



**UNIVERSIDADE DO PLANALTO CATARINENSE
DEPARTAMENTO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
CURSO DE SISTEMAS DE INFORMAÇÃO
(BACHARELADO)**

**APLICAÇÃO DA ANÁLISE DE PONTOS DE FUNÇÃO NO PROCESSO
DE DESENVOLVIMENTO DE *SOFTWARE***

MÔNICA PAGNO DA SILVA

LAGES, DEZEMBRO DE 2007

**UNIVERSIDADE DO PLANALTO CATARINENSE
DEPARTAMENTO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
CURSO DE SISTEMAS DE INFORMAÇÃO
(BACHARELADO)**

**APLICAÇÃO DA ANÁLISE DE PONTOS DE FUNÇÃO NO PROCESSO
DE DESENVOLVIMENTO DE *SOFTWARE***

**Relatório do Trabalho de Conclusão de
Curso submetido à Universidade do
Planalto Catarinense para obtenção dos
créditos de disciplina com nome
equivalente no curso de Sistemas de
Informação - Bacharelado.**

MÔNICA PAGNO DA SILVA

Orientadora: Prof^a. Viviane Duarte
Bonfim, M.Sc.

LAGES, DEZEMBRO DE 2007

**APLICAÇÃO DA ANÁLISE DE PONTOS DE FUNÇÃO NO PROCESSO DE
DESENVOLVIMENTO DE *SOFTWARE***

MÔNICA PAGNO DA SILVA

**ESTE RELATÓRIO, DO TRABALHO DE CONCLUSÃO DE CURSO, FOI
JULGADO ADEQUADO PARA OBTENÇÃO DOS CRÉDITOS DA
DISCIPLINA DE TRABALHO DE CONCLUSÃO DE CURSO DO VIII
SEMESTRE, OBRIGATÓRIA PARA OBTENÇÃO DO TÍTULO DE:**

BACHAREL EM SISTEMAS DE INFORMAÇÃO

Prof^a. Viviane Duarte Bonfim, M.Sc.
Orientadora

BANCA EXAMINADORA:

Prof. Wilson Castello Branco Neto, Dr.
UNIPLAC

Prof. Alexandre Perin de Souza, M.Sc.
UNIPLAC

Prof. Angelo Augusto Frozza, M.Sc.
Professor de TCC

Prof. Wilson Castello Branco Neto, Dr.
Coordenador de Curso

Lages, 10 de Dezembro de 2007

Dedico este trabalho para a minha mãe e irmã que me apoiaram durante a conquista desta etapa e também para o meu pai (*in memoriam*) que sempre sonhou com esta conquista.

Agradeço a minha orientadora, Viviane Duarte Bonfim pela orientação, amizade, paciência e compreensão durante o desenvolvimento deste trabalho.

Agradeço ao Prof. Angelo pela disponibilidade, paciência e dedicação.

Agradeço aos meus amigos e pessoas próximas pela compreensão e apoio durante esta etapa.

Agradeço a todas as pessoas que participaram de alguma maneira na realização deste trabalho.

Agradeço a Deus por mais esta conquista.

“Se podemos ser doutores pela ciência
alheia, também podemos ser sábios pela
nossa sabedoria”.
(Montaigne)

“Faça bem-feito não só para agradar aos
outros, mas principalmente para agradar a
si próprio”.
(Autor desconhecido)

SUMÁRIO

LISTA DE ILUSTRAÇÕES	IX
LISTA DE SIGLAS	XI
RESUMO	XII
ABSTRACT	XIII
1 INTRODUÇÃO	1
1.1 Apresentação	1
1.2 Descrição do problema	3
1.3 Justificativa	3
1.4 Objetivo geral	5
1.5 Objetivos específicos	5
1.6 Metodologia	5
2 MÉTRICAS DE SOFTWARE	7
2.1 Contextualização	7
2.2 Métricas de <i>software</i>	8
2.3 Tipos de métricas	11
2.3.1 Métricas do processo	11
2.3.2 Métricas do produto	11
2.4 Medidas diretas e indiretas	12
2.4.1 Medidas diretas	12
2.4.2 Medidas indiretas	13
2.5 Categorização das métricas	13
2.6 Conclusão	14
3 ANÁLISE DE PONTOS DE FUNÇÃO	16
3.1 Definição da APF	16
3.2 APF <i>versus</i> métricas de processo	17
3.3 Características e propósitos da APF	17
3.4 Estrutura do processo de contagem da APF e elementos envolvidos	18
3.5 Processo de contagem	24
3.5.1 <i>Passo 1</i>	24
3.5.2 <i>Passo 2</i>	24
3.5.3 <i>Passo 3</i>	24
3.5.4 <i>Passo 4</i>	28
3.5.5 <i>Passo 5</i>	29

3.5.6 Passo 6	29
3.6 Trabalhos correlatos	31
3.6.1 Medição de PF a partir da especificação de requisitos	32
3.6.2 Análise e melhoria de um processo de estimativas de tamanho de projetos de software	32
3.6.3 Roteiro para aplicação de métricas e qualidade em subcontratação de software	33
3.7 Conclusão	34
4 DEMONSTRAÇÃO DA APLICAÇÃO DOS PASSOS DA APF NO ESTUDO DE CASO.....	35
4.1 Estudo de caso	35
4.2 Demonstração	41
4.2.1 Determinação do tipo de contagem	41
4.2.2 Identificação da fronteira da aplicação e do escopo da contagem	41
4.2.3 Contagem dos PF não-ajustados	41
4.2.4 Definição do valor do fator de ajuste	46
4.2.5 Contagem dos PF ajustados.....	48
4.3 Aplicação dos PF na estimativa de prazos e custos.....	48
4.3.1 Estimativa de esforço empregado na execução do projeto.....	49
4.3.2 Estimativa de duração do projeto	51
4.3.3 Estimativa de custo do projeto	52
4.4 Conclusão	54
5 CONSIDERAÇÕES FINAIS	55
REFERÊNCIAS BIBLIOGRÁFICAS	57
APÊNDICE - TUTORIAL PARA APLICAÇÃO DA ANÁLISE DE PONTOS DE FUNÇÃO NO PROCESSO DE DESENVOLVIMENTO DE SOFTWARE	60
1. INTRODUÇÃO.....	60
2. ESTUDO DE CASO	61
3. DEMONSTRAÇÃO DA APLICAÇÃO DA APF	61
4. CONSIDERAÇÕES FINAIS.....	66
5. REFERÊNCIAS BIBLIOGRÁFICAS	66
ANEXOS.....	68

