquestração (ERL, 2005).
- Em projetos de SOA, serviços são projetados para o reuso. Desse modo, é muito comum que o escopo do projeto de serviço extrapole o escopo funcional da aplicação que está sendo desenvolvida (ERL, 2005).
- Em SOA, sistemas complexos precisam ser decompostos em serviços. Boa parte do esforço de análise e projeto está nesse processo de decomposição e posterior recomposição. Esse processo requer a aplicação explícita de métodos e princípios para obtenção de baixo acoplamento, capacidade de composição, padronização e mediação de transações, entre outros, que não estão presentes no desenvolvimento de software convencional (ERL, 2008). Desse modo, as métricas e *benchmarks* de estimativa de esforço com base em pontos de função, utilizadas para o desenvolvimento tradicional, não podem ser aplicadas diretamente (FARRAG, 2015).
- O processamento em uma aplicação com SOA é mais distribuído que o processamento em aplicações convencionais. Tal processamento distribuído tende a ser mais complexo do que o processamento monolítico convencional (não distribuído) (JOSUTTIS, 2007).
- SOA enfatiza o reuso de software, estando boa parte dos seus benefícios estratégicos (redução de custo, agilidade e redução de redundâncias) relacionada à capacidade de aplicação dessa prática. Nesse sentido, serviços são colecionados em um inventário, com governança própria, para serem reutilizados com mínimas adaptações. A reutilização de serviços reduz o esforço de desenvolvimentos futuros (ERL, 2008). Entretanto, esta redução não é contabilizada no cálculo convencional da APF.

- Outro fator que agrega desafios à mensuração de projetos SOA está relacionado ao paradigma BPM (*Business Process Management*), o qual é frequentemente utilizado em projetos SOA (ADAM, 2008). BPM é uma abordagem sistematizada para entendimento, análise, modelagem e gerenciamento de processos de negócio (ABPMP, 2015). Diversos estudos relacionados à mensuração de projetos BPM podem ser encontrados na literatura (BAKLIZKY, 2012; BAKLIZKY, 2013; BAKLIZKY, 2016; GOMES, 2016); no entanto, não foram identificadas práticas consolidadas no mercado, principalmente quando se trata de projetos SOA.

Em síntese, ainda é um grande desafio a estimativa de esforço, custo e prazo para o desenvolvimento de projetos SOA, uma vez que a literatura não apresenta métodos ou técnicas consolidadas e validadas que permitam a geração de estimativas de maneira consistente. Devido às suas especificidades, tais como baixo acoplamento, reutilização e composição de serviços, as técnicas tradicionais não se aplicam adequadamente quando softwares são desenvolvidos utilizando SOA (FARRAG, 2015).

Este trabalho investiga o uso de APF no dimensionamento funcional de projetos de software que seguem a abordagem de arquitetura orientada a serviço (SOA) - em especial quando esta é combinada com a abordagem de gerenciamento de processos de negócio (BPM), que é uma abordagem sistematizada para entendimento, análise, modelagem e gerenciamento de processos de negócio (ABPMP, 2015).

## **1.5 OBJETIVO**

### **1.6 Objetivo Geral**

A presente pesquisa tem por objetivo o desenvolvimento de proposta para mensuração de soluções de software SOA e BPM com utilização de PF, através de adaptação do método do IFPUG e aplicação em um projeto real.

### **1.7 Objetivos Específicos**

- Compilar conhecimento sobre da aplicação do método desenvolvido pelo IFPUG (APF) para medição do tamanho funcional de software, tanto tradicionais quanto projetos que utilizam a abordagem SOA.

- Avaliar o método do IFPG para a medição do tamanho de um software desenvolvido integralmente utilizando a abordagem SOA.
- Propor, a partir da combinação entre os conceitos do IFPUG e as especificidades da abordagem SOA, um método para a medição do tamanho funcional de softwares com SOA.
- Aplicar e validar empiricamente o método proposto.
- Contribuir para a geração de conhecimento científico relacionado à utilização de PF para a medição do tamanho funcional de softwares SOA

## **1.8 MOTIVAÇÃO DO PESQUISADOR**

Sou consultora em uma empresa especializada em SOA, a qual iniciou em abril de 2015 a prestação de serviço em uma empresa pública que dava início as atividades para a efetiva adoção da Orientação a Serviço.

Durante toda a execução do projeto, fomos questionados a respeito do esforço, prazo e custo estimados para a execução das atividades. A contratante teve dificuldades em entender tais estimativas e não tínhamos argumentos sólidos para justificar a falta de assertividade, uma vez que nenhum método padrão fora utilizado para a medição do tamanho funcional do software. A contratante utiliza PF para medição dos projetos desenvolvidos utilizando a abordagem tradicional de desenvolvimento de software e gostaria de utilizar a mesma métrica para a medição de projetos SOA.

A métrica de PF tem sido amplamente utilizada por instituições públicas e privadas para dimensionamento de tamanho funcional de projetos de software (SISP, 2015) e percebemos sua importância no contexto real de prestação de serviços para o setor público.

A necessidade real e o interesse pré-existente pela abordagem SOA foram os ingredientes que motivaram a condução desta pesquisa.

## **1.9 TRABALHOS RELACIONADOS**

Com o objetivo de entender e levantar os diversos problemas relacionados à aplicação de PF para medição de projetos SOA, foram analisadas diversas propostas para medição do

tamanho de softwares não tradicionais, como, por exemplo, softwares reutilizáveis, *middleware*, BPM (*Bussiness Processo Management*) e, mais especificamente, baseados em SOA. O Quadro 1-1 apresenta os trabalhos utilizados como fundamentos para essa pesquisa.

Quadro 1-1: Trabalhos Relacionados

Woodward (2009a)	Com base nas práticas definidas pelo IFPUG, apresenta uma proposta para utilização da APF para medição de uma aplicação de <i>middleware</i> , o qual é responsável pela mediação entre duas aplicações.
Woodward (2009b)	Apresenta uma proposta para medição do tamanho software baseado em componentes. Sua proposta considera as regras definidas pelo IFPUG e as características de um componente de software, o qual é construído para ser reutilizado na construção de aplicações maiores para atendimento de requisitos de negócio.
Woodward (2010)	Demonstra como o método do IFPUG pode ser utilizado para quantificar softwares reutilizáveis, considerando as perspectivas dos desenvolvedores de componentes e usuários finais das aplicações.
Gomes (2012)	Propõe uma abordagem para medição de aplicações SOA, considerando que muitos projetos não possuem as interfaces de usuário e modelos de dados utilizados na abordagem tradicional de desenvolvimento de software, ou seja, sua contagem é baseada nos modelos de serviços e modelo de dados canônico. Em sua proposta, a medição é realizada com base em artefatos como o documento de candidatos a serviço, modelo canônico de dados (XSD) e contratos de serviço (WSDL), os quais considera suficientes para prover uma contagem detalhada do software. Sua proposta é baseada nas regras definidas pelo IFPUG e nas três principais camadas de software (modelos de serviços) tradicionalmente utilizadas em SOA: serviços utilitários, serviços de entidade e serviços de tarefa, sendo que cada camada possui regras de contagem próprias.
Lindskoog (2009)	Propõe uma abordagem para a utilização da análise de pontos de função como medida de dimensionamento eficaz em projetos SOA.
Gomes (2016)	Propõe uma métrica para medição de tamanho de software, denominada BPMN <i>Activity Points</i> (BPMN-AP), que se baseia na contagem das atividades que compõem os processos de negócio de acordo com a notação BPMN.
Baklizky (2012; 2013; 2016)	Apresentam diferentes técnicas desenvolvidas com base no Manual de Práticas de Contagem (CPM – <i>Counting Practices Manual</i> ) para aplicação de Pontos de Função (PF) ao domínio de Gestão de Processos de Negócio (BPM).

Fonte: Elaboração Própria

As diversas propostas analisadas contribuíram fortemente para o desenvolvimento do método proposto neste trabalho. No entanto, nenhuma das propostas analisadas apresenta claramente um método para a contagem de projetos desenvolvidos utilizando simultaneamente as abordagens SOA e BPM. Além disso, os trabalhos apresentados não demonstraram sua aplicabilidade em projetos reais, não incluindo, portanto, comparações entre o custo real e o estimado de um projeto SOA/BPM. Na presente pesquisa, são

apresentadas a medição em PF (aplicação do método APF IFPUG) e PFSOA (unidade proposta por este trabalho). As projeções de ambas as medições são comparadas aos resultados obtidos empiricamente.

## **1.10 ORGANIZAÇÃO DO TRABALHO**

O restante do documento está assim estruturado: O Capítulo 2 apresenta os aspectos metodológicos que orientaram a pesquisa e os motivos que levaram a escolha das estratégias de condução da investigação; inclui, ainda, uma visão geral dos procedimentos metodológicos adotados. O Capítulo 3 apresenta uma compilação dos termos e conceitos relacionados à SOA e PF. O Capítulo 4 detalha o planejamento e a execução da intervenção (pesquisa de campo). O Capítulo 5 apresenta a descrição da proposta para medição do tamanho funcional de projetos SOA/BPM. Por fim, o Capítulo 6 apresenta os comentários finais e conclusões da pesquisa.

## **2 - METODOLOGIA DE PESQUISA**

### **2.1 PROJETOS DE PESQUISA**

A metodologia de pesquisa é um dos fatores determinantes para a qualidade de uma investigação científica. Está diretamente relacionada à maneira como a pesquisa é conduzida, ou seja, fornece as diretrizes a serem seguidas pelo pesquisador no que se refere à utilização de teorias; conceitualização dos problemas; formulação de hipóteses; construção, verificação e validação de modelos; e tomada de decisões de forma adequada (THIOLLENT, 1983).

Investigações científicas são disparadas a partir da formulação de problemas, antigos ou novos, e da busca por novas soluções (KNELLER, 1980). O plano, as suposições e os procedimentos que serão usados no processo investigativo formam o projeto de pesquisa, que pode assumir uma abordagem qualitativa, quantitativa ou mista (CRESWELL, 2010).

Uma pesquisa quantitativa é indutiva, determinística e baseada em levantamentos e experimentos. Geralmente busca criar relações do tipo causa-efeito a partir de dados numéricos e tratamento estatístico (BRYMAN 2004; CRESWELL, 2010). No outro extremo, a pesquisa qualitativa é dedutiva, baseada na observação e interpretação (ponto de vista do pesquisador) de fenômenos - a partir de imagens e textos. Entre as duas, situa-se a abordagem de métodos mistos (MAXWELL, 2008; CRESWELL, 2010). A Tabela 2-1 apresenta, resumidamente, as características das abordagens quantitativas e qualitativas:

Tabela 2-1: Abordagens Quantitativas e Qualitativas

	Pesquisa quantitativa	Pesquisa qualitativa
<b>Inferência</b>	Dedutivo	Indutivo
<b>Objetivo</b>	Comprovação	Interpretação
<b>Finalidade</b>	Teste de teorias, predição, estabelecimento de fatos e teste de hipóteses	Descrição e entendimento de realidades variadas, captura da vida cotidiana e perspectivas humanas
<b>Realidade investigada</b>	Objetiva	Subjetiva e complexa
<b>Foco</b>	Quantidade	Natureza do objeto
<b>Amostra</b>	Determinada por critério estatístico	Determinada por critérios diversos
<b>Característica da amostra</b>	Grande	Pequena
<b>Característica do instrumento de coleta de dados</b>	Questões objetivas, aplicações em curto espaço de tempo. Evita-se a interação entrevistador-entrevistado,	Questões abertas e flexíveis. Explora a interação pesquisador-entrevistado.
<b>Procedimentos</b>	Isolamento de variáveis. Anônima aos participantes.	Examina todo o contexto, interage com os participantes.
<b>Análise dos dados</b>	Estatística e numérica.	Interpretativa e descritiva. Ênfase na análise de conteúdo.
<b>Plano de pesquisa</b>	Desenvolvido antes de o estudo ser iniciado. Proposta estruturada e formal.	Evolução de uma idéia com o aprendizado. Proposta flexível.
<b>Resultados</b>	Comprovação de hipóteses. A base para generalização dos resultados é universal e independente do contexto.	Proposições e especulações. Os resultados são situacionais e limitados ao contexto.
<b>Confiabilidade e validade</b>	Pode ser determinada, dependendo do tempo e recurso.	Difícil determinação, dada à natureza subjetiva da pesquisa.

Fonte: Terence (2006)

De acordo com Creswell (2010), a definição da abordagem (quantitativa, qualitativa ou método misto) a ser utilizada no desenvolvimento da pesquisa envolve a análise conjunta de três diferentes elementos: (i) filosofia (ou concepções filosóficas); (ii) estratégia de investigação (ou metodologia de pesquisa) e (iii) métodos específicos (vide Figura 2-1).

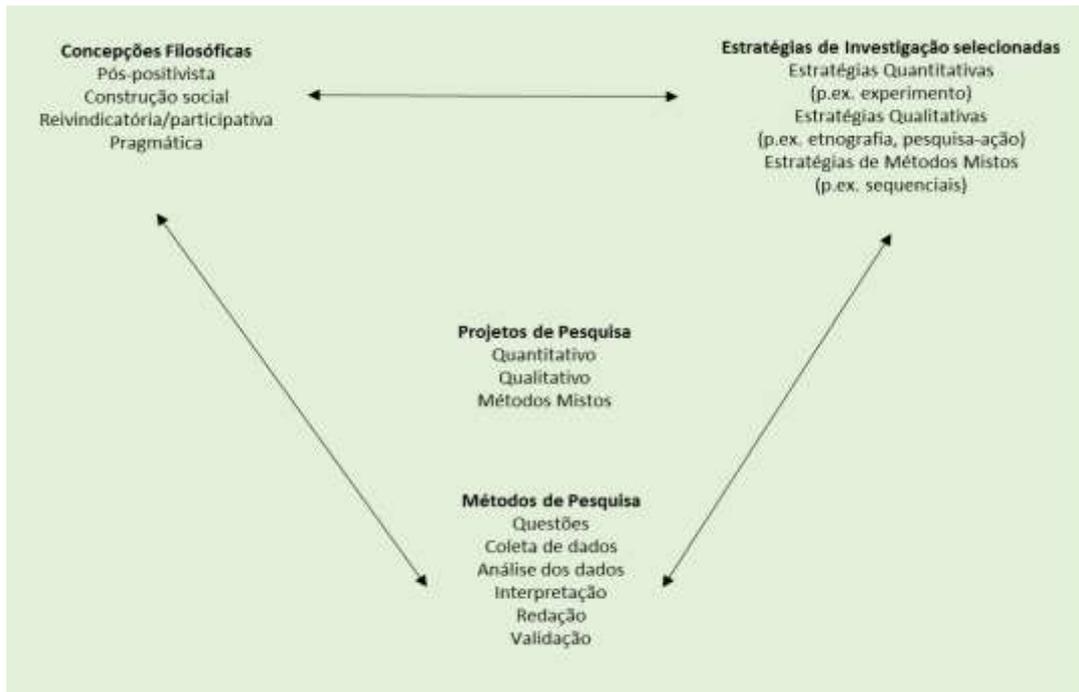


Figura 2-1: Elementos Envolvidos na Definição da Abordagem do Projeto de Pesquisa  
 Fonte: Adaptado de CRESWELL, 2010 p.28

## 2.2 Concepções Filosóficas

O termo concepção é aqui utilizado significando "um conjunto de crenças básicas que guiam a ação" (CRESWELL, 2010 p.28) e são moldadas pela área (assunto) de pesquisa do aluno, pelas crenças dos orientadores e dos professores na área de pesquisa do aluno e pelas experiências que tiveram em pesquisa. Creswell (2010) apresenta quatro diferentes tipos de concepções: pós-positivista, construtivista reivindicatória/participatória e pragmática. A Tabela 2-2 ilustra os principais elementos de cada posição.

Tabela 2-2: Quatro Concepções

<b>pós-positivista</b>	<b>Construtivista</b>
<ul style="list-style-type: none"> <li>• Determinação</li> <li>• Reduccionismo</li> <li>• Observação e mensuração empíricas</li> <li>• Verificação da teoria</li> </ul>	<ul style="list-style-type: none"> <li>• Entendimento</li> <li>• Significados múltiplos do participante</li> <li>• Construção social e histórica</li> <li>• Geração de teoria</li> </ul>
<b>Reivindicatória/Participatória</b>	<b>Pragmatista</b>
<ul style="list-style-type: none"> <li>• Política</li> <li>• Capacitação orientada para a questão</li> <li>• Colaborativa</li> <li>• Orientada para a mudança</li> </ul>	<ul style="list-style-type: none"> <li>• Conseqüências das ações</li> <li>• Centrada no problema</li> <li>• Pluralista</li> <li>• Orientada para a prática no mundo real</li> </ul>

Fonte: CRESWELL, 2010 p.29

A concepção pós-positivista, também conhecida como método científico e utilizada na abordagem quantitativa, defende uma filosofia determinística, onde as causas provavelmente determinam os efeitos ou os resultados. As suposições pós-positivistas são também consideradas reducionista na medida em que buscam reduzir as ideias um conjunto pequeno e distinto a serem testadas (CRESWELL, 2010).

A concepção reivindicatória e participatória, tipicamente parte da abordagem qualitativa, mas podendo também ser encontrada como base da abordagem quantitativa, defende que a investigação da pesquisa precisa estar associada a política e questões sociais atuais, como capacitação, desigualdade, opressão, dominação, supressão e alienação. O foco dessa concepção está nas necessidades dos indivíduos, parte da nossa sociedade, que por ventura possam estar marginalizados ou sem privilégios (CRESWELL, 2010).

O interpretativismo, também conhecido como concepção construtivista social, preconiza a ideia de que os indivíduos possuem capacidade para interpretar o mundo no qual vivem, sendo capazes de desenvolver significados subjetivos a partir de suas experiências. Nesta concepção, a visão dos participantes a respeito do assunto estudado ganha grande relevância (BURREL & MORGAN,1979; CRESWELL, 2010).

A concepção pragmática foi a concepção filosófica adotada nesta pesquisa. Nela, os pesquisadores estão preocupados com o problema da pesquisa e utilizam-se de todas as abordagens disponíveis para desvendá-lo. Frequentemente esta é a concepção que se aplica à pesquisa de métodos mistos, ou seja, quando os investigadores se baseiam tanto nas suposições quantitativas quanto nas qualitativas (CRESWELL;2010).

### 2.3 Estratégia de Investigação

O segundo elemento determinante para a definição da abordagem a ser utilizada na pesquisa é a estratégia de investigação (CRESWELL, 2010). Também conhecida como metodologia de pesquisa, Estratégia de Investigação são as abordagens escolhidas para condução da pesquisa. A Tabela 2-3 ilustra as principais estratégias apresentadas por CRESWELL (2010).