LISTA DE ILUSTRAÇÕES

FIGURA 1 -	Visão Geral do Processo de Contagem de Pontos de Função.....	18
FIGURA 2 -	Elementos da Contagem de Pontos de Função	23
QUADRO 1 -	Escala de nível de influência das CGS.....	29
QUADRO 2 -	Informações do arquivo aluno.....	38
QUADRO 3 -	Informações do arquivo autorização	38
QUADRO 4 -	Informações do arquivo categoria.....	38
QUADRO 5 -	Informações do arquivo chefe do SERED	38
QUADRO 6 -	Informações do arquivo curso de graduação.....	39
QUADRO 7 -	Informações do arquivo habilitação	39
QUADRO 8 -	Informações do arquivo pró-reitor de ensino	39
QUADRO 9 -	Informações do arquivo reconhecimento	39
QUADRO 10 -	Informações do arquivo registro	40
QUADRO 11 -	Informações do arquivo reitor	40
QUADRO 12 -	Informações do arquivo resolução	40
QUADRO 13 -	Informações do arquivo secretário acadêmico.....	40
QUADRO 14 -	Fator de ajuste da aplicação em questão	46
QUADRO 15 -	Requisito manter cadastro de curso de graduação	74
QUADRO 16 -	Requisito manter cadastro de aluno	74
QUADRO 17 -	Requisito manter cadastro de reitor.....	74
QUADRO 18 -	Requisito manter cadastro de pró-reitor de ensino.....	74
QUADRO 19 -	Requisito manter cadastro de secretário acadêmico.....	74
QUADRO 20 -	Requisito manter cadastro de registro	75
QUADRO 21 -	Requisito manter cadastro de chefe do SERED	75
QUADRO 22 -	Requisito manter cadastro de autorização	75
QUADRO 23 -	Requisito manter cadastro de reconhecimento.....	75
QUADRO 24 -	Requisito manter cadastro de categoria de curso de graduação	75
QUADRO 25 -	Requisito manter cadastro de habilitação de curso de graduação	75
QUADRO 26 -	Requisito manter cadastro de resolução de curso de graduação	76
QUADRO 27 -	Requisito gerar diploma	76

TABELA 1 -	Tabela de complexidade funcional dos ALI e AIE.....	26
TABELA 2 -	Tabela de contribuição dos PF não-ajustados das funções do tipo dado.....	26
TABELA 3 -	Tabela de complexidade funcional para EE.....	27
TABELA 4 -	Tabela de complexidade funcional para SE e CE	27
TABELA 5 -	Tabela de contribuição dos PF não-ajustados das funções do tipo transação.....	28
TABELA 6 -	Tabela para o cálculo dos PF não-ajustados ou brutos	28
TABELA 7 -	Identificação dos Arquivos Lógicos Internos (ALI)	41
TABELA 8 -	Identificação das Entradas Externas (EE)	43
TABELA 9 -	Cálculo dos PF não-ajustados ou brutos	46
TABELA 10 -	Produtividade em horas/PF por linguagem	50

LISTA DE SIGLAS

AIE	- Arquivo de Interface Externa
ALI	- Arquivo Lógico Interno
APF	- Análise de Pontos de Função
BFPUG	- <i>Brazilian Function Point Users Group</i>
CE	- Consulta Externa
CGS	- Características Gerais do Sistema
CMMI	- <i>Capability Maturity Model Integration</i>
COCOMO	- <i>Constructive Cost Model</i>
EE	- Entrada Externa
GQM	- <i>Goal/Question/Metrics</i>
HH	- Homens_hora
IBM	- <i>International Business Machines</i>
IFPUG	- <i>International Function Point Users Group</i>
PF	- Pontos de Função
SE	- Saída Externa
SERED	- Setor de Registro e Expedição de Diplomas e Certificados
SERPRO	- Serviço de Processamento de Dados
UML	- <i>Unified Modeling Language</i>
UNIPLAC	- Universidade do Planalto Catarinense

RESUMO

Análise de Pontos de Função é uma métrica para medir o tamanho funcional do *software* baseado no que o usuário entende e descreve sobre ele. Este trabalho tem como objetivo fornecer um referencial teórico sobre a importância desta métrica, proporcionar uma orientação sobre seu uso através da demonstração da sua aplicação em um estudo de caso e demonstrar a utilização do valor obtido nas estimativas de prazos e custos de projetos de *software*, visto que um dos maiores problemas enfrentados atualmente, por parte da gerência de projetos, é determinar com precisão o cronograma e custo dos projetos. A demonstração da aplicação da Análise de Pontos de Função foi realizada em um estudo de caso, através da análise dos seus requisitos (funcionalidades) para que o processo de contagem fosse possível. Dentro disto, após a aplicação da métrica verificou-se sua eficácia, pois possibilita os subsídios necessários para execução das estimativas, que consistem em cálculos que apresentam como resultado, indicativos, necessários para que uma análise precisa seja feita em relação ao andamento do projeto em questão.

Palavras-chave: Métricas de *Software*; Análise de Pontos de Função; Estimativas.

ABSTRACT

Analysis of Points of Function is a metric to measure the functional size of software based on what the user understand and writes about it. This work has the objective to give a theoretical referential over the importance of this metric, to proportionate an orientation over its use through the demonstration of its application on a case study and demonstrate the utilization of the obtained amount on the timing and cost estimate for software projects, since one of the major problems currently faced by the management is to determine precisely the chronogram and cost of the projects. The demonstration of the application of the Analysis of Points of Function was made on a case of study, through analysis of its requirements (functionalities) in order to make counting process possible. Thus, after the metric application, its efficacy was verified, because it makes it possible the necessary subsidies for the execution of the estimates, which consist on calculations that present indicatives as a result, necessary to the realization of a precise analysis in relation to the development of the aforementioned project.

Keywords: Software Metrics, Analysis of Points of Function, Estimates.

1 INTRODUÇÃO

1.1 Apresentação

As empresas estão investindo cada vez mais em tecnologia da informação, pois o mercado competitivo exige que investimentos sejam feitos para se manter nesse mercado. Com isto, as empresas procuram transformar atividades manuais em informatizadas, através de sistemas que possam atender tais necessidades. Contudo, as empresas esperam que os seus sistemas implantados tenham uma qualidade satisfatória e que sejam entregues dentro dos custos e prazos definidos inicialmente.

Dentre tais exigências e para contemplar a demanda existente, empresas de desenvolvimento de *software* também devem investir em alguns aspectos como, por exemplo, na qualidade do processo de desenvolvimento dos sistemas, o que muitas vezes não acontece.

Nesse contexto, a Engenharia de *Software* oferece um conjunto de ferramentas, métodos e procedimentos que, quando usados adequadamente no processo de desenvolvimento, podem garantir parte da qualidade do processo e produtividade (PAULA FILHO, 2003).

No entanto, além dessas ferramentas, métodos e procedimentos, a Engenharia de *Software* propõe atividades que podem ser adaptadas ao processo de desenvolvimento com a finalidade de aprimorar ainda mais o projeto em estudo.

Dentre as atividades existentes, destacam-se as Métricas de *Software*, através das quais é possível acompanhar, analisar e controlar o progresso dos projetos, permitindo identificar aspectos que necessitam de melhorias.

São vários os motivos que podem levar as empresas a adotar uma métrica

para controlar o desenvolvimento dos seus projetos de sistemas: prover subsídios para determinar esforço, recursos, duração e custo do desenvolvimento; gerenciar contratos de *software*; indicar a qualidade do produto; formar uma *baseline*¹ para estimativas; avaliar a produtividade do processo. Esta prática interessa principalmente aos gerentes de projetos, pois é deles a responsabilidade de verificar e acompanhar esses aspectos (PRESSMAN, 2001).

Para muitos gerentes de projetos, a qualidade do processo de *software* é tão importante quanto a qualidade do produto. O que mais os preocupa, pode-se dizer que seja a constante imprecisão na definição de prazos e custos de seus projetos, o que pode ser controlado através de métricas existentes para tal finalidade (ROCHA, MALDONADO e WEBER, 2001).

Para minimizar o problema da real imprecisão, destaca-se a Análise de Pontos de Função, que é um instrumento utilizado por profissionais da área de sistemas em empresas de todos os portes e segmentos da economia brasileira para oferecer indicadores que possam auxiliar no aspecto citado anteriormente. Pode-se dizer que é uma métrica importante para coletar métricas referentes às estimativas de custos e prazos e que tem sido bastante aceita na gerência de projetos (VAZQUEZ, SIMÕES e ALBERT, 2006).

A elaboração deste trabalho está voltada para o estudo e aplicação da Análise de Pontos de Função no desenvolvimento de *software* e está organizado da seguinte forma: o capítulo 1 contém a introdução do trabalho; o capítulo 2 consiste em uma abordagem geral sobre Métricas de *Software*. O capítulo 3 apresenta a contextualização da Análise de Pontos de Função. O capítulo 4 é composto pela demonstração da aplicação da Análise de Pontos de Função ao estudo de caso para obtenção da estimativa de prazos e custos. E, finalizando, no capítulo 5 são realizadas as considerações finais a respeito do trabalho desenvolvido.

¹ Linha base para estimativas.

1.2 Descrição do problema

Um dos principais problemas encontrados atualmente é o não cumprimento de prazos no desenvolvimento de *software*, o que gera um aumento considerável nos custos previstos para o projeto.

1.3 Justificativa

O *software* está presente em todos os aspectos no dia-a-dia das pessoas. Desde uma simples utilização de programas automatizados para pagamentos de contas, até sistemas críticos que podem afetar a saúde e o bem-estar quando não proporcionam a segurança e confiabilidade desejadas. Neste contexto, a Engenharia de *Software* é muito importante, pois suas boas práticas, procedimentos e ferramentas buscam assegurar a qualidade desejada, baixo custo e confiabilidade.

Entende-se como qualidade de um produto o seu grau de conformidade com os respectivos requisitos. Já, o usuário julga que um *software* é de qualidade quando este faz o que ele quer que faça, sendo também fácil de aprender e utilizar. No entanto, para se alcançar a qualidade do produto é importante que a qualidade no processo de desenvolvimento seja mantida. Isto é possível através de práticas de apoio sugeridas pela Engenharia de *Software*, mencionada anteriormente. Entretanto, ainda existindo várias práticas para melhorar o processo de desenvolvimento e, conseqüentemente, melhorar a qualidade do produto, alguns fatores de planejamento podem influenciar negativamente no desenvolvimento do *software*. Como exemplo, pode-se citar atraso nos prazos estimados e aumento do custo envolvido no desenvolvimento, o que muitas vezes pode resultar no abandono do *software* (ROCHA, MALDONADO e WEBER, 2001).

No entanto, a Engenharia de *Software* oferece práticas que podem auxiliar os gerentes de projeto a estimar prazos e custos mais precisos, sem que estes tenham que se basear em experiências anteriores, visto que é um procedimento não adequado, pois

nenhum projeto é idêntico ao outro. Essas práticas de apoio são chamadas de Métricas de *Software* (PAULA FILHO, 2003).

As métricas são cálculos, através dos quais é obtido um resultado e que, a partir deste resultado, é possível chegar a uma interpretação. Esta interpretação pode auxiliar os gerentes a tomar uma decisão dentro do processo de desenvolvimento, melhorando algum aspecto identificado com a interpretação do resultado obtido através da métrica aplicada (VAZQUEZ, SIMÕES e ALBERT, 2006).

Embora muitas empresas não se preocupem com esse aspecto, a idéia de estimar prazos e custos envolvidos em um projeto é algo bastante decisivo no desenvolvimento e, por este motivo, empresas que prezam pela qualidade estão cada vez mais aderindo a essa prática. Porém, muitas empresas deixam de adaptar métricas ao seu processo pelo desconhecimento do assunto por parte dos gerentes de projeto, que ainda estão mal preparados para definir prazos baseados em um indicativo mais preciso.