Tabela 2-3: Metodologias de Pesquisa

Quantitativa	Qualitativa	Métodos mistos
<ul style="list-style-type: none"><li>• Projetos experimentais</li><li>• Projetos não experimentais, como os levantamentos</li></ul>	<ul style="list-style-type: none"><li>• Pesquisa narrativa</li><li>• Fenomenologia</li><li>• Etnografias</li><li>• Estudos de teoria fundamentada</li><li>• Estudo de caso</li></ul>	<ul style="list-style-type: none"><li>• Seqüencial</li><li>• Concomitante</li><li>• Transformativa</li></ul>

Fonte: Creswell (2010 p.36)

A estratégia de investigação utilizada no desenvolvimento deste trabalho foi a Pesquisa-Ação (PA). Os fatores que motivaram esta escolha estão diretamente relacionados ao fato de PA (i) permitir (e exigir) que a pesquisa não se limite a aspectos acadêmicos e burocráticos; (ii) prever a solução de um problema prático e (iii) a participação explícita do pesquisador no objeto investigado (HULT & LENNUNG, 1980; THIOLENT, 1996). Ademais, na área de desenvolvimento de software, a PA já é reconhecida pelos profissionais da área como uma estratégia de intervenção nas organizações (THIOLENT, 1983).

A PA é apresentada pela literatura como uma das alternativas para condução de pesquisas qualitativas (HULT & LENNUNG, 1980; MCKERNAN,1993; THIOLENT, 1997; CRESWELL;2010) quando se busca, concomitantemente, intervenção, elaboração de conhecimento científico e seu desenvolvimento. As intervenções diretas nas organizações oferecem aos pesquisadores oportunidades de acesso a informações com uma frequência não encontrada nos estudos e pesquisas tradicionais (THIOLENT, 1997).

De acordo com Thiollent (1996), PA é um tipo de pesquisa social, empírica, que acontece em paralelo com a resolução de um problema. Ou seja, espera-se de uma PA mais do que apenas resultados em termos de produção de conhecimento, uma vez que o pesquisador deve assumir papel de agente ativo na mudança do contexto pesquisado. O foco da PA está em conhecimentos de cunho prático, importando a relevância do trabalho em detrimento ao

suposto rigor científico da ciência positivista (COUGHLAN & COGHLAN, 2002). Ou seja, a PA permite cumprir o objetivo de geração de conhecimento científico enquanto se contribui para melhoria da compreensão dos problemas abordados e das soluções propostas (THIOLLENT, 1996).

## 2.4 Métodos de Pesquisa

O terceiro e último elemento a ser analisado pelos pesquisadores é o método específico, os quais estão relacionadas às formas de coleta, análise e interpretação dos dados que os pesquisadores propõem para realização do seu trabalho. A Tabela 2-4 apresenta os métodos discutidos por Creswell (2010).

Tabela 2-4: Métodos de Pesquisa

Métodos quantitativos	→ Métodos mistos	Métodos qualitativos
<ul style="list-style-type: none"> <li>• Predeterminado</li> <li>• Questões baseadas no instrumento</li> <li>• Dados de desempenho, dados de atitudes, dados observacionais e dados de censo</li> <li>• Análise estatística</li> <li>• Interpretação estatística</li> </ul>	<ul style="list-style-type: none"> <li>• Tanto métodos predeterminados quanto emergentes</li> <li>• Tanto questões abertas quanto fechadas</li> <li>• Formas múltiplas de dados baseados em todas as possibilidades</li> <li>• Análise estatística e de texto</li> <li>• Por meio da interpretação dos bancos de dados</li> </ul>	<ul style="list-style-type: none"> <li>• Métodos emergentes</li> <li>• Perguntas abertas</li> <li>• Dados de entrevistas, dados de observação, dados de documentos e dados audiovisuais</li> <li>• Análise de texto e imagem</li> <li>• Interpretação de temas e de padrões</li> </ul>

Fonte: Creswell (2010 p.40)

Pesquisas quantitativas possuem um protocolo de pesquisa relativamente padrão, o qual é definido antes das atividades de coleta e interpretação dos dados (BRYMAN 2004; CRESWELL, 2010). Não dá para se dizer o mesmo quando se trata de pesquisas qualitativas, os quais são norteadas por questões de pesquisa, a partir das quais se derivam objetivos, se constrói um quadro conceitual e se escolhem os métodos (de coleta e análise) e os mecanismos de validação. Todavia, a definição desses elementos se dá de forma não linear, mas interativa, de tal maneira que a definição de um elemento pode influenciar os demais e promover revisões e mudanças até mesmo nas questões que deram origem à investigação (MAXWELL, 2008; CRESWELL, 2010).

A análise conjunta das concepções, estratégia e os métodos de pesquisa levam a definição da abordagem mista para a condução desta investigação. Os pesquisadores se basearam tanto em análises quantitativas quanto qualitativas. As características quantitativas estão presentes no método de contagem de pontos de função previamente definido pelo IFPUG. No entanto, a aplicação do método apresenta certo grau de subjetividade, que remete às características de uma pesquisa qualitativa. As características da pesquisa qualitativa podem ser encontradas, ainda, na criação de uma proposta para mensuração de soluções de software com SOA.

## 2.5 ASPECTOS OPERACIONAIS

A Figura 2-2 apresenta resumidamente os procedimentos metodológicos envolvidos no desenvolvimento desta pesquisa. Em seguida, serão descritas cada uma das etapas apresentadas.

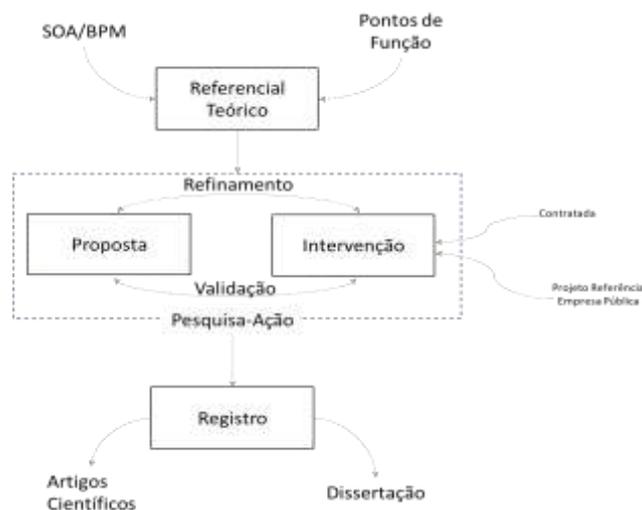


Figura 2-2: Procedimentos Metodológicos  
Fonte: Elaboração Própria

## 2.6 Construção do Referencial Teórico

O embasamento teórico necessário para o entendimento da problemática; a definição de objetivos e escopo; a formulação de propostas e premissas e a operacionalização da pesquisa foram construídas ao longo do tempo, através das disciplinas cursadas no curso de mestrado, da participação em projetos SOA, revisão bibliográfica e pela troca de experiência com o orientador e especialistas membros da equipe do projeto usado como pesquisa de campo.

A própria intervenção incluiu não apenas o desenvolvimento de um software, mas a construção de uma metodologia para desenvolvimento de projetos SOA, portanto exigiu um levantamento bibliográfico para construção de um referencial teórico.

Além disso, a experiência profissional do autor, em conjunto com a opinião especializada dos membros da equipe, contribuiu para a compilação do conhecimento sobre a realidade das abordagens SOA e APF, considerando tanto o contexto geral quanto particularidades deste trabalho e da pesquisa de campo. A extensa experiência do orientador, acadêmica e prática, contribui fortemente para a formar o entendimento sobre o problema e indicar os caminhos percorridos ao longo da pesquisa.

## **2.7 Pesquisa-Ação**

Devido à sua característica essencialmente empírica, o ponto de partida da pesquisa foi a construção de conhecimento científico a partir da contribuição para a solução de um problema prático ligado a uma realidade específica.

O objetivo foi expandir a compreensão da realidade e contribuir para solucionar um problema identificado; através de uma intervenção baseada na adaptação, proposição e aplicação de uma nova metodologia.

A definição de Hult & Lennung (1980) mostra o alinhamento entre a proposta de trabalho e a estratégia de investigação Pesquisa-Ação:

Pesquisa-Ação atende simultaneamente os objetivos da resolução de problemas práticos e expansão do conhecimento científico, como também desenvolve competência dos atores, sendo desenvolvido de forma colaborativa, numa situação particular, utilizando e alimentando os dados num processo crítico, tendo como alvo uma melhor compreensão da totalidade de uma determinada situação social, principalmente aplicável no entendimento dos processos de mudança dos sistemas sociais, realizada dentro de uma estrutura ética de trabalho, mutuamente aceitável. (HULT & LENNUNG, 1980: 1).

A PA envolveu a execução de duas etapas – Intervenção e Construção de uma Proposta – os quais serão descritas a seguir.

### *2.7.1.1 Intervenção*

A intervenção, ou pesquisa de campo, que deu sustentação à essa investigação ocorreu através da execução de um projeto real, cujo escopo foi a experimentação e adoção da

abordagem SOA por uma organização. Tal projeto foi desenvolvido através da contratação, por uma empresa pública, de uma consultoria especializada em SOA, da qual a pesquisadora era parte do time. Por razões de sigilo de informações, as empresas envolvidas no projeto serão doravante denominadas CONTRATANTE e CONTRATADA.

A escolha do contexto utilizado como pesquisa de campo ocorreu de forma natural, uma vez que a motivação do pesquisador para o tema dessa investigação surgiu exatamente durante a execução do projeto supracitado. Mais tarde, essa escolha se mostrou acertada, dada as próprias características da PA, apontadas por Coughlan & Coughlan (2002):

- Pesquisadores intervêm no contexto observado;
- Requer cooperação entre os pesquisadores e a equipe do cliente, bem como ajuste contínuo a novas informações e eventos;
- Requer um entendimento de padrões éticos, valores e normas relacionadas ao contexto particular de engajamento;
- Pesquisadores precisam de esforço para compreender previamente o ambiente corporativo, as condições de negócios, a estrutura e a dinâmica dos sistemas operativos, bem como as questões teóricas relacionadas a tais sistemas;
- Deve ser conduzida em tempo real

O papel principal assumido pelo pesquisador, ao longo da intervenção, foi de Gerente de Projeto. No entanto, o pesquisador desempenhou ainda, simultaneamente, atividades relacionadas aos perfis de Analista SOA, projetista e desenvolvedor de Banco de Dados. Desse fato decorrem dois aspectos principais que contribuiriam fortemente para a condução da pesquisa acadêmica: imersão no tema e acesso direto a equipe do projeto.

A imersão ocorreu de forma a permitir ao pesquisador controle sobre as atividades executadas, contato com especialistas no assunto e atuação direta na execução de atividades técnicas em diversos momentos durante a execução do projeto. O fato de o trabalho prático realizado pelo pesquisador estar fortemente alinhado às atividades desta pesquisa viabilizou intensa dedicação às atividades ligadas à preparação e à condução da investigação.

#### *2.7.1.2 Proposta e Registro*

A discrepância observada a partir de análises comparativas entre a estimativa do tamanho funcional utilizando-se PF (IFPUG, 2010) e a aferição do tamanho real do software com

SOA desenvolvido (vide seção 4.9) norteou a construção de uma proposta, a qual foi balizada pelo referencial teórico e gerada a partir de raciocínios abduativos<sup>1</sup>.

O processo de construção, validação e refinamento da proposta ocorreu de forma cíclica durante todo o processo de intervenção. De acordo com McKernan (1993), a pesquisa-ação possui caráter cíclico: as fases finais são usadas para aprimorar os resultados das fases anteriores.

Após finalizados os processos de intervenção, construção e validação da proposta, foi realizada uma análise crítica a partir dos resultados obtidos, os quais foram registrados nesta dissertação e em artigos científicos produzidos pela autora.

---

<sup>1</sup> Pinto & Santos (2012) sustentam que novos conhecimentos e teorias emergem principalmente a partir da interação entre abdução e indução, quando a coleta de fatos (indução) dá vazão a uma (ou mais) abdução (criação “imaginativa”, nova ideia, intuição ou palpite), que acaba se tornando um tipo de hipótese, que é então testada. O raciocínio abduativo requer criatividade, pois o pesquisador deve imaginar possibilidades teóricas para os dados observados e então formular e checar hipóteses, até que se chegue à mais plausível interpretação possível dos dados observados, considerando inclusive as surpresas, anomalias e mosaicos presentes nos dados coletados (CHARMAZ, 2008).

### 3 - FUNDAMENTAÇÃO TEORICA

#### 3.1 CONCEITOS E DEFINIÇÕES RELACIONADAS A PF

A Tabela 3-1 apresenta, de forma resumida, algumas das principais definições utilizadas no contexto da APF, mais particularmente, para o método desenvolvido pelo IFPUG, o qual será explorado nas seções seguintes. Esta lista não pretende ser exaustiva, maiores detalhes sobre conceitos ou definições serão descritas nas seções onde serão utilizadas buscando facilitar a leitura e entendimento.

Tabela 3-1: Conceitos APF

Termo	Descrição
Usuário	Para o IFPUG (2010), usuário é qualquer pessoa ou coisa que se comunica ou interage com o software a qualquer momento. Para COSMIC (2015), um usuário funcional é um (tipo de) usuário que é uma fonte e/ou um destino pretendido para os dados nos Requisitos Funcionais do Usuário de um pedaço de software.
Visão do Usuário	Requisitos Funcionais do Usuário, conforme percebidos pelo usuário. Ou seja, é uma descrição formal das necessidades de negócio do usuário na linguagem do próprio usuário (IFPUG, 2010).
Reconhecido pelo Usuário	Indica que os requisitos definidos para um determinado processo e/ou grupos de dados foram entendidos e aprovados tanto pelos usuários quanto pelos responsáveis pela construção do software (IFPUG, 2010). Por exemplo, usuários e analistas concordam e aprovam que a aplicação de Recursos Humanos a ser desenvolvida deve ser capaz de manter e guardar informações do funcionário dentro da própria aplicação (IFPUG, 2010).
Processo Elementar	Menor unidade de atividade que possui significado para o usuário. Um requisito funcional deve ser composto ou decomposto até a menor unidade de atividade cumprindo os itens a seguir: (i) é significativo para o usuário (ii) constitui uma transação completa (iii) é autocontida (iii) deixa o negócio da aplicação contada em um estado consistente. Exemplo: considerando o requisito para adicionar um funcionário, incluindo informações de salário e dependentes, somente será efetivamente registrado (criado) se todas as respectivas informações forem fornecidas. Para manter o requisito de negócio “incluir um funcionário” em um estado consistente, tanto o salário do empregado quanto as informações do (s) dependente(s) devem ser adequadamente incluídas. Incluir parcialmente as informações deixará o negócio “incluir um funcionário” em um estado inconsistente (IFPUG, 2010).

<b>Termo</b>	<b>Descrição</b>
Intenção primária	Representa a intenção que possui a maior importância (IFPUG, 2010).
Alterar o comportamento do sistema	Ocorre quando um parâmetro de negócio é alterado. A alteração deste parâmetro influencia o comportamento de outras transações. Por exemplo: "o comportamento do processo elementar de pagamento de funcionários é alterado quando uma mudança é feita para pagá-los às sextas-feiras, a cada duas semanas, ao invés de pagá-los no 15º dia e no último dia do mês, resultando em 26 períodos de pagamento por ano, contra 24." (IFPUG, 2010 p.7-7).
Lógica de Processamento	Quaisquer requisitos especificamente solicitados pelo usuário para concluir um processo elementar, como por exemplo: validações; algoritmos ou cálculos e leitura ou manutenção de uma função de dados (IFPUG, 2010).
Informações de Controle	Dados capazes de influenciar um processo elementar. Especificam quais dados que devem ser processados, quando e como. Exemplo: ciclos de pagamento são estabelecidos pelo responsável do departamento de folha de pagamento com o objetivo de definir quando os funcionários de cada local serão pagos. O ciclo que pagamento contém as informações de periodicidade que especificam quando o processo elementar de pagamento de funcionários ocorrerá (IFPUG, 2010).
Requisito Funcional	Representa um subconjunto dos requisitos do usuário que especifica o que o software será capaz de realizar (ISO/IEC, 2007). Os requisitos funcionais do usuário incluem transferência de dados (por exemplo: dados de entrada de cliente), transformação de dados (por exemplo, calcular juros bancários) e armazenamento de dados (por exemplo, armazenamento de pedido de cliente) (IFPUG, 2010).
Requisito Não-Funcional	De acordo com IFPUG (2010), a ISO não descreve uma definição formal para requisito não-funcional, porém cita exemplos como (i) restrições de qualidade (usabilidade, confiabilidade, eficiência e portabilidade) (ii) restrições organizacionais (por exemplo: locais para operação, hardware-alvo e conformidade com os padrões) (iii) restrições ambientais (por exemplo: interoperabilidade, segurança e privacidade) e (iv) restrições de implementação. Apesar de o termo já ser utilizado a mais de duas décadas, ainda não existe um consenso sobre sua natureza, procedimentos para documentação e validação (GLINZ, 2007).
Mantido	Refere-se à capacidade que um processo elementar possui para incluir, alterar ou excluir dados (IFPUG, 2010).

<b>Termo</b>	<b>Descrição</b>
Autocontido	Nenhum passo anterior ou subsequente do processamento é necessário para iniciar ou completar o Requisito Funcional do Usuário (IFPUG, 2010)
Estado consistente	Ponto no qual o processamento foi totalmente executado; o Requisito Funcional do Usuário foi satisfeito e não há nada mais a ser feito (IFPUG, 2010).
Tamanho do Software	O tamanho funcional é definido como “tamanho do software derivado pela quantificação dos requisitos funcionais do usuário” (DEKKERS, 2003).
Contagem Indicativa	A contagem indicativa baseia-se somente em informações a respeito das funções de dados, ou seja, na quantidade de arquivos lógicos existentes (ALI e AIE) (NESMA, 2015).
Contagem Estimada	A contagem estimada considera, além da quantidade de funções de dados, informações a respeito das funções de transação, de forma que os requisitos dos usuários precisam estar mais detalhados (NESMA, 2015)..
Contagem Detalhada	Para a contagem detalhada, além da quantidade de funções transacionais, é necessário determinar a complexidade funcional (Baixa, Média, Alta) de cada função individualmente (NESMA, 2015).

Fonte: Elaboração Própria

## 3.2 ANÁLISE DE PONTOS DE FUNÇÃO - IFPUG

### 3.3 Introdução

O *International Function Point Users Group* (IFPUG), fundado em 1986, é uma organização sem fins lucrativos que apoia dois diferentes tipos de metodologia para dimensionamento do tamanho do software:

- A primeira delas é definida no *Manual de práticas de Contagem de Pontos de Função* (CPM) (IFPUG, 2010) e tem por objetivo o dimensionamento do tamanho funcional do software.
- A segunda tem por objeto a medição do tamanho não-funcional, através da aplicação do método denominado *Software Non-functional Assessment Process* (SNAP) e descrito no *Manual de Práticas de Avaliação* (APM) (IFPUG, 2015).

A primeira versão pública do APM SNAP foi publicada recentemente em setembro de 2011, e definiu os conceitos, processos e regras para a avaliação dos requisitos não-funcionais de software. Alguns estudos já foram realizados no sentido de validar a utilização da técnica, entre eles Tichenor (2014).