Nesse contexto, destaca-se a Análise de Pontos de Função, a qual se trata de uma métrica que permite medir a funcionalidade de um *software* sob a visão do usuário e diante dos requisitos descritos pelo mesmo. Esta é uma das métricas de estimativa de tamanho mais sedimentadas no mercado e que proporciona resultados cada vez mais precisos à medida que artefatos das fases de análise e projeto são gerados (VAZQUEZ, SIMÕES e ALBERT, 2006).

Diante das considerações expostas, é demonstrada a métrica e a aplicação da Análise de Pontos de Função ao processo de desenvolvimento, especificamente na determinação de prazos e custos, como forma de orientação e auxílio para o uso da métrica. Neste enfoque, a medida obtida da aplicação da Análise de Pontos de Função é utilizada como a métrica que pode orientar a determinação desses prazos e custos como uma forma de estimar o que se deseja. Isto pode auxiliar gerentes de projetos em um controle mais preciso sobre o cronograma e custos estabelecidos no início do projeto de *software* e, também, no esclarecimento da importância do uso de Métricas de *Software*, justificando assim, a proposta do trabalho.

1.4 Objetivo geral

Aplicar a métrica Análise de Pontos de Função em um estudo de caso, visando melhorias na precisão das estimativas de prazos e custos.

1.5 Objetivos específicos

- a) Fornecer um referencial teórico sobre a importância de Métricas de *Software*, bem como, da Análise de Pontos de Função;
- b) Proporcionar uma orientação do uso da Análise de Pontos de Função através da demonstração da métrica na determinação de estimativas para prazos e custos no estudo de caso.

1.6 Metodologia

O desenvolvimento deste trabalho iniciou-se com a seleção de material bibliográfico adequado, entre eles, livros, artigos, dissertações e trabalhos relacionados.

Posteriormente, foram realizados estudos sobre Engenharia de *Software*, gerenciamento de projetos, estimativas e Métricas de *Software*, assunto de maior destaque. Neste, técnicas para aplicação de métricas foram abordadas.

A seguir, foi estudado o tema Análise de Pontos de Função, incluindo seu conceito, características, descrição da mesma, bem como, exemplos de trabalhos já elaborados. Este estudo foi importante porque esta métrica foi o objeto de estudo deste trabalho.

Na etapa seguinte, o estudo foi direcionado para a aplicação da Análise de Pontos de Função ao estudo de caso. Para isso, foram elaborados e descritos os passos para a aplicação desta análise, bem como, apresentação dos procedimentos da aplicação da Análise dos Pontos de Função ao estudo de caso, terminando com a demonstração dos resultados e aplicação da estimativa de prazos e custos.

Por fim, foram elaborados o artigo e as considerações finais a respeito do trabalho desenvolvido.

2 MÉTRICAS DE *SOFTWARE*

Neste capítulo, apresenta-se a definição de Métricas de *Software* e sua importância para o planejamento e acompanhamento de projetos de desenvolvimento de *software*.

2.1 Contextualização

Atualmente, a busca pelo desenvolvimento de *software* tem crescido constantemente e, devido a isto, a demanda é muitas vezes maior do que a própria oferta. Diante desse crescimento, algumas empresas procuram satisfazer essa demanda em vários aspectos, como: garantia de qualidade, que pode envolver testes, revisões, gerenciamento de configuração, estimativas, entre outros (BARTIÉ, 2002).

No entanto, ainda é possível encontrar problemas no decorrer do desenvolvimento de *software*, os quais se estendem há mais de trinta anos, tais como, o não cumprimento dos prazos e custos. Como justificativa, pode-se citar a ausência da cultura, conhecimento e prática do uso de métricas que podem controlar esses aspectos. Nesse contexto, há vários estudos que relatam a importância de um controle efetivo para os problemas citados. Assim, destaca-se o uso de métricas adaptado ao processo de desenvolvimento (PAULA FILHO, 2003).

Métricas de *Software* é um método de medição usado para obter a avaliação quantitativa de um produto ou processo (PRESSMAN, 2001). Tem como finalidade principal avaliar o andamento dos projetos, comparando os dados estimados com os valores medidos, permitindo assim, agir quando necessário, ou seja, tomar uma decisão em relação à melhoria do que está sendo analisado através dos resultados

obtidos com a coleta da métrica utilizada. Através do uso de métricas, é possível constatar possíveis equívocos nas estimativas, bem como, aspectos que necessitem de revisão enquanto projeto. Ainda, seu uso contribui para melhorar a gerência de projetos, identificar as melhores práticas para o desenvolvimento de *software*, avaliar melhor o próprio produto e, ainda, permite um melhor entendimento do processo de desenvolvimento do produto (CAMPÊLO, 2002).

Em resumo, métrica nada mais é do que um indicador². Dentre os indicadores mais conhecidos, podem-se destacar os indicadores de desempenho financeiro, ou seja, o custo gerado por tal projeto, além dos indicadores de qualidade e produtividade, sendo que este último diz respeito à duração do desenvolvimento do projeto em questão. Dentre os indicadores citados, a maior preocupação em termos de desenvolvimento, enquanto gerentes de projetos e clientes é a precisão em determinar custos e prazos envolvidos no projeto em questão.

No entanto, embora sejam conhecidas apenas algumas Métricas de *Software*, ressalta-se que estas podem ser utilizadas em qualquer aspecto do processo que precisa ser medido e controlado, buscando assim atingir objetivos dentro do planejamento, evitando imprevistos como: abandono de projetos, custos excedentes, prazos ultrapassados.

Com o auxílio das Métricas de *Software*, que são atividades capazes de gerar dados para serem analisados e utilizados como apoio pelos gerentes de projetos de *software*, pode ser possível obter o sucesso desejado no desenvolvimento.

2.2 Métricas de *software*

As Métricas de *Software* podem ser entendidas como medições de alguma propriedade do *software*, ou seja, cálculos matemáticos através dos quais se obtém um resultado que posteriormente será interpretado, para que este possa levar a uma possível tomada de decisão referente ao que foi medido. Estes resultados podem ser caracterizados como indicadores, os quais visam interferir positivamente no que está

² Índice (valor) que fornece uma orientação ou indicação prática de uma determinada situação.

sendo medido. Ainda, métricas buscam quantificar respostas para perguntas que têm como meta verificar o cumprimento de objetivos estabelecidos inicialmente (VAZQUEZ, SIMÕES e ALBERT, 2006).

Segundo GOMES³ (*apud* GRANDCHAMP, 2002, p. 32),

O termo Métrica de *Software* refere-se à mensuração dos indicadores quantitativos do tamanho e complexidade de um sistema. Esses indicadores são, por sua vez, utilizados para confrontar os desempenhos observados no passado, a fim de derivar previsões de desempenho futuro.

Baseado na definição exposta pode-se considerar que a aplicação de métricas pode proporcionar um melhor entendimento do processo de desenvolvimento, avaliação do desempenho do projeto, avaliação da qualidade do produto final, tomada de decisões mais precisas através dos indicadores obtidos, identificação e resolução de problemas, identificação e gerência de riscos.

No entanto, o uso de Métricas de *Software*, de certa forma, propõe algumas sugestões, tais como (PRESSMAN, 2001):

- Bom senso ao interpretar as métricas, ou seja, sensibilidade organizacional e conhecimento na interpretação dos dados obtidos para que a interpretação realizada possa realmente ser eficaz neste processo;
- Fornecimento de um *feedback* à equipe;
- Não utilizá-las para avaliar pessoas, pois neste caso as métricas usadas podem refletir como uma ameaça aos colaboradores de uma determinada empresa, ou seja, um desconforto ao saber que estão sendo analisados;
- Utilização de mais de uma métrica nos indicadores, para que se tenha alguma certeza do resultado obtido; definição de valores de referência (por exemplo, projetos anteriores), os quais servem de base à avaliação.

Destaca-se, ainda, que as Métricas de *Software* destinam-se tanto a medir atributos específicos de produtos de *software*, como atributos do processo de desenvolvimento de *software*.

³ GOMES, A. E. Métricas e Estimativas de Software: O Início de um Rallye. Developers Cio Magazine, Rio de Janeiro, ano 4 n. 39, p. 50-53, nov. 1999.

As Métricas de *Software* também oferecem algumas finalidades como (PRESSMAN, 2001):

- Funções de coleta de dados para avaliação e desempenho atribuindo estas responsabilidades a toda equipe envolvida no projeto;
- Reunir dados de desempenho pertencentes à complementação do *software*;
- Analisar os históricos dos projetos anteriores para determinar o efeito desses fatores, prevenindo erros para projetos futuros.

Assim, Métricas de *Software* podem ser consideradas um meio para prever custo, qualidade e pesquisa (PAULA FILHO, 2003).

Contudo, pode-se concluir que métricas destinam-se a medir *software*, tanto no produto, quanto em seu desenvolvimento e que, o resultado de uma medição permite a elaboração de uma interpretação, a qual, possivelmente, pode facilitar uma comparação entre os resultados existentes e o que foi obtido com a aplicação da métrica. A partir da comparação dos resultados obtidos é possível a tomada de decisão, quando necessária, para melhorias de algum aspecto que está sendo analisado. Como exemplo de métricas cita-se Análise de Pontos de Função, *Constructive Cost Model* (COCOMO), Pontos por Caso de Uso, entre outras.

Dentro deste contexto, é importante destacar que muitos autores utilizam o termo “métrica” para caracterizar Análise de Pontos de Função, COCOMO, Pontos por Caso de Uso, entretanto, entende-se que estes citados são na verdade técnicas para coletar determinadas métricas, porém não há problema em se referir a essas técnicas como métricas, visto que são utilizadas para coletar métricas, como mencionado.

Finalizando, ressalta-se que, embora as métricas possam ser utilizadas e aplicadas por algum membro da equipe de desenvolvimento, responsável por tal atividade, normalmente é de responsabilidade do gerente de projetos de *software*, interpretar os resultados obtidos pela mesma, bem como a tomada de decisão, em relação aos aspectos obtidos.

2.3 Tipos de métricas

As Métricas de *Software* podem ser divididas em dois tipos: processo e produto (PRESSMAN, 2001). Este tópico destina-se a definição e características destes dois tipos e visa possibilitar um melhor entendimento e esclarecimento sobre os mesmos.

2.3.1 Métricas do processo

Chama-se de métricas do processo aquelas destinadas à mensuração do processo de desenvolvimento de *software*. Consiste em um conjunto de teorias e práticas relacionadas com as medidas, que serão abordadas adiante. Estas permitem realizar uma estimativa de custo, desempenho e cronograma de um projeto, visto que sua função compreende avaliar o tamanho e o estado do sistema e não sua qualidade em si (SOMMERVILLE, 2003).

São exemplos de métricas do processo: erros cometidos por programadores, pedidos de alteração no *software*, número de relatórios pendentes, tempo, precisão da estimativa de cronograma, esforço, precisão da estimativa de esforço. Isso pode ser medido através da aplicação da Análise de Pontos de Função e geralmente são utilizadas pelos desenvolvedores para (PFLEEGER, 2004):

- Ajudar a fazer uma previsão do tamanho do sistema final;
- Derivar o nível de esforço que um projeto irá precisar
- Determinar se o projeto está em dia com o plano de trabalho previamente determinado.

Finalizando, métricas do processo podem ser consideradas um importante método para garantia do sucesso do projeto em desenvolvimento.

2.3.2 Métricas do produto

Consideram-se métricas do produto aquelas obtidas a partir de características do mesmo ou de artefatos em qualquer estágio de desenvolvimento, como o código-fonte ou uma especificação de requisitos (PFLEEGER, 2004). Destinam-se a prever

a qualidade dos sistemas.

Este tipo de métrica encontra-se nas características de qualidade, externa ou interna. Na visão de qualidade do produto: funcionalidade, confiabilidade, usabilidade, manutenibilidade, eficiência, modularidade, portabilidade (PRESSMAN, 2001).

Geralmente métricas de produto são usadas para fornecer:

- Diretrizes que sugerem ações locais e específicas para melhorar a qualidade de diferentes componentes do sistema;
- Comparações entre sistemas existentes;
- Comparações entre sistemas novos e outros sistemas conhecidos.

Finalizando, ressalta-se que as métricas do produto podem definir as características de qualidade desejáveis para um produto de *software*.