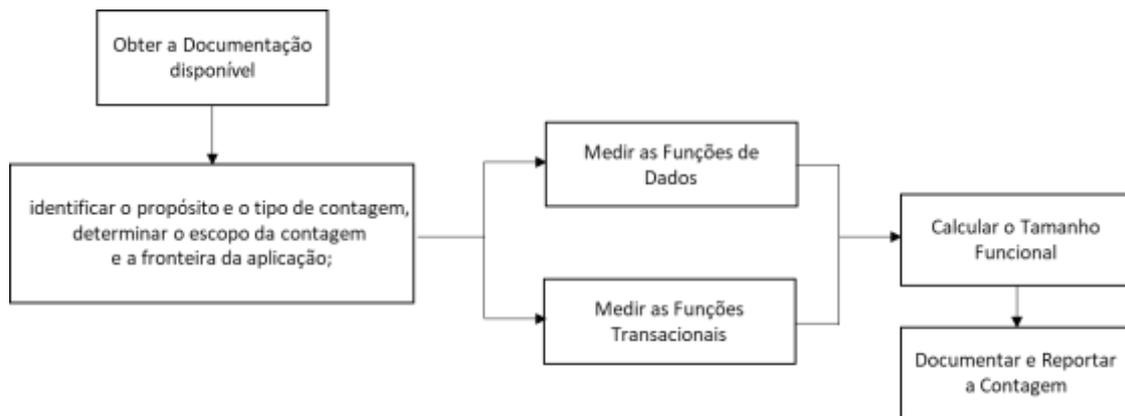
Já o método para dimensionamento do tamanho funcional do software, o qual foi derivado do método de Allan Albrecht e desenvolvido em conformidade com a ISO/IEC 14143-1:2007, teve a *release 2.0* publicada em abril de 1988 e encontra-se atualmente na versão 4.3.1, publicada em janeiro de 2010.

As organizações podem aplicar esse padrão internacional para medir o tamanho de um software com o intuito de: (i) estimar esforço, custo e prazo requeridos para o desenvolvimento, manutenção e melhoria do software; (ii) fornecer suporte à análise de qualidade e produtividade; (iii) fornecer um fator de normalização para a comparação de softwares (IFPUG, 2010). Sua utilização apresenta diversos benefícios dos quais destacam-se: regras de contagem objetivas, independência da solução tecnológica e linguagem de programação e possibilidade de geração de estimativa já nas fases iniciais do ciclo de vida do software (SISP, 2015).

Na seção seguinte serão descritas, de maneira resumida, as regras para aplicação da APF segundo o CPM.

### **3.4 Etapas APF**

A medição do tamanho do software realizada através da aplicação do método APF envolve a execução das etapas apresentadas na Figura 3-1 e descritas a seguir (IFPUG, 2010):



Fonte: Adaptado de IFPUG (2010) p.9  
 Figura 3-1: Etapas APF IFPUG

### 3.5 Obter a Documentação Disponível

A medição do tamanho funcional através da aplicação das regras do CPM tem início com identificação dos requisitos funcionais, através da análise da documentação do software sendo medido (SISP, 2015). A documentação necessária para aplicação das regras deve conter informações detalhadas sobre as funcionalidades entregues ou impactadas pelo software a ser medido. Em outras palavras, esta etapa tem por objetivo, a identificação dos requisitos funcionais do usuário (IFPUG, 2010). São exemplos de artefatos frequentemente utilizados, o documento de visão do projeto, diagrama de dados e manuais de usuário (IFPUG, 2010). Quando a documentação obtida não for suficiente para determinar a medição, especialistas no assunto podem ser consultados para suprir as falhas da documentação (IFPUG, 2010).

### 3.6 Identificar Propósito, Tipo da Contagem, Escopo e Fronteira da Aplicação

Após concluída a seleção e análise da documentação do projeto, o próximo passo é a definição do propósito, o qual determina razão para execução da contagem de pontos de função. Com base no propósito, é identificado o tipo da contagem, que pode ser: (i) Contagem de Desenvolvimento: (i) medida da funcionalidade fornecida aos usuários considerando a primeira instalação do software. Este tipo de contagem inclui funções de conversão (utilizadas para converter dados e/ou prover outros requisitos de conversão especificados pelo usuário) (ii) Melhoria/Manutenção: refere-se à modificações realizadas em um software já existente, o seja, funcionalidade incluída, alterada ou excluída. (iii) Aplicação: uma Aplicação é conjunto coeso de procedimentos automatizados e dados, que tem por objetivo suportar um objetivo de negócio e é composta por um ou mais componentes,

módulos ou subsistemas. Este tipo de contagem não inclui as funções de conversão de dados (IFPUG, 2010).

O escopo da contagem define as funcionalidades de software que serão incluídas na medição do tamanho do software.

A fronteira da aplicação é uma interface conceitual que separa logicamente a aplicação que está sendo medida e o mundo externo (usuário e outras aplicações que interagem com a aplicação sendo medida). É definida com base apenas na visão do usuário; aspectos técnicos não são considerados (IFPUG, 2010).

### **3.7 Medir Funções de Dados**

Definidos o Propósito, Tipo de Contagem, Escopo e Fronteira da Aplicação, o próximo passo para a medição do tamanho funcional do software é a medição das funções de dados (IFPUG, 2010). Para esta etapa são realizados passos descritos a seguir.

#### **I. Identificar as Funções de Dados**

As funções de dados representam os requisitos funcionais do usuário no que se refere a armazenamento e/ou recuperação de dados. Toda e qualquer funcionalidade de dados pertencente ao escopo da contagem deve ser avaliada para identificação dos grupos lógicos associados, que é um grupo de dados conforme visto pelo usuário e composto por uma ou mais entidade de dados. Cada grupo lógico de dados corresponde a uma Função de Dados na APF, que será classificada, de acordo com o seu propósito, como Arquivo Lógico Interno (ALI) ou Arquivo de Interface Externa (AIE) (IFPUG, 2010).

#### **II. Classificar cada função de dado como um ALI ou AIE**

As Funções de Dados são classificadas em Arquivos Lógicos Internos (ALI) e Arquivos de Interface Externa (AIE) (IFPUG, 2010):

- Arquivo Lógico Interno (ALI) é um grupo de dados logicamente relacionados, reconhecido pelo usuário e mantidos dentro da fronteira da aplicação por meio de um processo elementar da aplicação que está sendo medida (IFPUG, 2010).
- Arquivo de Interface Externa (AIE) é um grupo de dados ou informação de controle logicamente relacionados, reconhecido pelo usuário, mantido por meio de um

processo elementar pertencente a uma aplicação externa (outra aplicação) e referenciado pela aplicação sendo medida para fins de recuperação de dados. Ou seja, os dados são mantidos dentro da fronteira de outra aplicação. (IFPUG, 2010).

A principal diferença entre um ALI e um AIE está relacionada ao responsável pela sua manutenção. Um ALI é mantido pela aplicação sendo medida enquanto um AIE é mantido por outra aplicação e apenas referenciado pela aplicação que está sendo medida (IFPUG, 2010).

### III. Determinar a complexidade dos ALI ou AIE e sua contribuição para o tamanho funcional.

A complexidade de um ALI ou AIE é determinada com base na quantidade de Dados Elementares Referenciados (DER) e Registro Lógico de Dados (RLR), de acordo com matriz apresentada na Tabela 3-2. O DER é um campo único, reconhecido pelo usuário, e que atravessa a fronteira da transação, sendo exibido para o usuário. O RLR representa um subgrupo de dados elementares reconhecido pelo usuário dentro de uma Função de Dados e são normalmente derivados dos modelos de dados, preferencialmente o modelo lógico de banco de dados (IFPUG, 2010).

Tabela 3-2: Complexidade das Funções de Dados

		DER		
		1-19	20-50	>50
RLR	1	Baixa	Baixa	Media
	2-5	Baixa	Média	Alta
	>5	Media	Alta	Alta

Fonte: Adaptado de Manual CPM p.6-8

A partir da complexidade funcional determina-se, de acordo os dados apresentados na Tabela 3-3, o tamanho em PF de cada ALI e AIE. A contribuição em PF de cada ALI e AIE para o tamanho funcional do software pode ser totalizada<sup>2</sup>.

---

<sup>2</sup> *Maiores detalhes sobre as regras para medição das funções de dados podem ser encontradas em IFPUG (2010).*

Tabela 3-3: Tamanho das Funções de Dados

		Tipo	
		ALI	AIE
Complexidade Funcional	Baixa	7	5
	Média	10	7
	Alta	15	10

Fonte: Adaptado de IFPUG, 2010 p.6-8

### 3.8 Medir as Funções de Transação

As funções de transação representam os processos elementares responsáveis por fornecer funcionalidades ao usuário para processamento de dados e são classificadas como Entrada Externa (EE), Saída Externa (SE) ou Consulta Externa (CE) (IFPUG, 2010). A seguir são descritos os passos necessários para identificação e classificação das Funções de Transação.

#### I. Identificar as Funções de Transação (Processos Elementares)

Para a identificar as Funções Transacionais, deve-se identificar os processos elementares envolvidos no software sendo medido. Para isso, um requisito funcional deve ser composto ou decomposto até a menor unidade de atividade cumprindo os itens a seguir: (i) é significativo para o usuário (ii) constitui uma transação completa (iii) é autocontida (iii) deixa o negócio da aplicação contada em um estado consistente (IFPUG, 2010).

#### II. Determinar o processo elementar único.

Nesta etapa, deve-se verificar se não existe nenhuma outra transação com os mesmos tipos de dados, arquivos referenciados, e lógica de processamento que já tenha sido contada, uma vez que não é permitida a contagem de transações com as mesmas características (IFPUG, 2010; VAZQUEZ, 2013).

#### III. Classificar cada função de transação como Entrada Externa (EE), Saída Externa (SE) ou Consulta Externa (CE).

A intenção primária do processo elementar é o guia para a classificação da Função de Transação em Entrada Externa (EE), Saída Externa (SE) ou Consulta Externa (CE). Para a identificação da intenção primária deve-se analisar se o mesmo altera o comportamento do sistema, mantém um mais ALI e apresenta informações ao usuário.

Uma EE é um processo elementar responsável pelo processamento de dados ou informações de controle que tem sua origem fora da fronteira da aplicação. Sua intenção primária é manutenção de um ou mais ALI e/ou alteração do comportamento do sistema.

Uma CE é um processo elementar responsável pelo envio de dados ou informações de controle que saem para fora da fronteira da aplicação. Sua intenção primária é a apresentação de informações ao usuário por meio de uma simples recuperação de dados ou informações de controle vindas de um ALI ou AIE. O processamento de uma CE não inclui cálculos, fórmulas matemáticas ou derivação de dados. Durante o processamento de uma CE, ALI não são mantidos e comportamento do sistema não é alterado.

Uma SE é um processo elementar responsável pelo envio de dados ou informações de controle que saem para fora da fronteira da aplicação, incluindo lógica de processamento adicional, além do identificado em uma CE. Sua intenção primária é a apresentação de dados ao usuário com inclusão de processamento adicional, além da simples recuperação de dados ou informação de controle. A lógica de processamento de uma SE deve conter, necessariamente, um cálculo ou fórmula matemática e/ou criar dados, e/ou manter um ou mais ALI e/ou alterar o comportamento do sistema (IFPUG, 2010).

A Tabela 3-4 apresenta, resumidamente, as funções possíveis de serem executadas por cada uma das funções de transação (IFPUG, 2010).

Tabela 3-4: Relação entre a Intenção Primária e o Tipo de Função de Transação

Função	Tipo de Função de Transação		
	EE	SE	CE
Alterar o comportamento do sistema	IP	F	N/A
Manter um ou mais ALIs	IP	F	N/A
Apresentar a informação ao usuário	F	IP	IP

**Legenda:**

<b>IP</b>	Intenção primária do tipo de função de transação
<b>F</b>	É uma função do tipo Função de Transação, mas não é a intenção primária e está presente algumas vezes.
<b>N/A</b>	A função não é permitida para o tipo de Função de Transação

Fonte: Adaptado de IFPUG, 2010 p. 1-24

**IV. Determinar a complexidade funcional para cada Função de Transação e sua contribuição para o tamanho funcional.**

A complexidade das funções de transação é determinada a partir das quantidades de RLR e DER, considerando as matrizes das Tabela 3-5 e Tabela 3-6. Um RLR pode ser tanto um ALI lido ou mantido quanto um AIE lido pela função de transação.

Tabela 3-5: Complexidade Funcional das EE

		DER		
		1-4	5-15	>15
RLR	0 - 1	Baixa	Baixa	Média
	2	Baixa	Média	Alta
	> 2	Média	Alta	Alta

Fonte: Adaptado de IFPUG, 2010 – p. 7-19

Tabela 3-6: Complexidade Funcional das SE e CE

				DER		
				1-5	6-19	>19
RLR			0 - 1	Baixa	Baixa	Média
			2-3	Baixa	Média	Alta
			> 3	Média	Alta	Alta

Fonte: Adaptado de IFPUG, 2010 – p. 7-19

A determinação do tamanho funcional de cada Função de Transação é realizada de acordo com matriz apresentada na Tabela 3-7.<sup>3</sup>

Tabela 3-7: Tamanho Funcional da Função de Transação

		Tipo		
		EE	SE	CE
Complexidade Funcional	Baixa	3	4	3
	Média	4	5	4
	Alta	6	7	6

Fonte: Adaptado de IFPUG, 2010 p.7.20

### 3.9 Calcular o Tamanho Funcional

O tamanho funcional é o tamanho do software obtido através da quantificação dos requisitos funcionais do usuário, ou seja, através da medição das funções de dados e transação, ou seja, a medição das funções de dados e de transação resulta no tamanho funcional do software. (IFPUG, 2010).<sup>4</sup>

### 3.10 Documentar e Reportar a Contagem

A documentação da contagem de pontos de função inclui, entre outras informações, o propósito, tipo e escopo da contagem; fronteira da aplicação; data de realização da contagem; uma listagem contendo todas as Funções de Dados e Funções de Transação identificadas, incluindo detalhamento como o tipo, complexidade, quantidade de DER e RLR ou ALR envolvidos; documentação de origem utilizada como base para a contagem; uma referência

<sup>3</sup> Maiores detalhes sobre as regras para medição das funções de transação podem ser encontradas no CPM (IFPUG, 2010).

<sup>4</sup> Maiores detalhes sobre o cálculo do tamanho funcional podem ser encontrados no CPM (IFPUG, 2010).

cruzada entre as funções de dados e funções de transação; quaisquer premissas assumidas e questões resolvidas e um sumário com o resultado da contagem (IFPUG, 2010).

O nível de detalhamento da contagem deve ser previamente acordado com o cliente. Um alto nível de detalhamento da contagem resulta em maior facilidade para a rastreabilidade, usabilidade e manutenibilidade. No entanto, se o cliente está interessado apenas no resultado, este detalhamento pode ser desnecessário (IFPUG, 2010). Finalmente, o resultado da contagem deve ser consistentemente reportado de forma a permitir ao leitor a fácil identificação do padrão com o qual os mesmos mantem conformidade (IFPUG, 2010).

### **3.11 ARQUITETURA ORIENTADA A SERVIÇO**

Apesar de já existir há algum tempo, SOA ganhou relevância apenas a partir de meados dos anos 2000 (ERL, 2005). Segundo pesquisa realizada recentemente (WINTERGREEN, 2014), o mercado de SOA está crescendo rapidamente. O relatório da pesquisa indica que o mercado SOA atingiu US \$ 5,7 bilhões em 2013 e pode chegar a US \$ 16,4 bilhões até 2020. De acordo com a pesquisa, este crescimento significativo se deve ao fato de SOA oferecer processos automatizados mais eficientes e proporcionar à TI a possibilidade de investir uma maior parte do orçamento no crescimento do negócio.

A literatura apresenta diferentes e variadas definições para SOA. Para melhor compreendê-las é preciso, antes, entender o significado do termo serviço, o qual é amplamente utilizado nas definições e implementações de SOA.

Para Erl (2005), serviço é a unidade fundamental da lógica Orientada a Serviço. Uma unidade de lógica é classificada como um serviço quando a Orientação a Serviço é aplicada em uma extensão significativa, ou seja, os princípios SOA (descritos na seção 3.12) são aplicados em uma extensão significativa (ERL, 2005). Para Arsanjani (2008), do ponto de vista de negócios, um serviço é uma funcionalidade bem definida, encapsulada, reutilizável e alinhada com os objetivos de negócios. Para Endrei (2004), os serviços mapeiam as funções de negócios identificadas durante a análise dos processos de negócios. Cada serviço tem uma interface bem definida que permite sua publicação, descoberta e invocação por seus consumidores.

Posto isso, são apresentadas algumas das definições e propostas de SOA encontradas na literatura.

- Para Erl (2005) SOA é um paradigma de projeto onde a lógica de automação é decomposta em unidades menores individualmente desenhadas (serviço). Quando utilizadas coletivamente, estas unidades representam uma parte significativa da automação do negócio.
- Para Arsanjani (2007), SOA oferece para as empresas, uma arquitetura flexível que permite o alinhamento entre os objetivos de negócio e TI, além de entregar soluções desacopladas e reutilizáveis. Com uma arquitetura flexível e extensível, SOA reduz custo, aumenta a receita e fornece agilidade na entrega de novas aplicações, bem como facilita a integração com aplicações ou sistemas legados.
- Para Bieberstein (2008), SOA é um framework para integração de processos de negócios e suporte à infraestrutura de TI, na forma de componentes (serviços) seguros e padronizados que podem ser combinados para atender as necessidades de evolução do negócio.
- Para Josuttis (2007), SOA representa uma abordagem onde os recursos de TI são utilizados visando apoiar o negócio da organização. A utilização desta abordagem visa, ainda, apoiar o processo de alinhamento estratégico da organização.
- Para Endrei (2004), SOA apresenta uma abordagem para construção de sistemas distribuídos que oferecem funcionalidades como serviços para serem consumidos diretamente pelo usuário final ou por outros serviços.
- De acordo com o Gartner (2005), SOA é uma abordagem arquitetural corporativa que permite a criação de serviços de negócio interoperáveis que podem facilmente ser reutilizados e compartilhados entre aplicações e empresas.

### **3.12 Princípios SOA**

De acordo com Erl (2008) são oito os princípios que norteiam a Orientação a Serviço, sendo que o Baixo Acoplamento de Serviço, Abstração de Serviço e Capacidade de Composição de Serviço agem como reguladores dos demais (ERL, 2008), conforme ilustra a Figura 3-2. Cada um dos princípios será descrito nas seções seguintes.

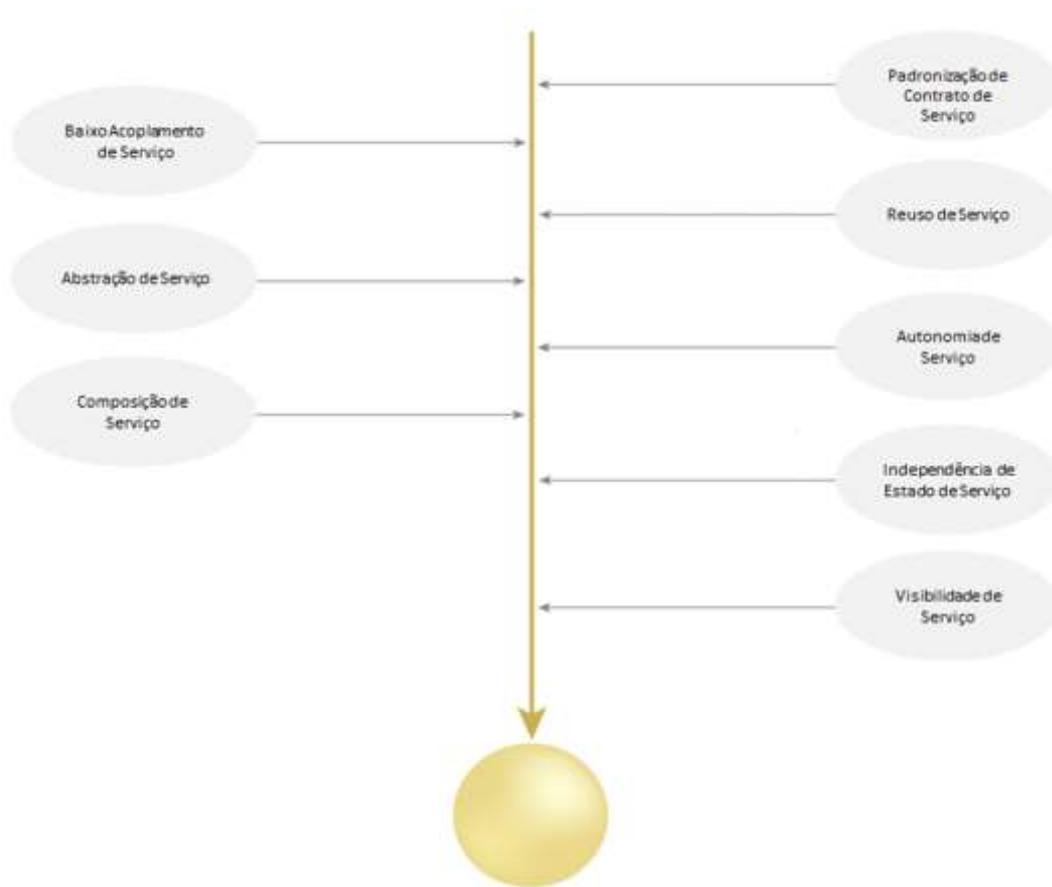


Figura 3-2: Princípios SOA  
 Fonte: Traduzido de Erl (2008 p.113)

a) Padronização de Contrato de Serviço

Um contrato de serviço é uma interface técnica que expressa meta-informação sobre um serviço e estabelece os padrões para invocação e interação com o serviço. É através do contrato de serviço que são expostos o propósito e as capacidades de um serviço. SOA implica em grande ênfase no projeto de contrato de serviços que preconiza o padrão de projeto “contrato primeiro”, ou seja, os contratos são construídos de maneira padronizada e independentes de linguagem de programação e com uma representação canônica das mensagens. Para acessar e interagir com um serviço, o consumidor deve conformar com os padrões estabelecidos no contrato do serviço.

b) Baixo Acoplamento de Serviços

Acoplamento refere-se a uma conexão entre duas coisas e sua extensão é determinada pelo nível de dependência entre elas. No contexto de SOA, o baixo acoplamento é alcançado através da utilização do contrato de serviço que permite os serviços interagirem a partir de parâmetros pré-definidos. De maneira prática, único acoplamento desejável (positivo) ocorre

entre a lógica do serviço e contrato de serviço (Lógica-para-Contrato). Qualquer outro tipo de acoplamento é indesejável, tendo destaque o acoplamento entre o consumidor e a implementação (consumidor-para-Implementação), ou seja, o consumidor ignora o contrato do serviço e tem acesso direto a lógica do serviço.

#### c) Abstração de Serviço

Este princípio está relacionado com a necessidade de ocultar, tanto quanto possível, os detalhes subjacentes de um serviço. A informação sobre o serviço limita-se ao que está publicado no contrato de serviço, ou seja, estão disponíveis apenas as informações essenciais para utilização do serviço (ERL, 2005). O propósito dessas limitações é aumentar a longevidade do contrato de serviço e permitir a evolução da implementação subjacente pelos proprietários de serviço.

#### d) Capacidade de Reuso de Serviço

Quando um serviço encapsula uma lógica que é útil para mais que um consumidor, ele pode ser considerado reutilizável. SOA tem como objetivo primário a construção de serviços, em sua maioria, reutilizáveis. Ou seja, SOA tem por objetivo a construção de serviços com um contexto funcional agnóstico, ou seja, serviços com propósito genérico, ou multi-propósito, que são posicionados como recursos corporativos reutilizáveis.

#### e) Autonomia de Serviço

Refere-se ao nível de controle que se tem sobre um serviço em tempo de projeto e em tempo de execução. A aplicação deste princípio busca minimizar a quantidade de recursos compartilhados dos quais um serviço é dependente. A baixa autonomia pode comprometer a previsibilidade do desempenho e comportamento do serviço.

#### f) Independência de Estado de Serviço

Estado refere-se à condição geral de alguma coisa. Por exemplo, um carro em movimento representa um estado enquanto o carro sem movimento representa um outro estado. No contexto de software, significa que a aplicação pode transitar entre diferentes estados em *runtime* (tempo de execução). O que faz que o estado seja normalmente temporário por natureza. No contexto de SOA, o princípio da independência de estado enfatiza a necessidade dos serviços de reduzir ou mesmo eliminar o consumo de recursos de sistemas devido ao processamento de gerenciamento de estado desnecessário. Desta forma,

maximiza-se a escalabilidade, especialmente quando se trata de serviços agnósticos, os quais são projetados para reuso e composição.

g) Visibilidade de Serviço (Capacidade de Descoberta)

É desejável que serviços, especialmente os agnósticos, possam ser facilmente localizados e compreendidos (deve ser possível identificar seu propósito e capacidades) de maneira a propiciar o seu reuso. Isso pode ser alcançado através da disponibilização de meta informação (ou metadados), ou seja, dados sobre os dados.

h) Capacidade de Composição de Serviço

Composição de serviços é a criação de novo serviço a partir da combinação de outros serviços visando a automatização de uma tarefa ou processo de negócio em particular. Importante ressaltar que os demais princípios existem principalmente para suportar o princípio de composição de serviço. Ou seja, serviços são projetados para se tornarem membros efetivos de composição.

### **3.13 Objetivos Estratégicos SOA**

ERL (2008) apresenta os sete objetivos estratégicos (longo prazo) que fornecem o alicerce para a correta adoção de SOA. Conforme ilustra a Figura 3-3, os quatro primeiros objetivos levam a obtenção dos três últimos, que podem ser considerados benefícios de SOA. Esta natureza estratégica representa uma característica própria de SOA, que contrasta com a natureza tática (curto prazo) da abordagem tradicional de desenvolvimento de software. A seguir são descritos cada um dos objetivos estratégicos de SOA.

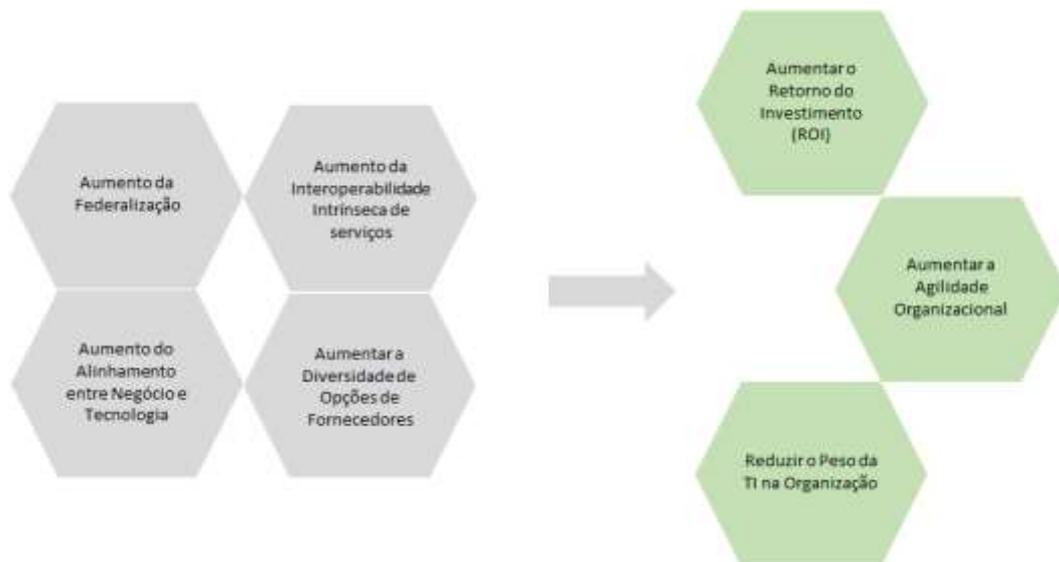


Figura 3-3: Objetivos Estratégicos de SOA  
 Fonte: Adaptado de Erl (2008 p. 56)

a) Aumento do Alinhamento entre Negócio e Tecnologia

O alinhamento entre o Negócio e Tecnologia é representado pela tradução das metas de negócio em metas de TI. Isso é alcançado quando especialistas da área de negócio e tecnologia trabalham de maneira colaborativa durante as fases de análise e modelagem do negócio.

b) Aumento da Federalização

Federalização é a unificação de ambientes (recursos e aplicativos) díspares que mantêm sua autonomia e governança independentes. De maneira prática, a partir da federalização o acesso aos serviços é padronizado de forma a unificar a visão dos seus consumidores (ERL, 2005). Isso é realizado através da implementação de serviços padronizados (contrato de serviço) sendo que cada serviço encapsula um sistema diferente.

c) Aumento da Interoperabilidade Intrínseca de Serviços

Interoperabilidade intrínseca representa a habilidade do software para interagir e trocar dados de maneira nativa, enquanto a integração representa o esforço requerido para atingir a interoperabilidade entre programas de software. SOA tem como objetivo a construção de serviços interoperáveis, ou seja, serviços são projetados para serem compatíveis de modo a reduzir a necessidade de integração.

d) Aumentar a Diversidade de Opções de Fornecedores

A diversificação de fornecedores representa uma alternativa para que apenas parte de uma organização possa ser reconstruída ou estendida com tecnologias pertencentes a um novo fornecedor. SOA pode ser projetado de forma neutra a plataformas proprietárias. O contrato de serviço permite posicionar serviços como terminais padronizados, de maneira a abstrair detalhes de implementações proprietárias (ERL, 2008).

e) Aumentar o Retorno do Investimento (ROI)

A Redução de Custos é medida em termos do aumento do retorno do investimento (ROI) ou da redução do custo total de propriedade (TCO). SOA preconiza a construção de lógica agnóstica, o qual tende a ser reutilizável e interoperável para promover aumento do ROI ao longo do tempo (ERL, 2008).

f) Aumentar a Agilidade Organizacional

O Aumento da Agilidade é medido em termos da redução do tempo médio para o desenvolvimento ou evolução de soluções ou da redução do tempo de entrega de uma solução. Serviços agnósticos, por serem reutilizáveis, tornam-se ativos de TI, que podem ser repetidamente compostos e recompostos em diferentes soluções.

g) Reduzir o Peso da TI na Organização

A Otimização do Ambiente de TI é medida em termos da redução do número de sistemas em operação existentes e do número de sistemas redundantes eliminados. A aplicação de SOA de maneira consistente resulta em uma TI corporativa onde a redução de redundâncias, desperdícios, custo operacional e *overhead* associado com sua governança e evolução são drasticamente reduzidos. Os benefícios são percebidos através do aumento significativo da capacidade de resposta e da efetividade dos custos.

### **3.14 Modelos de Serviço**

Modelo de serviços são camadas conceituais que estabelecem um contexto funcional base para os serviços e permite identificar o propósito e função geral do mesmo. São estabelecidos com base no tipo de lógica encapsulada pelo serviço (negócio ou utilitária) e no potencial de reuso desta lógica (agnóstica ou não agnóstica) (ERL, 2008). Dentro deste contexto, deve-se considerar (ERL, 2008):

## I. Lógica de negócio vs. Lógica utilitária.

A lógica é classificada como lógica de negócio quando ela é derivada de modelos de análise e especificações de negócio, como por exemplo, definições de processos de negócio (especificações), documento de regras de negócio e modelos lógicos de dados. Já partes da lógica de processamento que não é relacionada ou derivada do modelo de negócio é classificada como lógica utilitária. Esse tipo de lógica está relacionado a diversas funções de processamento disponíveis como recursos tecnológicos que apoiam a construção de soluções automatizadas.

## II. Lógica agnóstica vs. Lógica não agnóstica.

Lógica agnóstica está relacionada ao desenvolvimento e execução de serviços reutilizáveis. Uma lógica agnóstica é multi-propósito, não atendendo os objetivos de uma tarefa em particular, e está relacionada ao desenvolvimento e execução de serviços reutilizáveis. Por outro lado, uma lógica não-agnóstica é específica e atende ao propósito único de uma tarefa em particular.

São quatro os modelos de serviços propostos por Erl (2008): serviços de processo, serviços de tarefa, serviços de entidade e serviços utilitários, conforme ilustra a Figura 3-4.

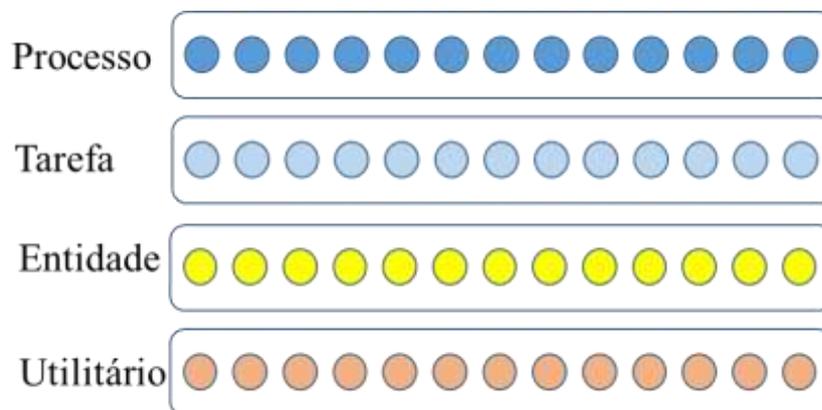


Figura 3-4: Modelo de Serviços  
Fonte: Adaptado de Erl (2008, p. 43)

## **I. Camada de Serviços de Processo**

Esta camada é composta por serviços que implementam processos de negócio, tipicamente definidos com uma combinação de tarefas interativas (de usuário) e de serviço (automáticas). De natureza intencionalmente não agnóstica possuem potencial de reuso limitado ou inexistente.

## **II. Camada de Serviços de Tarefa**

Esta camada é composta por uma coleção de serviços, denominados serviços de tarefa, os quais encapsulam lógica específica de uma tarefa de negócio. De natureza intencionalmente não agnóstica possuem potencial de reuso limitado ou inexistente. Normalmente, expõem um número reduzido de capacidades (método) (i.e. apenas uma).

## **III. Serviço de Entidade**

Esta camada é composta por uma coleção de serviços, denominados serviços de entidade, cujo contexto é derivado de uma entidade de negócio específica (p.e. Cliente, produto, etc.) ou de um grupo de entidades de negócio relacionadas. Capacidades de serviços de entidade proveem funcionalidades centradas em torno do processamento do conjunto de informações associado com a entidade de negócio. Isso geralmente resulta na definição de capacidades de CRUD, termo utilizado para designar as quatro operações básica de banco de dados: *Create* (inserir), *Read* (consultar), *update* (atualizar), *delete* (excluir).

## **IV. Serviços Utilitários**

Esta camada é composta por um conjunto de serviços, denominados serviços utilitários, os quais são responsáveis por prover funcionalidades que endereçam interesses múltiplos, enquanto mantêm um escopo funcional não-negocial. Frequentemente, encapsulam recursos corporativos, tais como sistemas legados e bases de dados e os expõem em contextos únicos.

### **3.15 Modelos Canônicos de Dados**

Modelo Canônico de Dados refere-se a esquemas/tipos globalmente padronizados para representação de informações sobre as entidades de negócio. Sua utilização tem como principal consequência a reutilização e padronização de mensagens na comunicação entre serviços (ERL, 2008). Para um melhor entendimento, pode se fazer uma analogia do modelo

canônico de dados e modelo conceitual de dados da abordagem tradicional de software. A Figura 3-5 apresenta um modelo de entidades do qual foi derivado o modelo canônico de dados do PROJETO DE REFERÊNCIA.

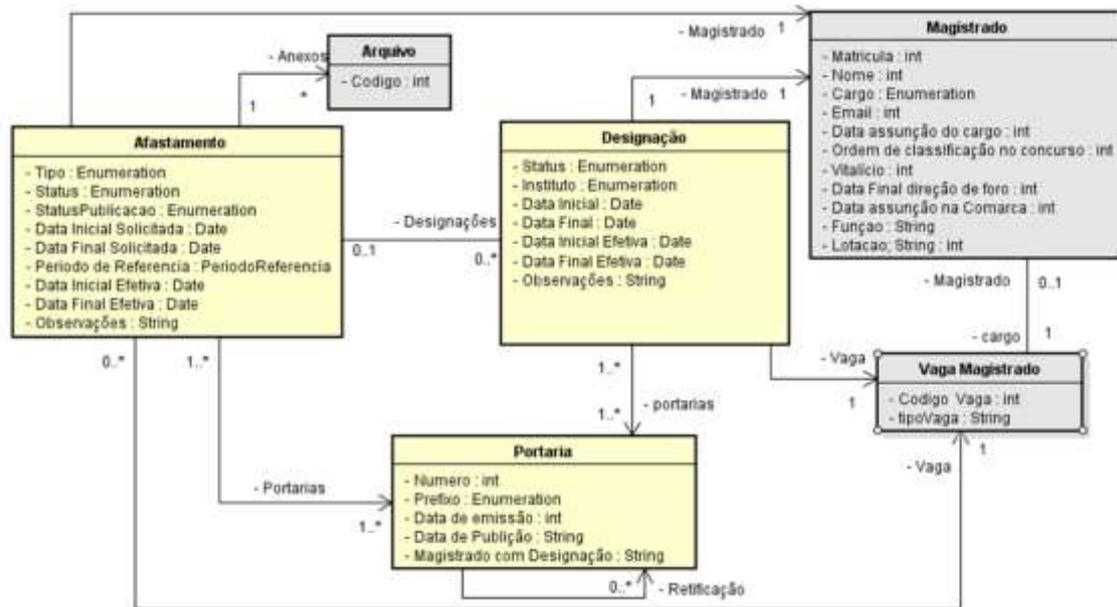


Figura 3-5: Modelo de Entidades/Canônico  
Fonte: PROJETO DE REFERÊNCIA

## 4 - INTERVENÇÃO

### 4.1 INTRODUÇÃO

O escopo de trabalho no qual a pesquisa de campo esteve inserida incluía estruturação, desenvolvimento, implantação e monitoramento do programa de adoção SOA; atividades de diagnóstico situacional, definição e desenvolvimento de metodologia, modelos arquiteturais, diretrizes, políticas, padronizações, processos e práticas corporativas associadas à adoção de SOA; e, ainda, a criação de um sistema desenvolvido integralmente utilizando SOA, o qual será denominado PROJETO DE REFERÊNCIA.