2.4 Medidas diretas e indiretas

Medição ou mensuração é o processo pelo qual números ou símbolos são associados a atributos de entidades no mundo real, com o objetivo de descrevê-la de acordo com um conjunto de regras claramente definidas (PETERS e PEDRYCZ, 2001).

Medida trata-se do valor de uma métrica. Uma medida pode permitir quantificar, dimensionar ou até mesmo definir o tamanho ou capacidade de algum produto ou processo de *software*. Fornece um mecanismo de avaliação objetiva.

As medidas podem ser divididas em medidas diretas e medidas indiretas.

2.4.1 Medidas diretas

Considera-se medida direta, também conhecida como fundamentais ou básicas, a medida realizada em termos de atributos observados (usualmente determinada pela contagem).

Entre as medidas diretas incluem-se custo e esforço aplicados, quantidade de linhas de código produzidas, capacidade de memória, quantidade de erros, quantidade

de páginas de documentação, quantidade de diagramas. No mundo físico, pode-se citar como exemplo o comprimento de um parafuso (PRESSMAN, 2001).

Considera-se que medidas diretas apresentam maior facilidade de obtenção. O mesmo não ocorre com as indiretas tornando-se assim mais complexas de serem obtidas.

A seleção das medidas mais adequadas para cada caso depende muito dos objetivos definidos inicialmente.

Destaca-se que, geralmente as medidas diretas estão diretamente associadas às métricas do processo.

2.4.2 *Medidas indiretas*

Medidas indiretas, também chamadas de derivadas, são obtidas a partir de outras métricas e, por isto, estão diretamente associadas às métricas do produto (PRESSMAN, 2001). Sendo assim, pode-se considerar que este tipo de medida está relacionado às características de qualidade do produto.

Como exemplos de medidas indiretas consideram-se complexidade, eficiência, confiabilidade, facilidade de manutenção, entre outras. Por exemplo, a qualidade dos parafusos produzidos, medida pela contagem dos parafusos rejeitados (PRESSMAN, 2001).

2.5 **Categorização das métricas**

As métricas podem ser divididas nas seguintes categorias (PRESSMAN, 2001):

- a) *Métricas de qualidade*: oferecem uma indicação de quanto o *software* pode adequar-se às exigências implícitas e explícitas do cliente. Exemplo: quantidade de erros por fase;
- b) *Métricas de produtividade*: concentra-se na saída do processo de Engenharia de *Software*. Exemplos: quantidade de casos de uso, iteração;
- c) *Métricas técnicas*: concentram-se nas características do *software* e não no

processo por meio do qual o *software* foi desenvolvido. Exemplos: complexidade lógica e grau de manutenibilidade;

- d) *Métricas orientadas ao tamanho*: são medidas diretas do tamanho dos artefatos de *software* associados ao processo por meio do qual o *software* é desenvolvido. Originam-se pela normalização das medidas de qualidade e/ou produtividade considerando o tamanho do *software* produzido. Pode-se utilizar linhas de código como valor de normalização para desenvolver métricas que podem ser integradas com métricas semelhantes de outros projetos. Exemplos: esforço, custo, quantidades de páginas de documentação, quantidade de erros;
- e) *Métricas orientadas para função*: consiste em um método para medição de *software* do ponto de vista do usuário, determinando de forma consistente o tamanho e a complexidade de um *software*. Pontos de Função são gerados usando uma relação empírica baseada em medidas da contagem de domínio de informações do *software* e na avaliação da complexidade do *software*. Destaca-se a Análise de Pontos e Função como uma das mais utilizadas e indicadas para este caso. Exemplos de Métricas de Pontos de Função: esforço, custo, quantidade de defeitos, quantidade de erros.

Completando, é importante ressaltar que a classificação pode sofrer divergências na literatura e, por isto é possível encontrar classificações diferentes. A classificação apresentada acima teve como base PRESSMAN (2001).

2.6 Conclusão

Através dos conceitos abordados neste capítulo, pode-se entender a importância da integração das métricas ao processo de desenvolvimento de *software*, tendo como justificativa a grande demanda pela qualidade, pontualidade e economia no mesmo.

A abordagem realizada neste referencial teórico descreve os conceitos relacionados às Métricas de *Software*. A categorização das Métricas de *Software*

demonstra sua utilidade e, com isto, pode-se identificar como e onde podem ser aplicadas, além das melhorias que seu uso pode proporcionar no processo de desenvolvimento de sistemas e no produto final.

Concluindo, pode-se considerar que Métricas de *Software* podem alavancar uma nova concepção para o bom desenvolvimento de sistemas, pois procura, de certa forma, garantir um produto final de qualidade e com ausência de imprevistos, como atrasos na entrega e custos excessivos, comuns neste segmento da informática. Ainda, caso ocorram imprevistos, através de uma efetiva utilização de métricas, é possível comparar o histórico de sua aplicação, por meio de uma *baseline*, na tentativa de adequar correções e possíveis melhorias.

3 ANÁLISE DE PONTOS DE FUNÇÃO

No presente capítulo são abordados temas referentes à Análise de Pontos de Função (APF), como definição, características e descrição. Também são apresentados trabalhos correlatos para justificar o uso da APF.

3.1 Definição da APF

A APF é uma métrica para medir o tamanho funcional de um *software* do ponto de vista do usuário. Entende-se como usuário qualquer pessoa ou entidade que interaja com a aplicação ou especifique seus requisitos. Esta métrica surgiu na *International Business Machines* (IBM), no início da década de 70, como uma alternativa às métricas baseadas em linhas de código. No Brasil, seu uso iniciou significativamente na década de 90 (VAZQUEZ, SIMÕES e ALBERT, 2006).

Essa métrica, também conhecida como Contagem de Pontos de Função, é regulamentada pela *International Function Point Users Group* (IFPUG), organização internacional sem fins lucrativos sediada nos Estados Unidos. Esta organização publicou o *Counting Practices Manual* (Manual de Práticas de Contagem), o qual estabelece os padrões para o cálculo dos Pontos de Função (PF) (IFPUG, 2007).

No Brasil, usuários desta métrica fundaram o *Brazilian Function Point Users Group* (BFPUG), que é um grupo constituído com o objetivo de estimular e divulgar a utilização de métricas no desenvolvimento de sistemas, em particular a Análise de Pontos de Função.