De todo esse escopo, o PROJETO DE REFERÊNCIA é a parte do contrato que mantém relação estreita com essa investigação e, em vários momentos, se confunde com a intervenção realizada durante a pesquisa. Durante a execução do PROJETO DE REFERÊNCIA, foram realizadas as atividades descritas a seguir:

- I. Aplicação da APF através das regras puras definidas no CPM (IFPUG, 2010) para medição do tamanho funcional. Para garantir a qualidade, confiabilidade e credibilidade do resultado, a pesquisa contou com a colaboração de uma consultora especializada em medição por meio da utilização de PF. É uma prática recomendada a realização das estimativas por um analista de métricas que não atue diretamente na equipe do projeto (SISP, 2015).
- II. Derivação de custo, prazo e esforço a partir da aplicação do método de estimativas de Capers Jones, o mesmo utilizado oficialmente pelo SISP (2015).
- III. Comparação dos resultados obtidos a partir da medição por meio de PF (IFPUG, 2010) e resultados reais obtidos a partir da execução do projeto real.

Os resultados obtidos a partir da execução de cada uma dessas etapas serão descritos no decorrer deste capítulo.

## 4.2 RESULTADOS

### 4.3 PROJETO DE REFERÊNCIA

O PROJETO DE REFÊNCIA fez uso de uma Metodologia de Desenvolvimento SOA (MDSOA, 2016) própria, construída levando em consideração as necessidades e especificidades da CONTRATANTE, bem como padrões de indústria e práticas estabelecidas de desenvolvimento de software orientado a serviço preconizadas no mercado atualmente.

Para representar as etapas que compõem a metodologia SOA, foi utilizado um Modelo de Processos. De acordo com o CBOK (ABPMP, 2015), “Processo é um conjunto de atividades interdependentes, ordenadas no tempo e no espaço de forma encadeada, que ocorrem como resposta a eventos e que possui um objetivo, início, fim, entradas e saídas bem definidas.

A Figura 4-1 ilustra o fluxo de processos que compõem a Metodologia de Desenvolvimento SOA utilizada no desenvolvimento do PROJETO DE REFERÊNCIA, o qual é composto por 7 Serviços de Processos, 23 Serviços de Tarefa, 6 Serviços de Entidade e 4 Serviços Utilitários, totalizando a construção de 40 serviços. Além dos serviços, foram construídas interfaces gráficas para interação do usuário.

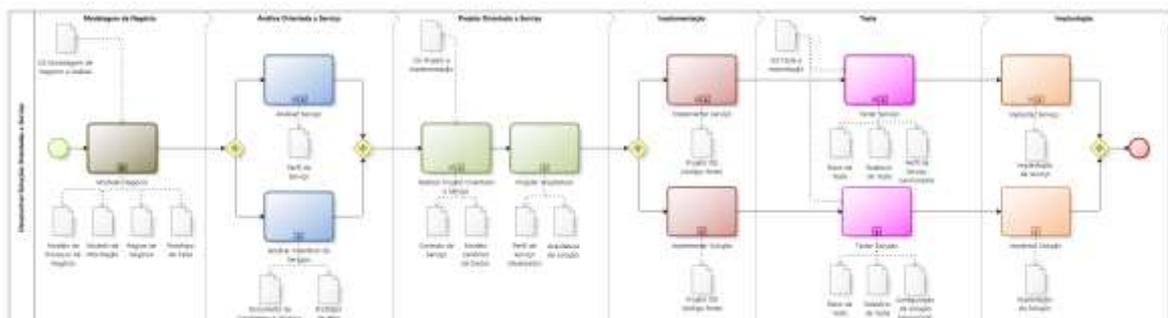


Figura 4-1: Metodologia de Desenvolvimento SOA  
Fonte: MDSOA(2016)

A fase de Modelagem de Negócio trata do entendimento da situação atual dos processos de negócio (as-is) que são objetos do desenvolvimento, seguida de análise desses processos onde se busca oportunidades de melhorias. Na fase de Análise Orientada a Serviços, identificam-se os serviços e as composições de serviços que fornecessem conjuntamente as

capacidades computacionais requeridas na automação pretendida (identifica-se os candidatos a serviços).

A fase de Projeto Orientado a Serviço trata do projeto de contratos de serviços e especificação da arquitetura da lógica de serviço, suas composições (orquestrações) e a aplicação de padrões de projeto relacionados ao contexto arquitetural de cada serviço. Na fase de Implementação são construídos os serviços, interfaces gráficas e demais componentes de software necessários para a automatização do processo de negócio.

A fase de Teste permite verificar a qualidade do produto de software implementado, identificando defeitos e atestando o funcionamento adequado do software e de seus componentes.

E, finalmente, na fase de Implantação são disponibilizados os serviços e solução de software como um todo, bem como é realizada a coleta/atualização de metadados comunicativos que suportam a descoberta e o reuso de serviços no diretório de serviços (repositório de serviços).

A seguir são apresentados os resultados reais obtidos a partir da execução do PROJETO DE REFERÊNCIA, incluindo informações sobre a equipe, esforço, prazo e custos do projeto.

#### **4.4 Equipe de Projeto**

O Quadro 4-1 apresenta a equipe alocada para o desenvolvimento da solução. São apresentadas informações a respeito da quantidade de pessoas que participaram do projeto por tipo de recurso/papel; o papel exercido pelo recurso alocado na equipe com sua respectiva porcentagem de participação; e a dedicação relativa, em percentuais, de um recurso/papel específico ao longo de todo o projeto.

Quadro 4-1: Equipe de Projeto

Qtde	Recursos	Dedicação
1	Gerente de Projetos	100%
1	Analista SOA	50%
1	Analista de Negócios	50%
1	Analista SOA	20%
1	Arquiteto de Software	20%
1	Desenvolvedor SOA	20%
1	Líder Técnico	20%
1	Analista de Infraestrutura Tecnoló	100%
2	Desenvolvedor SOA	100%
1	Desenvolvedor JAVA	45%
1	Analista SOA	16%
1	Desenvolvedor de Banco de Dado	17%
1	Analista de Banco de Dados	17%
1	Desenvolvedor SOA	45%

Fonte: PROJETO DE REFERÊNCIA

#### 4.5 Esforço Realizado

O PROJETO DE REFERÊNCIA teve uma duração de seis meses e o esforço dispensado na realização do PROJETO DE REFERÊNCIA foi registrado através do JIRA<sup>5</sup>, uma ferramenta que possibilita o acompanhamento de projetos. O Quadro 4-2 apresenta o esforço dispensado na realização de cada etapa do desenvolvimento do projeto.

Quadro 4-2: Esforço Realizado

Fase Projeto	Horas
Modelagem de Negócio e Análise	1076
Implementação	826
Teses e Implantação	268
<b>Total Horas</b>	<b>2170</b>

Fonte: PROJETO DE REFERÊNCIA

#### 4.6 Custo Realizado

O custo foi calculado através da conversão de horas para UST (Unidade de Serviço Técnico), que é a unidade de medida utilizada para quantificar o trabalho. Ou seja, a quantidade de homem-hora é convertida em UST conforme definição contratual para derivação de custos. O cálculo de UST leva em consideração a complexidade da atividade executada (baixa,

<sup>5</sup> <https://www.atlassian.com/software/jira>

média e alta). O Quadro 4-3 apresenta o resultado da conversão de horas/UST para cada fase do projeto.

Quadro 4-3: Custo Realizado

<b>Fase Projeto</b>	<b>Horas</b>	<b>UST</b>
Modelagem e Análise	1076	1765
Implementação	826	894
Teses e Implantação	268	304
<b>Total Horas</b>	<b>2170</b>	<b>2963</b>

Fonte: PROJETO DE REFERÊNCIA

Considerando o valor de UST contratado de R\$247,50, o projeto teve um custo total no valor de **R\$733.342,50**

#### 4.7 Resultado Geral

A Quadro 4-4 apresenta o sumário de esforço, prazo e custo dispensados na realização do PROJETO DE REFERÊNCIA, os quais servirão de base para análise comparativa com o resultado da medição do tamanho funcional do software por meio de PF.

Quadro 4-4: Resultado Geral

<b>PROJETO REFERÊNCIA</b>	
Esforço (horas):	2170
Prazo (meses):	6
Custo (R\$):	R\$733.342,50

Fonte: PROJETO DE REFERÊNCIA

#### 4.8 MEDIÇÃO DO TAMANHO FUNCIONAL – APF

O Quadro 4-5 apresenta o resultado da contagem tradicional, que corresponde ao valor do tamanho funcional do PROJETO DE REFERÊNCIA, calculado de acordo com regras de contagem de PF definidas pelo IFPUG (2010), onde apenas uma única fronteira envolveria toda a aplicação, independente da abordagem arquitetural utilizada. A derivação de Prazo, Esforço, Equipe e Custo foram determinados com base no modelo utilizado pelo Roteiro de Métricas do SISP (2015)<sup>6</sup> e é apresentado no Quadro 4-6.

<sup>6</sup> SISP - Sistema de Administração dos Recursos de Tecnologia da Informação. Instituído pelo Decreto nº 1.048 de 21 de janeiro de 1994 e atualizado pelo Decreto nº 7.579 de 11 de outubro de 2011, com o objetivo

Quadro 4-5: Resultado Contagem IFPUG

Documentação Disponível	Artefatos do projetos como documento de modelo de processo de negócio, regras de negócio, modelo de informação (modelo de dados) e acesso software propriamente dito.				
Propósito	Medição do tamanho funcional do PROJETO DE REFERÊNCIA, o qual foi utilizado pela CONTRATANTE como referência no processo de adoção de SOA/BPM pela organização.				
Tipo de Contagem	Desenvolvimento				
Escopo	Processo de Movimentação Funcional de Magistrados, mais especificamente, os processos de substituição de Magistrado (registro de afastamentos e gerenciamento de designações – institutos: substituindo, respondendo e cumulando) e cooperação de Magistrado (gerenciamento de designações – institutos: cooperando, cooperando (Legitimidade), cooperando (Alocação) e cooperando (Folga).				
Fronteira da Aplicação	Contou-se a aplicação como um todo, com a visão funcional do ponto de vista do usuário,				
Técnica de Contagem	Detalhada				
Funções de Dados	Total	Complexidade			
		Baixa	Média	Alta	
	Quantidade de ALI	6	5	1	0
Quantidade de AIE	9	10	0	0	
Funções de Transação	Total	Complexidade			
		Baixa	Média	Alta	
	Quantidade de EE	39	17	12	10
	Quantidade de CE	32	7	17	8
Quantidade de SE	10	3	3	4	
Tamanho Funcional do Projeto	<b>437.00</b>				

Fonte: PROJETO DE REFERÊNCIA

de organizar a operação, controle, supervisão e coordenação dos recursos de tecnologia da informação da administração direta, autárquica e fundacional do Poder Executivo Federal.

Quadro 4-6: Derivação de Prazo, Esforço, Equipe e Custo

SISP 2.1 – Produtividade Diária (horas)	6			
SISP 2.1 – Índice de Produtividade (horas/PF)	16			
SISP 2.1 – Esforço (pessoas/hora)	6.992,00	⇒ PF * Taxa de entrega * (% de esforço total)		
SISP 2.1 – Ambiente	Sistema OO			
SISP 2.1 – Prazo recomendado	8,92	Para o Ambiente acima ⇒ PF ^ 0,36		
SISP 2.1 – Carga de trabalho mensal (horas)	126	⇒ 21 dias * Produtividade Diária		
SISP 2.1 – Equipe	6,2	⇒ Esforço / (Carga Mensal * Prazo)		
Equipe informada	Qtde	Recursos	Dedicação	
	1	Gerente de Projetos	100%	
	1	Analista SOA	50%	
	1	Analista de Negócios	50%	
	1	Analista SOA	20%	
	1	Arquiteto de Software	20%	
	1	Desenvolvedor SOA	20%	
	1	Líder Técnico	20%	
	1	Analista de Infraestrutura Tecnológica	100%	
	2	Desenvolvedor SOA	100%	
	1	Desenvolvedor JAVA	45%	
	1	Analista SOA	16%	
	1	Desenvolvedor de Banco de Dados	17%	
	1	Analista de Banco de Dados	17%	
1	Desenvolvedor SOA	45%		
Total da equipe	7,2			
Outras Simulações de Estimativas	Prazo	Equipe	Esforço	
	(meses)	(pessoas)	(pessoas-hora)	
	Simulação de prazo (equipe informada)	7,71	7,2	6.992,00
	SISP 2.1 – Menor prazo possível (até -25%)	6,69	14,1	11.886,40
SISP 2.1 – Redução de prazo (estabelecido)	6,00	15,7	11.886,40	
Valor do PF	R\$ 800,00			
Custo do Projeto	Sem Red:	R\$ 349.600,00	Com Red: R\$ 593.620,00	

Fonte: PROJETO DE REFERÊNCIA

**Técnica de Contagem:** determina se a contagem será indicativa, estimada ou detalhada (NESMA, 2015).

**Tipo de Contagem:** refere-se ao tipo de contagem conforme definido na seção 3.6.

**Tamanho Funcional do Projeto:** Refere-se ao valor do tamanho funcional do PROJETO DE REFERÊNCIA.

**Produtividade Diária (horas):** Refere-se à quantidade de horas produtivas de uma pessoa no dia. Neste trabalho, foi considerado 6 horas/dia, conforme recomendado pelo (SISP, 2015).

**Índice de Produtividade (horas/PF):** Refere-se à quantidade de horas dedicadas a produção de 1 PF. A determinação do valor do índice de produtividade depende de diversos fatores, que podem incluir, entre outros, a plataforma tecnológica, segurança, desempenho,

usabilidade e tamanho do projeto. Cada organização deverá possuir uma tabela de produtividade própria, considerando dados históricos de projetos anteriores.

**Esforço (pessoa/hora):** Refere-se à quantidade de horas a serem dispensadas na entrega do projeto, ou seja, na entrega do tamanho funcional determinado. É calculado com base no tamanho funcional do projeto e índice de produtividade (Tamanho (PF) x Índice de Produtividade (HH /PF)).

**Ambiente:** Refere-se ao expoente a ser utilizado na formula para determinação do prazo recomendado. A Tabela 4-1 apresenta as opções disponíveis. Neste trabalho foi selecionado o “Sistema OO – Sistema Orientado a Objetos”, o qual mais se aproxima das características de um projeto SOA.

Tabela 4-1: Expoente por Tipo de Projeto

TIPO DE SISTEMA	EXPOENTE T
Sistema Comum – Mainframe (desenvolvimento de sistema com alto grau de reuso ou manutenção evolutiva)	0,32 a 0,33
Sistema Comum – WEB ou Cliente Servidor	0,34 a 0,35
Sistema OO (se o projeto OO não for novidade para equipe, não tiver o desenvolvimento de componentes reusáveis, considerar sistema comum)	0,36
Sistema Cliente/Servidor (com alta complexidade arquitetural e integração com outros sistemas)	0,37
Sistemas Gerenciais complexos com muitas integrações, Datawarehousing, Geoprocessamento, Workflow	0,39
Software Básico, Frameworks, Sistemas Comerciais	0,40
Software Militar (ex: Defesa do Espaço Aéreo)	0,45

Fonte: SISP, 2015 p. 45

**Prazo Recomendado:** É calculado de acordo com fórmula de Capers Jones [Jones, 2007].

**Carga de Trabalho Mensal (horas):** Refere-se à quantidade de horas úteis trabalhadas em mês por pessoa.

**Equipe:** Refere-se à quantidade de recursos necessários para desenvolvimento do projeto e deve-se considerar os percentuais de alocação do recurso ao projeto. No resultado apresentado, a equipe de projeto de 6,2 recursos poderia corresponder, por exemplo, a uma equipe de 12 pessoas com 50% de alocação e um líder de projeto com 20% de alocação ao projeto.

**Equipe Informada:** Refere-se a equipe que efetivamente trabalhou no desenvolvimento do projeto. Tem por objetivo apoiar outras simulações de esforço e prazo e a distribuição dos recursos na equipe de acordo com a dedicação de cada um ao projeto.

**Outras simulações de Estimativas:** Foram geradas outras simulações de estimativas com base na equipe informada e considerando redução de prazo. O valor do ponto de função utilizado foi definido com base em referências históricas da empresa CONTRATANTE. Foram calculados os custos do projeto sem considerar a redução de prazo e considerando a redução de prazo.

#### **4.9 ANÁLISE DOS RESULTADOS - APF**

O custo real do projeto SOA e o obtido a partir da medição por meio da APF resultaram em valores significativamente diferentes (custo real: R\$733.342,50 e custo estimado: R\$349.600,00). Observou-se, ainda, uma diferença significativa entre o esforço necessário para desenvolvimento do projeto e esforço real dispensado (esforço real: 2170 horas e esforço estimado: 6992 horas). O custo real e estimado (custo real: R\$733.342,50 versus custo estimado: R\$593.620,00) somente apresentaram valores relativamente próximos quando o método de estimativa para derivação de custo considera a redução de prazo para desenvolvimento do projeto em relação ao idealmente recomendado (prazo real: 6 meses versus prazo recomendado: 7,71 meses).

No que se refere a documentação, pode-se dizer que os artefatos produzidos nas fases iniciais de um projeto SOA são similares aos artefatos produzidos durante a fase de análise de sistemas tradicional, como, por exemplo, os documentos de regras de negócio e protótipos de telas. A diferença principal está na forma como os requisitos funcionais são representadas em cada abordagem. Enquanto a abordagem tradicional faz uso dos “Casos de Uso”, que descreve como um usuário final irá interagir com o sistema (PRESSMAN, 2011), a abordagem SOA faz uso dos Modelos de Processos de Negócios mapeados utilizando o formalismo BPMN (BPMN, 2013). No entanto, esta diferença não prejudicou o processo de contagem uma vez que tivemos acesso ao software em funcionamento.

## 5 - APRESENTAÇÃO DA PROPOSTA

### 5.1 PILARES

O método proposto, denominado APFSOA, teve como base três pilares fundamentais: (i) as regras de contagem definidas no CPM (IFPUG, 2010); (ii) as características e especificidades da abordagem de desenvolvimento SOA; e (iii) a abordagem de contagem de componentes reutilizáveis, preconizada por Woodward (2009b). Além disso, o APFSOA foi construído simultaneamente ao desenvolvimento de um projeto de software que utilizou as abordagens SOA e BPM, o que possibilitou perceber os pontos fortes, fracos e limitações de cada um dos pilares da proposta.

No sentido de manter a conformidade com o modelo definido no CPM (IFPUG, 2010), o APFSOA utiliza as mesmas etapas de contagem da APF (vide seção 3.4). No entanto, para antever às especificadas da abordagem SOA, as etapas envolvendo a contagem das Funções de Dados (Etapa iii) e Funções de Transação (Etapa iv) sofreram adaptações consideráveis, não perdendo, contudo, a essência do processo tradicional de contagem de PF do IFPUG. O método é apresentado na Figura 5-1.

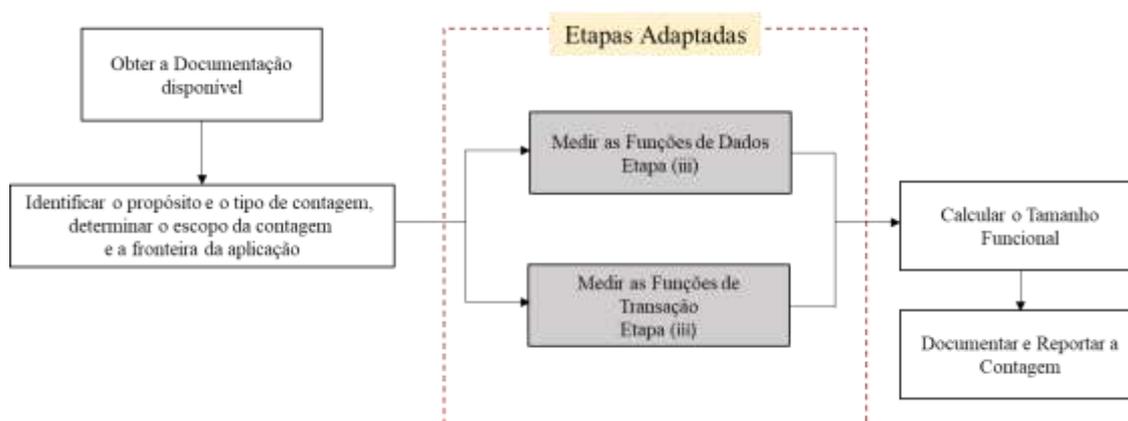


Figura 5-1: Etapas contagem de PF em Projetos SOA/BPM  
Fonte: Adaptado de IFPUG (2010)

No que se refere a SOA, o método baseou-se fortemente no Modelo de Serviços proposto por ERL (2008) e descrito na seção 3.14 deste documento.

O APFSOA baseou-se ainda, na abordagem de contagem de componentes reutilizáveis, preconizada por WOODWARD (2009b), independente por serviço, onde cada serviço é

analisado isoladamente de forma a permitir gerenciar seu uso e desenvolvimento individualmente. Ou seja, cada serviço será considerado como um componente com fronteira própria, sendo a análise das funções de dados e funções de transação realizadas individualmente, de modo a se obter o tamanho de cada serviço. Isso se justifica pelo fato de a abordagem SOA ter como principais características o reuso e composição de serviços, sendo comum a orquestração de serviços pertencentes a uma mesma camada. Desta forma, não atenderia ao propósito de evidenciar a complexidade do trabalho juntar uma ou mais camadas em uma mesma fronteira.

A seguir, são descritas as adaptações e premissas propostas neste trabalho para a contagem dos serviços, sendo que cada modelo de serviço (processo, tarefa, entidade e utilitário), devido às suas especificidades, possui regras de identificação de funcionalidades próprias.

## **5.2 DESCRIÇÃO DO MÉTODO APFSOA**

Nesta seção serão descritas as particularidades do APFSOA. Ou seja, serão descritas apenas as etapas que sofreram adaptações consideráveis em relação à contagem do IFPUG. Importante ressaltar que, devido às adaptações e convenções adotadas, os PF obtidos a partir da aplicação do APFSOA serão denominados Pontos de Função SOA (PFSOA).

### **5.3 Contar as Funções de Dados**

Serviços SOA utilizam o Modelo Canônico de Dados para compor as mensagens utilizadas por um serviço (ERL, 2008). Desse modo, a identificação das funções de dados é derivada diretamente dos esquemas do Modelo Canônico de Dados. Isso se justifica pelo fato de o modelo canônico estar totalmente voltado para a necessidade do negócio e retratar a visão do usuário (ERL, 2008). Todos os esquemas pertencentes ao escopo da contagem devem ser avaliados para identificação dos grupos lógicos associados, chamados de Arquivo Lógico Referenciado (ALR), que é um grupo de dados conforme visto pelo usuário e composto por uma ou mais entidade de dados (esquemas) denominadas Registro Lógico de Dados (RLR).

Cada ALR corresponde a uma Função de Dados na APFSOA, que será classificada, de acordo com o seu propósito, como Arquivo Lógico Interno (ALI) ou Arquivo de Interface Externa (AIE), conforme regras do CPM (IFPUG, 2010).

O Quadro 5-1 apresenta as heurísticas definidas para a contagem das Funções de Dados.

Quadro 5-1: Heurísticas Funções de Dados

<b>Heurística</b>
Cada grupo lógico de dados é composto pelo esquema principal e esquemas referenciados (esquemas importados). Cada um dos esquemas referenciados, bem como o esquema principal, é considerado um RLR - Registro Lógico Referenciado - subgrupo de dados reconhecido pelo usuário dentro do ALI/AIE.
Os atributos simples dos esquemas principais definem os DER – Dados Elementares Referenciados. Um Dado Elementar Referenciado (DER) é um campo único não-repetido e reconhecido pelo usuário (IFPUG, 2010). Os DER dos esquemas referenciados (importados) não são contabilizados na contagem. Isso se justifica pelo fato de o método não permitir a contagem de DER repetidos.
Cada Função de Dados é classificada (com pesos diferenciados), conforme o número de DER e RLR que possui, em baixa, média e alta complexidade, conforme regras descritas no CPM 4.3.1 (IFPUG, 2010). A quantidade de PFSOA (Pontos de Função SOA - Unidade de medida do APFSOA) para a função de dados de serviço é representado por $PF_{fds}$ .
A contagem das Funções de Dados para contabilização no total de PFSOA de um serviço será realizada analisando cada serviço individualmente. Devem ser contabilizados na contagem apenas as Funções de Dados referenciadas (lida/mantida) pelo serviço específico.

Fonte: Elaboração Própria

### **Exemplo:**

Neste exemplo, *DesignacaoMagistrado* é um esquema no modelo canônico de dados que inclui, em sua estrutura, além dos dados da *Designação*, dados do *Magistrado*, *VagaMagistrado* e *Portaria* (esquemas importados). Desta forma, *Designação* é uma função de dados que possui 9 DER e 4 RLR, sendo classificada com baixa complexidade, conforme ilustra o Quadro 5-2. Os atributos de *Magistrado*, *VagaMagistrado* e *Portaria* não contabilizam DER para *Designação* uma vez que não é permitida a contagem repetida de DER (já foram contabilizados em seu esquema principal) (IFPUG, 2010).

Quadro 5-2: Identificação Grupo lógico de Dados

<pre> &lt;schema elementFormDefault="qualified"   targetNamespace="http://example.com/DesignacaoMagistrado"   xmlns:tns="http://example.com/DesignacaoMagistrado"   xmlns:magistrado="http://example.com/Magistrado"   xmlns:vaga="http://example.com/VagaMagistrado"   xmlns:portaria="http://example.com/Portaria"   xmlns="http://www.w3.org/2001/XMLSchema"&gt;   &lt;!-- Arquivos Importados - RLR --&gt;   &lt;import namespace="http://example.com/Magistrado"/&gt;   &lt;import namespace="http://example.com/VagaMagistrado"/&gt;   &lt;import namespace="http://example.com/Portaria"/&gt;   &lt;!-- Grupo Lógico- RLR principal --&gt;   &lt;element name="DesignacaoMagistrado"     type="tns:DesignacaoMagistradoType"/&gt;   &lt;complexType name="DesignacaoMagistradoType"&gt;     &lt;sequence&gt;       &lt;!-- DERs --&gt;       &lt;element name="Protocolo" type="tns:ProtocoloType"/&gt;       &lt;element name="Status" type="tns:StatusType"/&gt;       &lt;element name="StatusPublicacao" type="tns:StatusType"/&gt;       &lt;element name="Instituto" type="tns:InstitutoType"/&gt;       &lt;element name="DataInicial" type="xsd:date"/&gt;       &lt;element name="DataFinal" type="xsd:date"/&gt;       &lt;element name="DataInicialEfetiva" type="xsd:date"/&gt;       &lt;element name="DataFinalEfetiva" type="xsd:date"/&gt;       &lt;element name="Observacoes" type="xsd:string"/&gt;       &lt;!-- RLR --&gt;       &lt;element ref="vaga:VagaMagistrado"/&gt;       &lt;element ref="magistrado:Magistrado"/&gt;       &lt;element ref="portaria:Portaria"/&gt;     &lt;/sequence&gt;   &lt;/complexType&gt; </pre>
<p>Grupo Lógico de Dados: DesignacaoMagistrado          RLRs: 4 (1 principal + 3 importados)          DERs: 9          Complexidade: Baixa (conforme Tabela 3-2)          PF<sub>fds</sub>: 7 (conforme Tabela 3-3)</p>

Fonte: PROJETO DE REFERÊNCIA

#### 5.4 Contar as Funções de Transação

A identificação das Funções de Transação considera as características básicas de um processo elementar. Esse conceito é fundamental para definir o processo de contagem em um projeto de software realizado com SOA e BPM. A seguir, são apresentadas algumas heurísticas usadas na contagem de PFSOA, considerando os modelos de serviço definidos:

- 1) As Funções de Transação (processo elementar) são derivadas das capacidades (funcionalidades) dos serviços identificados durante a análise do negócio ou diretamente do contrato de serviço.

- 2) Funções de Transação são classificadas conforme regras definidas no CPM em Entradas Externas (EE), Consultas Externas (CE) ou Saídas Externas (SE) conforme regras do CPM (IFPUG, 2010).
- 3) Os DER das Funções de Transação de cada capacidade de serviço compreendem a maior possibilidade (todos os atributos possíveis) de Entradas e Saídas documentadas nas Especificações Técnicas da capacidade (documentos das fases de análise e modelagem de negócio). Nestes artefatos são documentadas as entradas e saídas de cada capacidade de serviço.
- 4) Serviços de processo são compostos por diferentes tarefas, classificadas em tarefas de usuário e tarefas de serviço.
  - a. As “tarefas” devem ser analisadas, agrupadas ou decompostas para que sejam obtidos os processos elementares (Funções de Transação) que elas representam. Tarefas humanas, normalmente, são agrupadas às tarefas não humanas para compor a menor unidade de atividade significativa que deixa o sistema em estado consistente (processo elementar).
  - b. As tarefas manuais, por não fazerem parte do serviço de processo, não devem ser consideradas na contagem.

O modelo utilizado para contagem de serviços de processo é exemplificado na Figura 5-2. Neste exemplo, são identificadas duas Funções de Transação: “Realizar Tarefa 1” e “Realizar Tarefa 2”. As Funções de Dados são derivadas dos *payloads* (mensagens) de entrada e saída, que por vez são derivados são modelos canônicos de dados.

- 5) Quando ocorrem composições, cada composição é representada por uma interação com outro serviço e, mais especificamente, pela invocação de uma capacidade de serviço. Tipicamente, as composições são definidas identificando-se o fluxo de chamadas das capacidades de serviço. Desse modo, cada chamada de (capacidade de) serviço na composição envolve uma transação que ocorre fora da fronteira do serviço e já contabilizada no serviço de origem. Não contribuindo, portanto, para o tamanho funcional do software.

- 6) De um modo geral, o cálculo dos PF das funções de transação do serviço (PF<sub>fts</sub>) é dado pelo somatório dos PF de cada capacidade do serviço (PF<sub>c</sub>):

$$PF_{fts} = \sum PF_c$$

Finalmente, o cálculo dos PF totais do serviço (PF<sub>SOA<sub>s</sub></sub>) é dado pela soma de PF de suas funções de dados e suas funções de transação:

$$PF_{SOA_s} = PF_{fts} + PF_{fds}$$

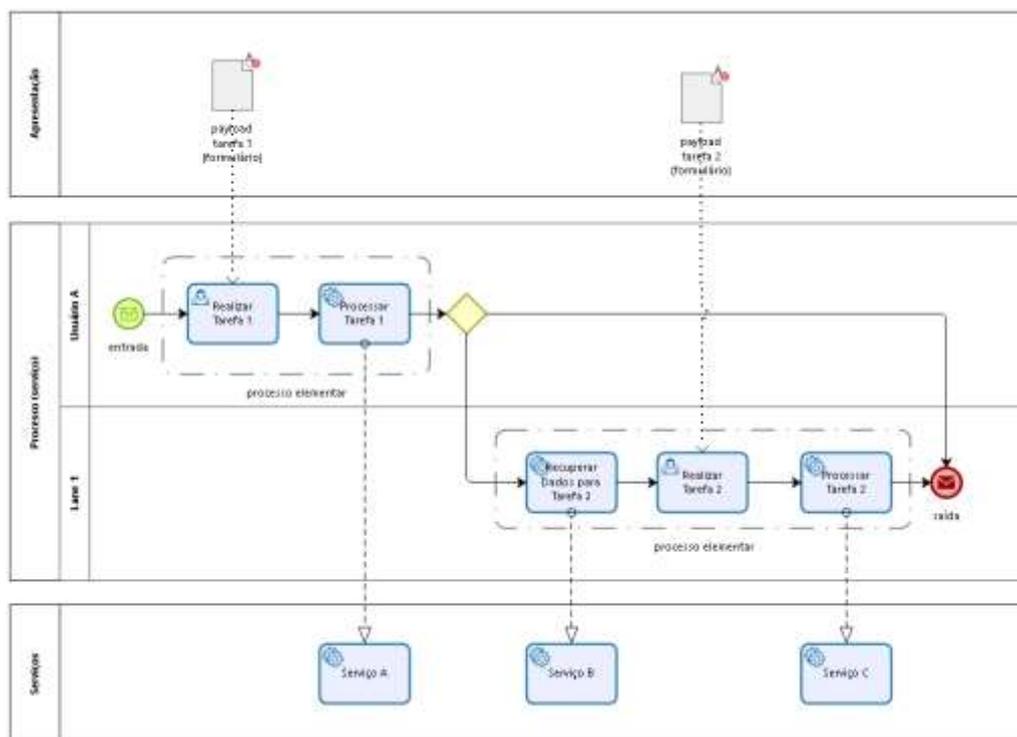


Figura 5-2: Exemplo Serviço de Processo  
Fonte: Elaboração Própria

## 5.5 Calcular o Tamanho Funcional

O cálculo do tamanho funcional é realizado individualmente por serviço através do somatório do PF<sub>fts</sub> e PF<sub>fds</sub> para obtenção do tamanho funcional do serviço em PFSOA. Os PFSOA de diferentes serviços podem ser somados para obtenção do tamanho final do software sendo medido.

## 5.6 EXEMPLO DE SERVIÇO DE ENTIDADE

Neste exemplo, o serviço de entidade *DesignacaoMagistradoService* possui duas Funções de Transação (capacidades de serviço) e cinco Funções de Dados associadas conforme

ilustram os Quadro 5-4 e Quadro 5-5 respectivamente. As funções de transação foram identificadas diretamente do contrato de serviço (vide Quadro 5-3) e de seu contexto funcional. O Quadro 5-6 apresenta o tamanho funcional do serviço *DesignacaoMagistrado* em PFSOA.

Quadro 5-3: Contrato do Serviço *DesignacaoMagistradoService*

```
<definitions
targetNamespace="http://example.com/DesignacaoMagistradoService"
xmlns:tns="http://example.com/DesignacaoMagistradoService"
xmlns:types="http://example.com/DesignacaoMagistradoService/types"
xmlns="http://schemas.xmlsoap.org/wsdl/">
<types>
<xsd:schema elementFormDefault="qualified"
targetNamespace="http://example.com/DesignacaoMagistradoService/types"
xmlns:xsd=http://www.w3.org/2001/XMLSchema
xmlns:magistrado="http://example.com/Magistrado"
xmlns:vaga="http://example.com/VagaMagistrado"
xmlns:designacao="http://example.com/DesignacaoMagistrado">
<xsd:import namespace="http://example.com/DesignacaoMagistrado"/>
<xsd:import namespace="http://example.com/Magistrado"/>
<xsd:import namespace="http://example.com/VagaMagistrado"/>
</types>
<message name="consultar">
<part name="request" element="designacao:Protocolo"/>
</message>
<message name="consultarResponse">
<part name="response" element="designacao:DesignacaoMagistrado"/>
</message>
<message name="inserir">
<part name="request" element="designacao:DesignacaoMagistrado"/>
</message>
<message name="inserirResponse">
<part name="request" element="designacao:Protocolo"/>
</message>
<portType name="DesignacaoMagistradoServicePortType">
<operation name="consultar">
<input message="tns:consultar"/>
<output message="tns:consultarResponse"/>
</operation>
<operation name="inserir">
<input message="tns:inserirRequest"/>
<output message="tns:inserirResponse"/>
</operation>
</portType>
</definitions>
```

Fonte: PROJETO DE REFERÊNCIA

Quadro 5-4: Funções de Transação Serviço *designacaoMagistradoService*

Tipo	Capacidade	Entrada (DER)	Saída (DER)	ALR	QTD DER	QTD ALR	Complexidade	PFSOA
CE	consultar	Código da Designação	DesignaçãoType (9) MagistradoType (11) VagaType (2)  Nota: derivados do modelo canônico de dados	Designação Magistrado Vaga	22	3	Alta  (conforme Tabela 3-6)	6  (conforme Tabela 3-7)
EE	inserir	Código da vaga Matricula Status Instituto Data Inicial Data Final Observacoes	Identificação da Designação	Designação	9	1	Baixa  (conforme Tabela 3-5)	3  (conforme Tabela 3-7)
<b>Total PF<sub>ts</sub></b>								<b>9</b>

Fonte: Elaboração Própria

Quadro 5-5: Funções de Dados do Serviço *designacaoMagistrado*

Tipo	Nome	QTD DER	QTD RLR	Complexidade	PF
ALI	DESIGNAÇÃO	19	4	Baixa	7
ALI	MAGISTRADO	11	1	Baixa	7
AIE	VAGA	2	4	Baixa	5
<b>Total PF<sub>fds</sub></b>					<b>19</b>

Fonte: Elaboração Própria

Quadro 5-6: Total de PFSOAs do Serviço *designacaoMagistrado*

Pontos de Função de Dados SOA - PF <sub>fds</sub>	<b>19</b>
Pontos de Função de Transação SOA - PF <sub>fts</sub>	<b>9</b>
<b>PFSOAs = PF<sub>fts</sub> + PF<sub>fds</sub></b>	<b>28</b>

Fonte: Elaboração Própria

## 5.7 EXEMPLO DE SERVIÇO DE TAREFA

Neste exemplo, o serviço de tarefa *processarAfastamentoDeferidoService* possui uma Função de Transação (capacidade de serviço) e uma Função de Dados associada conforme ilustram o Quadro 5-8 e Quadro 5-9 respectivamente. A função de Transação foi derivada diretamente do contrato de serviço (vide Quadro 5-7) e de seu contexto funcional. Apesar o serviço apresentar duas chamadas (composições) a dois serviços externos, conforme ilustra a Figura 5-3, os mesmos não contribuíram para o tamanho funcional do serviço. O Quadro 5-10 apresenta o tamanho funcional para o serviço *processarAfastamentoDeferidoService*.