Conforme apresentado pelo BFPUG (2007), através da Análise de Pontos de Função podem ser medidos:

- *Projeto de desenvolvimento*: neste caso, a contagem mede a funcionalidade fornecida aos usuários finais do *software* quando da sua primeira instalação;
- *Projeto de melhoria*: neste caso, a contagem mede as funcionalidades adicionadas, modificadas ou excluídas do sistema pelo projeto e também as eventuais funcionalidades de conversão de dados;
- *Projeto de aplicação*: esta contagem mede a funcionalidade fornecida aos usuários por uma aplicação instalada.

3.2 APF versus métricas de processo

Pode-se dizer que a relação existente entre APF e métricas do processo deve-se ao fato de APF utilizar as métricas do processo para medir prazo e custo de um projeto (LEMKE, 2005).

Na APF o processo de definição dos requisitos possui papel fundamental, pois é por meio dele que as diversas necessidades dos usuários são mapeadas para as características e funcionalidades de um *software*, as quais podem ser medidas ou contadas e, de acordo com isto, é possível ressaltar a relação direta das métricas de processo e APF, visto que as medições que podem ser obtidas a partir da APF são derivadas da identificação das funcionalidades.

3.3 Características e propósitos da APF

A métrica possui alguns objetivos, definidos por CALAZANS (2004):

- Medir a funcionalidade que o usuário solicita e recebe;
- Medir o desenvolvimento e manutenção de *software* de forma independente da tecnologia utilizada para sua implementação;
- Melhorar as estimativas de projetos, permitindo prevê-las com maior acurácia em fases iniciais do ciclo de desenvolvimento de sistemas;
- Melhorar a gerência de projetos;

- Identificar as práticas com melhor desempenho visando a sua propagação na empresa;
- Prover uma métrica de medição para apoiar a análise de produtividade e qualidade e custos;
- Prover uma forma de estimar o tamanho do *software*;
- Prover um fator de normalização para a comparação entre *softwares*.

3.4 Estrutura do processo de contagem da APF e elementos envolvidos

O processo de contagem da APF é dividido em várias etapas. A figura 1 apresenta as etapas do Processo de Contagem, ou seja, a estrutura deste processo, bem como suas relações de interdependência e, logo em seguida, é apresentada a descrição de cada elemento, de acordo com VAZQUEZ, SIMÕES e ALBERT (2006).

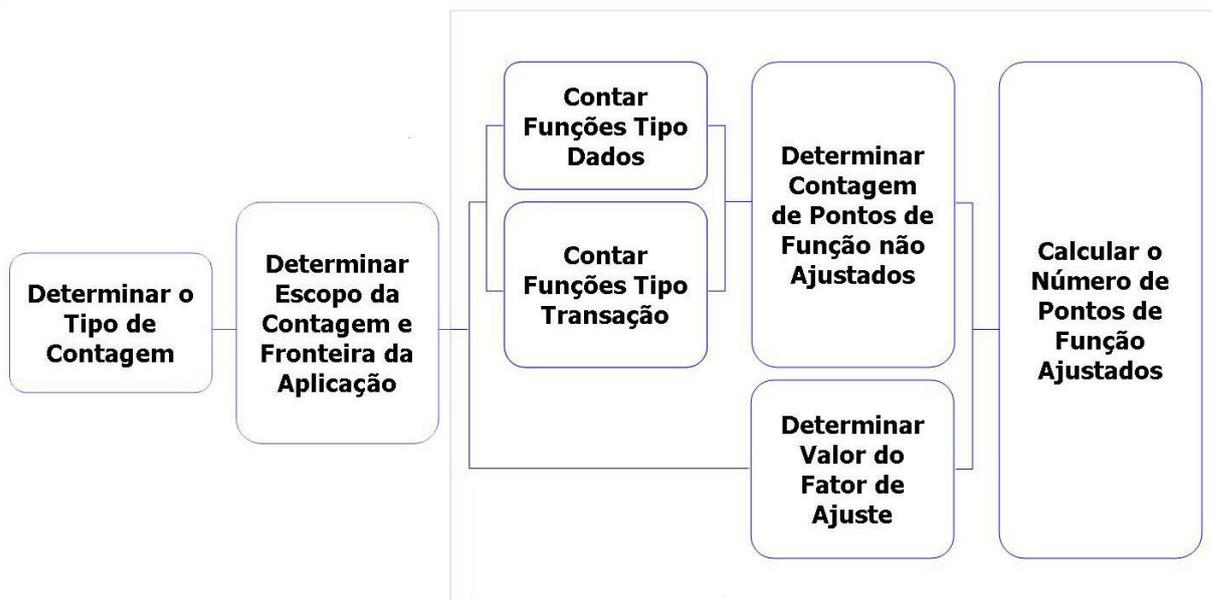


FIGURA 1 - Visão Geral do Processo de Contagem de Pontos de Função (FONTE: VAZQUEZ, SIMÕES e ALBERT, 2006, p. 45).

Conforme mencionado acima, a seguir são descritas as etapas que compõem o processo de contagem e os elementos envolvidos no mesmo, conforme exposto na figura 1.

- Determinação do tipo de contagem

É o primeiro passo do Processo de Contagem de Pontos de Função. Nesta etapa, os responsáveis pela medição estabelecem o tipo de contagem que será empregado para medir o *software*. A contagem pode ser aplicada tanto em projetos quanto em aplicações. É importante destacar que qualquer contagem realizada antes do término do projeto é na verdade uma estimativa da funcionalidade que é entregue ao seu final.

Os três tipos de contagem são os seguintes:

- *Contagem de um projeto de desenvolvimento*: mede a funcionalidade fornecida aos usuários finais do *software* quando da sua primeira instalação;
- *Contagem de um projeto de melhoria*: mede as funcionalidades adicionadas, modificadas ou excluídas do sistema pelo projeto e também as eventuais funcionalidades de conversão de dados;
- *Contagem de uma aplicação*: mede a funcionalidade fornecida aos usuários por uma aplicação instalada.