Quadro 5-7: Contrato do Serviço *processarAfastamentoDeferidoService*

```
<?xml version='1.0' encoding='UTF-8'?>
<wsdl:definitions
targetNamespace="http://www.tjsc.jus.br/soa/wsdl/tarefa/processarAfastamentoDeferidoService"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:service="http://www.tjsc.jus.br/soa/schemas/services/entidade/AfastamentoMagistradoService"
xmlns:status="http://www.tjsc.jus.br/soa/schemas/messages/StatusProcessamento"
xmlns:fault="http://www.tjsc.jus.br/soa/schemas/messages/FaultMessage"
xmlns:tns="http://www.tjsc.jus.br/soa/wsdl/tarefa/processarAfastamentoDeferidoService"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
<wsdl:types>
<xsd:schema elementFormDefault="qualified"
targetNamespace="http://www.tjsc.jus.br/soa/wsdl/tarefa/processarAfastamentoDeferidoService/types">
<xsd:import id="service"
namespace="http://www.tjsc.jus.br/soa/schemas/services/entidade/AfastamentoMagistradoService"
schemaLocation="../../Schemas/services/entidade/AfastamentoMagistradoService_v1r1.xsd"/>
<xsd:import id="status" namespace="http://www.tjsc.jus.br/soa/schemas/messages/StatusProcessamento"
schemaLocation="../../Schemas/messages/StatusProcessamento_v1r0.xsd"/>
```

```

<xsd:import id="fault" namespace="http://www.tjsc.jus.br/soa/schemas/messages/FaultMessage"
schemaLocation="../../Schemas/messages/FaultMessage_v1r0.xsd"/>
</xsd:schema>
</wsdl:types>
<wsdl:message name="processarRequest">
<wsdl:part name="request" element="service:inserirRequest"/>
</wsdl:message>
<wsdl:message name="processarResponse">
<wsdl:part name="response" element="status:StatusProcessamento"/>
</wsdl:message>
<wsdl:message name="fault_message">
<wsdl:part name="fault_message" element="fault:FaultMessage"/>
</wsdl:message>
<wsdl:portType name="processarAfastamentoDeferidoServicePortType">
<wsdl:operation name="processar">
<wsdl:input message="tns:processarRequest"/>
<wsdl:output message="tns:processarResponse"/>
<wsdl:fault name="FaultMessage" message="tns:fault_message"/>
</wsdl:operation>
</wsdl:portType>
<wsdl:binding name="processarAfastamentoDeferidoServiceBinding"
type="tns:processarAfastamentoDeferidoServicePortType">
<soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>
<wsdl:operation name="processar">
<soap:operation style="document" soapAction="processar"/>
<wsdl:input>
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
<soap:body use="literal"/>
</wsdl:output>
<wsdl:fault name="FaultMessage">
<soap:fault name="FaultMessage" use="literal"/>
</wsdl:fault>
</wsdl:operation>
</wsdl:binding>
<wsdl:service name="processarAfastamentoDeferidoService">
<wsdl:port name="processarAfastamentoDeferidoServicePort"
binding="tns:processarAfastamentoDeferidoServiceBinding">
<soap:address location="http://ofmwdev01.tjsc.jus.br:8001/soa-
infra/services/default/processarAfastamentoDeferidoService/processarAfastamentoDeferidoService"/>
</wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

Fonte: PROJETO DE REFERÊNCIA

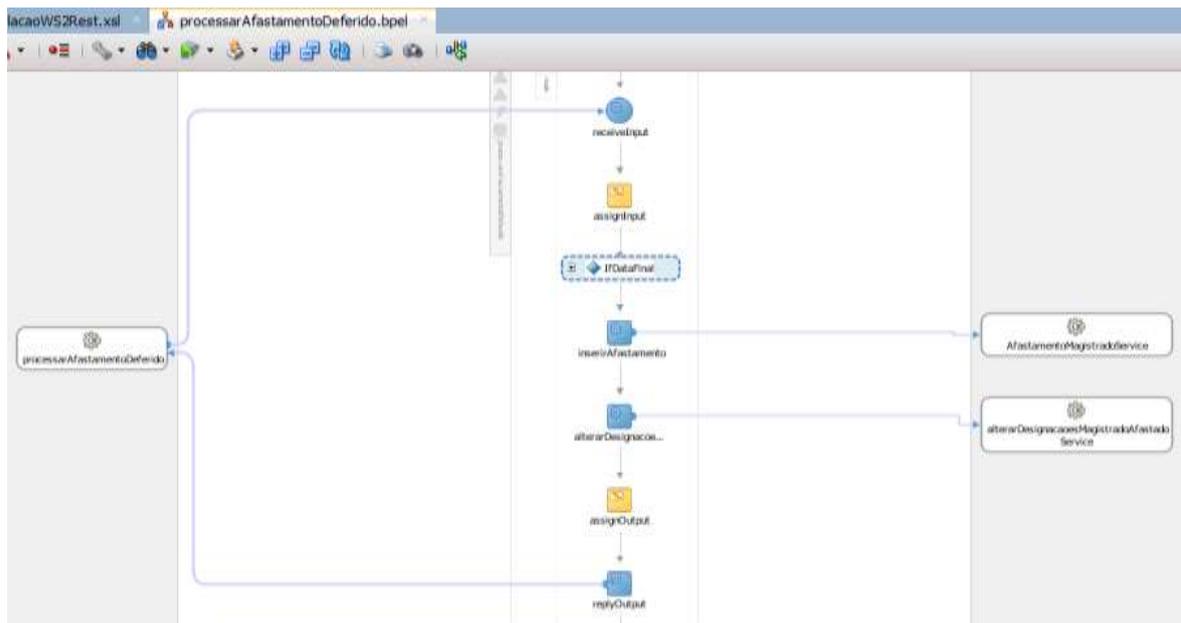


Figura 5-3: BPEL Serviço *processarAfastamentoDeferidoService*  
 Fonte: PROJETO DE REFERÊNCIA

Quadro 5-8: Funções de Transação Serviço *processarAfastamentoDeferidoService*

<b>Tipo</b>	<b>Capacidade</b>	<b>Entrada (DER)</b>	<b>Saída (DER)</b>	<b>ALR</b>	<b>QTD DER</b>	<b>QTD ALR</b>	<b>Complexidade</b>	<b>PFSOA</b>
EE	processar	Matrícula Magistrado Tipo Status PeriodoReferencia DataInicialSolicitada DataFinalSolicitada Observacoes Anexos	Status do Processamento (Código, Descrição)	Afastamento	11	1	Baixa  (conforme Tabela 3-5)	3  (conforme Tabela 3-7)
<b>Total PFSOAs</b>								<b>3</b>

Fonte: Elaboração Própria

Quadro 5-9: Funções de Dados Serviço *processarAfastamentoDeferidoService*

Tipo	Nome	QTD DER	QTD RLR	Complexidade	PF
ALI	AFASTAMENTO	15	4	Baixa	7
<b>PF<sub>fds</sub></b>					<b>7</b>

Fonte: Elaboração Própria

Quadro 5-10: PFSOA Serviço *processarAfastamentoDeferidoService*

<b>Pontos de Função de Dados SOA - PF<sub>fds</sub></b>	<b>7</b>
<b>Pontos de Função de Transação SOA - PF<sub>fts</sub></b>	<b>3</b>
<b>PFSOA<sub>s</sub> = PF<sub>fts</sub> + PF<sub>fds</sub></b>	<b>10</b>

Fonte: Elaboração Própria

## 5.8 EXEMPLO DE SERVIÇOS DE PROCESSO

Neste exemplo, o serviço de processo *registrarAfastamentoService* possui três Funções de Transação e uma Função de Dados associada conforme ilustram Quadro 5-11 e Quadro 5-12 respectivamente. As funções de Transação foram derivadas diretamente do processo BPM, sendo que as tarefas humanas e de serviço foram agrupadas para compor o processo elementar, conforme ilustra a Figura 5-4. O Quadro 5-13 apresenta o tamanho funcional do serviço em PFSOA.

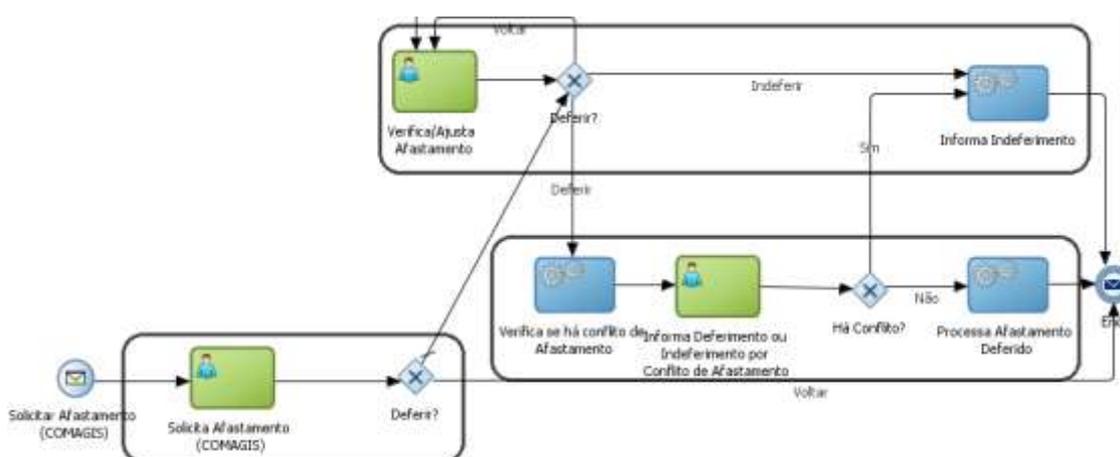


Figura 5-4: Serviço de Processo *registrarAfastamentoService*

Fonte: PROJETO DE REFERÊNCIA

Quadro 5-11: Funções de Transação do Serviço *registrarAfastamentoService*

<b>Tipo</b>	<b>Capacidade</b>	<b>Entrada (DER)</b>	<b>Saída (DER)</b>	<b>ALR</b>	<b>Qtde DER</b>	<b>QTD ALR</b>	<b>Complexidade</b>	<b>PFSOA</b>
CE	Verifica/Ajusta Afastamento	ProcessID	AfastamentoType (15)	Afastamento	16	1	Baixa (conforme Tabela 3-6)	3 (conforme Tabela 3-7)
EE	Informa Deferimento ou Indeferimento por conflito de Afastamento	ProcessID AfastamentoType (15)	AfastamentoType (15)	Afastamento	16	1	Media (conforme Tabela 3-5)	4 (conforme Tabela 3-7)
EE	Solicita Afastamento	ProcessID AfastamentoType (15)	Status do Processamento (Código, Descrição)	Afastamento	17	1	Media (conforme Tabela 3-5)	4 (conforme Tabela 3-7)
<b>Total PFSOAs</b>								<b>11</b>

Fonte: Elaboração Própria

Quadro 5-12: Funções de Dados do Serviço *registrarAfastamentoService*

Tipo	Nome	QTD DER	QTD RLR	Complexidade	PF
ALI	AFASTAMENTO	15	1	Baixa	7
<b>Pontos de Função de Dados - PF<sub>fds</sub></b>					<b>7</b>

Fonte: Elaboração Própria

Quadro 5-13: PFSOA Serviço *registrarAfastamentoService*

<b>Pontos de Função de Dados SOA - PF<sub>fds</sub></b>	<b>7</b>
<b>Pontos de Função de Transação SOA - PF<sub>fts</sub></b>	<b>11</b>
<b>PFSOA<sub>s</sub> = PF<sub>fts</sub> + PF<sub>fds</sub></b>	<b>18</b>

Fonte: Elaboração Própria

## 5.9 MEDIÇÃO DO PROJETO POR MEIO DA APLICAÇÃO DO APFSOA

A aplicação do APFSOA obteve os resultados descritos a seguir:

- a) Documentação Disponível: Foram utilizados, como insumos, a documentação do projeto, a qual inclui, entre outros, os Modelos de Processos e suas descrições e os modelos de Entidade e Informação, além do acesso direto aos artefatos de implementação e funcionalidades do sistema.
- b) Propósito: Identificar o tamanho funcional do que é entregue pelos serviços de APFSOA para obtenção do esforço de projetos SOA e BPM.
- c) Tipo de Contagem: Desenvolvimento
- d) Escopo da Contagem: funcionalidades (capacidades) de cada serviço de Tarefa, Entidade e Utilitário e funcionalidades (tarefas ou composição de tarefas) que compõem processos elementares dos serviços de Processo.
- e) Fronteira da Aplicação: Cada serviço delimita sua própria fronteira e, portanto, possui sua contagem própria.
- f) Tamanho Funcional: O Quadro 5-14 apresenta o resultado final considerando o escopo de contagem do PROJETO DE REFERÊNCIA, calculado somando-se PFSOA obtidos a partir da contagem individual de cada serviço.

Quadro 5-14: Tamanho Funcional em PFSOA para o PROJETO DE REFERÊNCIA

Modelo de Serviço	QTD	PF <sub>fds</sub>	PF <sub>fts</sub>	PFSOA	Media PFSOA por Serviço
Serviços de Processos	7	130	91	221	31.57
Serviços de Tarefa	23	240	115	355	14.43
Serviços de Entidade	6	71	151	222	37
Serviços Utilitários	4	42	30	72	18
Total	40	483	387	870	

Fonte: Elaboração Própria

- g) Derivação de esforço, prazo, custo: a derivação de custo apresentado no Quadro 5-15 foi realizada com base no mesmo valor de PF utilizado na derivação de custo a partir da aplicação da APF (IFPUG, 2010) (vide Quadro 4-6). Neste trabalho consideramos apenas a derivação de custo, não sendo escopo a aplicação ou proposição de um método de estimativas para a derivação de esforço, custo e prazo para projetos SOA.

Quadro 5-15: Derivação de Custo PROJETO DE REFERÊNCIA (APFSOA)

Derivação de Custo	
Tamanho Funcional em PFSOA	870
Valor do PFSOA	R\$800,00
Custo do Projeto	R\$696.000,00

Fonte: Elaboração Própria

**Tamanho Funcional em PFSOA:** quantidade de PFSOA obtidos a partir da aplicação do APFSOA.

**Valor do PFSOA:** valor para a produção de 1 PFSOA estabelecido em horas/PFSOA.

**Custo do Projeto:** custo total do projeto (Tamanho Funcional em PFSOA \* Valor do PFSOA).

## 5.10 ANÁLISE DOS RESULTADOS

A análise dos resultados apresentados no Quadro 5-14 mostra que a média de PFSOA prevista para a construção dos serviços de processo, tarefa, entidade e utilitário são respectivamente, 31.57, 15.43, 37 e 18. Este resultado pode ser considerado consistente, na medida em que as camadas de serviços que mais contribuíram para o tamanho funcional foram os serviços de processo e entidade. Isso se justifica pelos serviços de processos já contemplam a medição da camada de apresentação do software e os serviços de entidade serem fortemente agnósticos e reutilizáveis.

Os serviços de Tarefa e Utilitários apresentaram os menores índices, uma vez que os serviços de tarefa são, em sua maioria, serviços não-agnósticos e, portanto, não reutilizáveis; os serviços utilitários são em sua maioria agnósticos, porém não envolvem contexto negocial, o que justifica a baixa contribuição para o tamanho funcional do software.

De maneira geral, o custo real do projeto e a projeção obtida através da aplicação do APFSOA não apresentaram diferença significativa (custo real: R\$733.342,50 versus custo estimado: R\$696.000,00). No entanto, é importante salientar que a derivação de custo foi realizada com base apenas na execução do PROJETO DE REFERÊNCIA e experiência da empresa CONTRATADA. Para definição de um valor de referência para o PFSOA é necessário a aplicação do método em uma diversidade maior de projetos. Outro fator de desvio que precisa ser considerado neste contexto é o fato de o custo do PROJETO DE REFERÊNCIA ter sido baseado em horas de consultoria, que possuem custo maior que horas de desenvolvimento de software.

Os resultados obtidos mostraram, ainda, que as Funções de Dados têm maior contribuição para tamanho funcional do software do que as das Funções de Transação (Funções de Dados: 507 versus Funções Transacionais: 387). Este resultado mostra-se consistente, na medida em que serviços são construídos para serem reutilizáveis e permite, portanto, a maior combinação possível na entrada e retorno dos dados com base no Modelo de Dados Canônico, que por sua vez tem grande relevância em soluções SOA. Uma exceção a esta regra ocorreu com os serviços de entidade (Funções transacionais 151 e Funções de Dados 71), justificável pelo fato de os serviços de entidade possuírem, normalmente, um maior número de capacidades (Funções transacionais) que os demais tipos de serviços. Os serviços de Tarefa, por exemplo, possuem, em sua maioria, apenas uma capacidade de serviço.

## **6 - CONSIDERAÇÕES FINAIS**

### **6.1 SOBRE A CONDUÇÃO DA PESQUISA**

A escolha da PA como metodologia de pesquisa se mostrou acertada, uma vez que estabeleceu um processo de colaboração efetiva entre o pesquisador e os demais participantes do projeto, indispensável para a construção dos resultados obtidos. Através da PA foi possível compreender os problemas, propor e validar soluções, bem como construir o conhecimento científico de forma empírica, durante e através da intervenção, e não somente pela compilação de trabalhos acadêmicos. Apoiada pela revisão da literatura, mas construída de forma empírica, a investigação pôde mais claramente perceber as potencialidade e limitações da proposta.

### **6.2 Perspectiva Profissional *versus* Perspectiva Acadêmica**

Intervir em um projeto onde pesquisador assumiu, além do papel de pesquisador, os papéis de Gerente de Projeto e Técnico, exigiu cuidados especiais, particularmente na distinção dos papéis de pesquisador e de profissional. Na prática, essa situação gerou expectativas por parte do cliente no que se refere à imediata utilização do método proposto, bem como sua evolução em um curto espaço de tempo. Gerência de conflitos foi de extrema importância para evitar impactos negativos no papel da consultoria e, simultaneamente, viabilizar a coleta de dados necessários para a condução da investigação.

### **6.3 Principais Desafios**

A pesquisadora não possuía conhecimentos prévios a respeito de técnicas de estimativa de softwares e utilização de PF, mesmo quando a medição referia-se a projetos de softwares tradicionais (não SOA). Isso exigiu maior dedicação para absorção de conhecimentos. Igualmente, a consultora responsável pela aplicação da APF não possuía conhecimentos prévios sobre a abordagem de desenvolvimento de software com SOA.

Soma-se a isso, o fato de a documentação produzida pelo projeto apresentar falhas, não possuindo todas as informações necessárias para a utilização da APF. Para preencher esta lacuna, foi necessário acesso direto ao sistema em funcionamento, acesso aos arquivos físicos do sistema (código fonte) e discussões com a equipe de projeto.

No que se refere à construção e validação do APFSOA, a escassez de literatura científica representou desafios de cunho prático para a condução da pesquisa, razão pela qual a intervenção foi de grande valia.

Por fim, o prazo para realização da intervenção estava diretamente relacionado ao prazo contratual para realização do projeto. A pesquisadora precisou gerenciar o tempo de forma a conciliar prazos do PROJETO DE REFERÊNCIA às exigências temporais da pesquisa, e vice-versa.

#### **6.4 SOBRE O USO DA APF EM PROJETOS SOA**

A discrepância verificada a partir da comparação entre os valores projetados pela aferição do método do IFPUG (2010) e os dados empíricos do PROJETO DE REFERÊNCIA nos permite concluir que a APF não é adequada para a medição do tamanho funcional de projetos SOA (detalhes na seção 4.9). Além disso, consideramos a APF limitada por não ser adequada para a medição fragmentada da solução, situação frequente na abordagem SOA. E é justamente essa discrepância e inadequação da abordagem APF que justificam a construção de uma proposta para a medição de projetos SOA; especialmente porque essa agenda ainda não foi abraçada pelo IFPUG.

A aplicação do método APF foi importante não apenas para verificar na prática a sua limitação, já apontada na literatura, mas também para gerar insumos que contribuiriam para a construção do APFSOA.

Importante também comentar que as diferenças entre custo real de projetos SOA e estimativas geradas pela aplicação da APF não significam necessariamente que projetos SOA são mais caros que projetos de software tradicionais, algo que se poderia concluir numa primeira análise (vide Quadro 6-1). O custo de projetos SOA tende a diminuir à medida que novos projetos são desenvolvidos, devido à construção de serviços reutilizáveis e interoperáveis. A longo prazo, projetos SOA tendem a promover a redução de custo com desenvolvimento de softwares na organização.

#### **6.5 SOBRE A PROPOSTA - APFSOA**

Apesar das diversas adaptações e convenções propostas, o APFSOA foi construído de forma a mensurar o trabalho com SOA, a partir de uma abordagem que busca seguir os conceitos

do IFPUG com relação à medição funcional. Somente sofreram adaptações as etapas para contagem das Funções de Dados e Funções Transacionais.

A inclusão da medição dos serviços de processo (BPM) agregou valor ao método, uma vez que apesar de SOA e BPM serem paradigmas diferentes, o seu uso conjunto apresenta diversas oportunidades e cenários interessantes, tais como o formalismo para representação dos processos de negócio na Análise Orientada a Serviços (especificação de projetos SOA); e processos de automação, onde são construídos modelos de soluções tecnológicas que mimetizam muito proximamente os modelos de processos de negócio construídos. Desse modo, as soluções resultantes dessa abordagem conjunta apresentam graus elevados de alinhamento entre negócio e tecnologia.

## **6.6 Aplicação da Proposta**

A aplicação do APSOA no PROJETO REFERÊNCIA aponta boas perspectivas para seu uso para medição de projetos SOA. Os custos do projeto real e o estimado pelo APFSOA não apresentaram diferença significativa entre valores reais e projetados (vide Quadro 6-1); possui regras de contagem claras para futura utilização; permite a medição fragmentada da solução (individual por serviços), o que alinha ao propósito da aplicação de soluções baseadas em SOA; além de permitir a medição da solução completa, inclusive com processos BPM. Além dessas características, o método proposto é abrangente e facilmente adaptável para utilização em outros projetos e organizações. A aplicação do método mostrou que a medição dos serviços de processo já contempla o esforço necessário para a construção da camada de apresentação da solução, não sendo necessária nenhuma medição adicional.

Em relação aos pré-requisitos para aplicação do APSOA, sugere-se que os profissionais que desejem utilizar a proposta tenham conhecimentos prévios de SOA e BPM, além de domínio dos conceitos de Pontos de Função. Nesta pesquisa, o conhecimento de SOA foi essencial, pois nem todas as informações sobre o PROJETO DE REFERÊNCIA estavam devidamente documentadas, o que exigiu conhecimento técnico e acesso a arquivos físicos do sistema.

Importante salientar que o APFSOA não pode ser utilizado como referência de produtividade, custo ou taxas de entrega, valores de mercado que consideram projetos medidos em APF, por serem medidas distintas. Conseqüentemente, PF obtidos por meio da contagem

tradicional e PFSOA obtidos através da aplicação do APFSOA não podem ser somados nem comparados.

## 6.7 Projeto Real, APF e APFSOA

Uma análise conjunta dos resultados obtidos a partir da execução do Projeto Real, tamanho funcional em APF e APFSOA, nos permite visualizar e concluir que a APF não é adequada para a medição de projetos SOA, ao contrário do APFSOA, como mostra o Quadro 6-1.

Quadro 6-1: Comparação Projeto Real, APF e APFSOA

	<b>PROJETO DE REFERÊNCIA (Projeto Real)</b>	<b>APF (IFPUG)</b>	<b>APFSOA (Proposta)</b>
<b>Custo</b>	R\$733.342,50	R\$349.600,00	R\$696.000,00
<b>Prazo (Meses)</b>	6,00	7,17	<i>Não contemplado nesta pesquisa</i>
<b>Esforço</b>	2.170	6.992	

Fonte: Elaboração Própria

A comparação leva em consideração apenas estimativas de custo, já que prazo e esforço não foram contemplados nesta pesquisa.

## 6.8 SOBRE O CUMPRIMENTO DOS OBJETIVOS DA PESQUISA

Os resultados da pesquisa cumpriram o objetivo geral almejado; e ao longo da investigação os seus objetivos específicos foram alcançados e registrados (vide Quadro 6-2).

Quadro 6-2: Objetivos Específicos da Pesquisa

OBJETIVO ESPECÍFICO	EVIDÊNCIA	REGISTRO
Realizar estudo a respeito da aplicação do método desenvolvido pelo IFPUG (APF) para medição do tamanho funcional de software.	Revisão da literatura a sobre a utilização de Pontos de Função e contato com especialistas na área.	Ao longo do capítulo 1 e Capítulo 3
Realizar estudo sobre a aplicação do método de contagem de PF para medição do tamanho funcional do software em projetos SOA e BPM.	Revisão da literatura; análises dos resultados alcançados com a execução do projeto de referência	Capítulo 1 e Capítulo 5
Aplicar o método do IFPUG para a medição do tamanho de um software desenvolvido integralmente utilizando a abordagem SOA.	O método do IFPUG foi aplicado ao projeto de referência (SOA/BPM) sem considerar, no entanto, os detalhes de implementação.	Capítulo 4
Definir, a partir de uma abordagem que siga os conceitos do IFPUG e especificidades da abordagem SOA, um método para a medição do tamanho funcional de softwares SOA.	Foi definido o método APFSOA o qual considera tanto as regras básicas de contagem do IFPUG como as especificidades de uma aplicação SOA.	Capítulo 5
Aplicar e validar empiricamente o método proposto.	O APFSOA foi aplicado ao projeto de referência para medição do tamanho funcional do software.	Capítulo 5 (mais especificamente 5.6)
Contribuir para a geração de conhecimento científico relacionado à utilização de PF para a medição do tamanho funcional de softwares SOA	Elaboração de artigos e Dissertação.	Publicação de Artigo e Dissertação

Fonte: Elaboração Própria

## 6.9 LIMITAÇÕES E TRABALHOS FUTUROS

A aplicação dos métodos tradicional e APFSOA foi realizada após ser concluído o PROJETO DE REFERÊNCIA. Desta forma, tratou-se apenas a contagem detalhada do software. É interessante que o método seja aplicado utilizando as técnicas de contagem indicativa e estimada (NESMA, 2015). Além disso, a confiabilidade do método só poderá ser de fato aferida quando sua utilização for replicada em uma diversidade maior de projetos, o que não ocorreu durante o desenvolvimento desta pesquisa. A aplicação do método limitou-se a um projeto apenas.

O método precisa ainda ser aprimorado no que se refere à contagem dos serviços de processo. Apesar de a literatura já apresentar diversos estudos para a medição de BPM, não foi

identificada uma proposta que atendesse adequadamente às necessidades de medição de projetos SOA, e mais especificamente à metodologia utilizada no PROJETO DE REFERÊNCIA. Por este motivo, o método precisou contemplar a medição deste tipo de serviço. No entanto, devido à sua complexidade, poderá ser refinado em trabalhos futuros.

Outro fator não explorado neste trabalho refere-se à derivação de esforço, custo, prazo e equipe a partir do APFSOA. Uma análise histórica com base em projetos SOA precisa ser realizada para a obtenção de valores que levem a uma medição que se aproxime dos valores reais.

A derivação de custo, prazo e esforço derivados do tamanho funcional do software obtido a partir da utilização da APF limitou-se a modelo de estimativas de Capers Jones, o qual foi recomendado pelo consultora especializada em APF pelo fato de ser o modelo utilizado em órgãos públicos, como por exemplo, o SISP (2015). No entanto, existem outros modelos de estimativas no mercado, como o COCOMO II, que é um modelo paramétrico que considera diferentes aspectos no cálculo de prazo (BOEHM, 2009). Em trabalhos futuros, este modelo pode ser aplicado para avaliar se a discrepância em relação ao prazo (vide Quadro 6-1) se manterá.

Trabalhos futuros podem envolver, ainda, a aplicação do método desenvolvido recentemente pelo IFPUG (SNAP Points) para medição do tamanho não-funcional (IFPUG, 2015), em adição à medição do tamanho funcional. Desta forma, seriam consideradas a medição com a visão funcional do ponto de vista do usuário, conforme preconiza a APF padrão do IFPUG, e aplicaria o valor agregado não funcional a partir do SNAP Points para fins de valoração de todo o projeto.

Apesar de o APFSOA ainda precisar de refinamentos, já foi possível perceber oportunidades para continuidade do processo de validação e refinamento do método. A empresa CONTRATANTE mostrou-se interessada em aplicar o método proposto nos projetos SOA em andamento na organização. Desta forma, pode-se dizer que o trabalho científico realizado agregou valor à consultoria prestada, e vice-versa.

## REFERÊNCIAS BIBLIOGRÁFICAS

ABPMP (2015). *BPM CBOOK* (3ª edição).

ADAM, Sebastian; DOERR, Joerg. How to better align BPM & SOA—*Ideas on improving the transition between process design and deployment*. In: 9th Workshop on Business Process Modeling, Development and Support. 2008.

ALBRECHT, Allan J. *IBM CIS&A Guideline 313*. AD/M Productivity, 1984.

ARSANJANI, Ali, et al. "S3: A service-oriented reference architecture." *IT professional* 9.3 (2007).

ARSANJANI, Ali et al. *SOMA: A method for developing service-oriented solutions*. IBM systems Journal, v. 47, n. 3, p. 377-396, 2008.

BAKLIZKY, Maruscia; FANTINATO, Marcelo. *Relatório Técnico PPgSI-003/2012 FPA4BPM—Function Point Analysis for Business Process Management* (v. 1.0). 2012.

BAKLIZKY, Maruscia et al. *Business Process Points-A Proposal To Measure BPM Projects*. In: ECIS. 2013. p. 2.

BAKLIZKY, Diana; *Análise de Pontos de Função Aplicada a Projetos Baseados em BPM*. Métricas 2016. São Paulo. 17 de novembro de 2016. Disponível em <https://bfpug.wordpress.com/conferencia-2016/>.

BIEBERSTEIN, Norbert et al. *Executing SOA: A Practical Guide for the Service-Oriented Architect*. Pearson Education, 2008.

BPMN 2.0 - *Business Process Model and Notation*, Innovator for Business Analysts, 2013. [Online]. Available: <http://www.omg.org/spec/BPMN/2.0/>. [Accessed: 27/02/2017].

BRYMAN, Alan. *Social research methods*. Oxford university press, 2004.

BOEHM, Barry W., Ray Madachy, and Bert Steece. *Software cost estimation with Cocomo II with Cdrom*. Prentice Hall PTR, 2000.

BUREEL, G & MORGAN, G. (1979) *Sociological Paradigms and Organizational Analysis*. London, Heineman.

CALAZANS Angélica, Iraina Cristina Diniz Lisboa, Marcelo Antonio Lopes de Oliveira. *Avaliação das Características Gerais de Sistemas na Análise por Pontos de Função - APF por meio da aplicação do GQM – Goal, Questions, Metrics*. VII Simpósio Internacional de Melhoria de Processos de Software. São Paulo. Novembro de 2011.

CHARMAZ, K. *Grounded Theory as an Emergent Method*. Handbook of Emergent Methods, v. 3, p. 155–170, 2008.

Coughlan, P. & Coughlan D. *Action Research for Operations Management*. International Journal of Operations and Production Management, Vol. 22: 2, pp. 220-240. 2002.

COSMIC, *The COSMIC Functional Size Measurement Method v4.0.1*, Measurement Manual, 2015.

CRESWELL, J. W. *Projeto de pesquisa: Métodos qualitativo, quantitativo e misto*. 3. ed. Porto Alegre: Artmed, 2010.

DEKKERS, C. *Measuring the “logical” or “functional” Size of Software Projects and Software Application*. *Spotlight Software*, ISO Bulletin May 2003 pp10-13.

ENDREI, Mark et al. *Patterns: service-oriented architecture and web services*. IBM Corporation, International Technical Support Organization, 2004.

ERL, Thomas. *Service-Oriented Architecture: Concepts, Technology, and Design*. Crawfordsville: Prentice Hall PTR, Aug. 2005.

ERL, Thomas. *SOA Principles of Service Design*. Prentice Hall, 2008.

FLORAC, William A.; PARK, Robert E.; CARLETON, Anita D. *Practical software measurement: Measuring for process management and improvement*. CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST, 1997.

FARRAG, Esraa A.; MOAWAD, Ramadan; IMAM, Ibrahim F. *An Approach for Effort Estimation of Service Oriented Architecture (SOA) Projects*. November 20, 2015.

GENCEL, Cigdem; DEMIRORS, Onur. *Functional size measurement revisited*. ACM Transactions on Software Engineering and Methodology (TOSEM), v. 17, n. 3, p. 15, 2008.

GOMES, Y. M. P. *Functional size, effort and cost of SOA projects with function points*. no. LXVIII, November, 2012.

Gartner (D. Cearley, J. Fenn, and D. Plummer). *Gartner's Positions on the Five Hottest IT Topics and Trends*. 2005.

GLINZ, Martin. *On non-functional requirements*. In: 15th IEEE International Requirements Engineering Conference (RE 2007). IEEE, 2007. p. 21-26.

GOMES, Luiza Barcelos Gualberto et al. *Software measure based on BPMN activity points*. In: Information Systems and Technologies (CISTI), 2016 11th Iberian Conference on. IEEE, 2016. p. 1-6.

HULT, M. e LENNUNG, S. (1980) *Towards a Definition of Action Research: A Note and Bibliography*, Journal of Management Studies 17(2), pp. 242-250.

IFPUG - INTERNATIONAL FUNCTION POINT USERS' GROUP. *Manual de Práticas de Contagem de Pontos de Função*. Version 4.3.1, Janeiro 2010

IFPUG - INTERNATIONAL FUNCTION POINT USER' GROUP. *Assessment Practices Manual*. Version 2.3, Maio 2015.

ISO/IEC, "14143-1:2007 *Information technology - Software measurement - Functional size measurement - Part 1: Definition of Concepts*", JTC 1 / SC 7, ISO/IEC, 2007.

JOSUTTIS, Nicolai M. *SOA in practice: the art of distributed system design*. " O'Reilly Media, Inc.", 2007.

JONES, C. *Estimating Software Costs – Bringing Realism to Estimating*. 2nd Edition, Mc Graw Hill, New York, 2007. New York.

KAUR, Prabhjot. *A Review of Software Metric and Measurement*. *International Journal of Computer Applications & Information Technology*, v. 9, n. 2, p. 187, 2016.

KNELLER, G. F. *A ciência como atividade humana*. Rio de Janeiro: Zahar, 1980, p. 11-35.

LINDSKOOG Jeff. *Applying Function Points Within a SOA Environment*. EDS, An HP Company. 1401 E. Hoffer St. Kokomo, IN 46902. USA. September 2009.

MaXWELL, J. A. *Designing a qualitative study*. In: BICKMAN, L; ROG, D. (Ed.) *Handbook of Applied Social Research Methods*. Thousand Oaks CA: Sage, 2008. p. 214-253.

McKERNAN, apud HOPKINS, D. *A teachers guide to classroom research*. Buckingham, 1993. p. 52.

MDSOA. *Metodologia de Desenvolvimento de Softwares SOA*. Empresa CONTRATANTE, 2016.

NESMA, *Netherlands Function Points Users*, 2015.

PINTO, M. D. R., & SANTOS, L. L. D. S. *A Grounded Theory como abordagem metodológica: relatos de uma experiência de campo*. *Organizações & Sociedade*, v. 19, n. 62, p. 417–436, 2012.

PRESSMAN, Roger S. *Engenharia de software: uma abordagem profissional*. 7ª Edição. Ed: McGraw Hill, 2011.

ROETZHEIM, William H. *Estimating software costs*. *SOFTWARE DEVELOPMENT-SAN FRANCISCO-*, v. 8, n. 10, p. 66-68, 2000.

SISP, *Roteiro de Métricas de Software do SISP: versão 2.1*, Ministério do Planejamento, Orçamento e Gestão. Secretaria de Logística e Tecnologia da Informação. Brasília: MP, 2015.

SOARES DOS REIS, Julio Cesar; BARBOSA, Marcelo Werneck. *PROPOSTA DE UMA TÉCNICA DE ESTIMATIVA PARA REQUISITOS*. *Revista de Sistemas e Computação-RSC*, v. 3, n. 1, 2013.

TICHENOR Charley, SNAP-Case Study 1 (vsn 1. 0): *Assessing the Size of Extensive Mathematical Operations Using SNAP*. Julho de 2013.

TICHENOR Charley, SNAP-Case Study 2 (vsn 1. 0): *How to Use Function Points and SNAP to Improve a Software Acquisitions Contract*. Junho de 2014.

TICHENOR, Charley. *A new software metric to complement function points: the Software Non-functional Assessment Process (SNAP)*. DEFENSE SECURITY COOPERATION AGENCY WASHINGTON DC, 2013.

THIOLLENT, M. (1983) *Problemas de metodologia*. In: Fleury, A.C.C & Vargas, N. Organização do Trabalho. São Paulo. Atlas

THIOLLENT, M. (1996) *Metodologia da Pesquisa-Ação*. São Paulo. Cortez

THIOLLENT, M. (1997) *Pesquisa-Ação nas Organizações*. São Paulo: Editora Atlas.

TERENCE, Ana Cláudia Fernandes; ESCRIVÃO FILHO, Edmundo. *Abordagem quantitativa, qualitativa e a utilização da pesquisa-ação nos estudos organizacionais*. Encontro Nacional de Engenharia de Produção, v. 26, p. 1-9, 2006.

VASQUEZ, Carlos Eduardo; SIMÕES, Guilherme Siqueira; ALBERT, Renato Machado; *Análise de ponto de função: Medição, Estimativas e Gerenciamento de Projetos de Software*. 13. ed. SP: Érica, 2013.

WOODWARD, Steven. Maschino, Deb et al (2009a). *Function Points & Counting Middleware Software Applications*, New Environments Committee NEC White Paper, June 22, 2009.

WOODWARD Steven and Heller Roger (2009b); "*Sizing Component-Based Development using Function Points*", New Environments Committee, IFPUG, September 2009.

WOODWARD Steven and Heller Roger; "*Using Function Points to Measure Reusable Software*", New Environments Committee, IFPUG, August 2010.

WINTERGREEN RESEARCH. *SOA Market Research Report*, 2014. Disponível em: <http://wintergreenresearch.com/reports/SOA%20Engines.html>, acessado em 24/01/2017.

WILKIE, F. George et al. *The value of software sizing*. *Information and Software Technology*, v. 53, n. 11, p. 1236-1249, 2011.