- Determinação da fronteira de aplicação e o escopo da contagem

Este é o segundo passo do Processo de Contagem de Pontos de Função. A *fronteira da aplicação* é a interface conceitual que delimita o *software* a ser medido e o mundo exterior (seus usuários). O *escopo da contagem* define se a contagem abrange um ou mais sistemas ou apenas parte de um sistema.

Quando há mais de uma aplicação incluída no escopo de uma contagem, várias fronteiras devem ser identificadas. Algumas regras para a determinação da fronteira da aplicação (VAZQUEZ, SIMÕES e ALBERT, 2006):

- Sua determinação deve ser feita baseada no ponto de vista do usuário, ou seja, o foco deve estar no que ele pode entender e descrever;
- A fronteira entre as aplicações deve ser baseada nas funções conforme estabelecido pelos processos de negócio, não em considerações tecnológicas;

- Em projetos de melhorias, a fronteira estabelecida no início do projeto deve estar de acordo com a fronteira já estabelecida para a aplicação sendo modificada.

As dicas abaixo ajudam a identificar a fronteira da aplicação (VAZQUEZ, SIMÕES e ALBERT, 2006):

- Desenhar uma fronteira em volta para destacar quais partes são internas e externas à aplicação e obter uma documentação do fluxo de dados no sistema;
- Verificar como se mantêm os grupos de dados no sistema;
- Identificar áreas funcionais pela atribuição de propriedades de certos objetos de análise, como entidades e processos;
- Comparar os critérios utilizados em outras métricas como esforço, duração, custo e defeitos. As fronteiras para efeito da Análise de Pontos de Função e as outras métricas devem ser as mesmas;
- Verificar como a aplicação é gerenciada; se é desenvolvida ou mantida na sua totalidade por uma equipe distinta;
- Verificar se o *software* possui ordens de serviço específicas e independentes.

A identificação da fronteira da aplicação é considerada um passo essencial para a medição funcional em um processo de contagem, pois a contagem difere totalmente quando são identificadas mais que uma fronteira da aplicação, interferindo diretamente no resultado final da mesma. No entanto, devido à dificuldade em determinar onde uma aplicação termina e outra começa, sugere-se delinear a fronteira através de uma perspectiva de negócio. O posicionamento incorreto da fronteira pode alterar a perspectiva da contagem.

Segundo VAZQUEZ, SIMÕES e ALBERT (2006), complementando, o escopo da contagem pode abranger:

- Todas as funcionalidades disponíveis;

– Apenas as funcionalidades efetivamente usadas pelo usuário; Apenas algumas funcionalidades especificadas (relatórios, transações cadastrais etc.).

- Contagem das funções do tipo dado

Representam as funcionalidades fornecidas pelo sistema ao usuário, para atender as suas necessidades de dados. Classificam-se em:

- *Arquivo Lógico Interno (ALI)*: grupo logicamente relacionado de dados ou informações de controle, mantido dentro da fronteira da aplicação que está sendo contada, e é identificado pelo usuário. Possui como finalidade armazenar dados mantidos por meio de uma ou mais transações da aplicação sendo contada. Exemplo: tabelas do banco de dados atualizadas pela aplicação;
- *Arquivo de Interface Externa (AIE)*: grupo logicamente relacionado de dados ou informações de controle, mantidos fora da fronteira da aplicação sendo contada. Assim como o ALI, também é identificado pelo usuário. Tem como finalidade armazenar dados referenciados por meio de uma ou mais transações da aplicação sendo contada. Exemplo: tabelas de banco de dados lidas pela aplicação, mas atualizadas por outra aplicação.

- Contagem das funções do tipo transação

Representam as funcionalidades de processamento de dados fornecidas pelo sistema ao usuário. Classificam-se em:

- *Entrada Externa (EE)*: é um tipo de transação que processa dados ou informações de controle recebidos de fora da fronteira da aplicação. Tem como finalidade manter um ou mais arquivos lógicos internos e/ou alterar o comportamento do sistema. Exemplo: incluir cliente, alterar cliente, excluir cliente;
- *Saída Externa (SE)*: tipo de transação que envia dados ou informações

de controle para fora da fronteira da aplicação. Tem como finalidade apresentar informação ao usuário através de lógica de processamento que não seja apenas uma simples recuperação de dados ou informações de controle. Seu processamento deve conter cálculo, criar dados derivados, ou manter um arquivo lógico interno, ou alterar o comportamento do sistema. Exemplo: relatório de totais de faturamento de cliente;

- *Consulta Externa* (CE): tipo de transação que envia dados ou informações de controle para fora da fronteira da aplicação. Tem como finalidade apresentar ao usuário através da simples recuperação de dados ou informações de controle de ALI e/ou AIE. Exemplo: consulta cadastro de clientes.

- Determinação dos pontos de função não-ajustados

Os pontos de função não-ajustados consistem na soma dos valores das complexidades das funções do tipo dados e das funções do tipo transação. A complexidade de cada tipo de função possui um valor associado em PF não-ajustados fornecidos por tabelas específicas. Por exemplo, um ALI de complexidade média possui dez PF não-ajustados. Após a identificação e classificação de todas essas funções na contagem, o número de PF não-ajustados será simplesmente a soma do valor de cada uma dessas funções.

- Determinação do fator de ajuste

O fator de ajuste é um valor responsável pela correção das distorções ocorridas na etapa de contagem das funções do tipo dados e funções do tipo transação, é opcional na aplicação da métrica APF. Tem como finalidade ajustar os PF não-ajustados em +-35% de acordo com a influência de 14 características gerais de sistema (CGS) (VAZQUEZ, SIMÕES e ALBERT, 2006):

1. Comunicação de dados;
2. Processamento distribuído;

3. Performance;
4. Configuração altamente utilizada;
5. Volume de transações;
6. Entrada de dados *on-line*;
7. Eficiência do usuário final;
8. Atualização *on-line*;
9. Complexidade de processamento;
10. Reutilização;
11. Facilidade de instalação;
12. Facilidade de operação;
13. Múltiplos locais;
14. Facilidade de mudanças.

- Determinação dos PF ajustados

A determinação dos PF ajustados é obtida através do valor resultante da contagem dos pontos de função não-ajustados agregado ao valor do fator de ajuste.

Cada tipo de contagem (projeto de desenvolvimento, projeto de melhoria e aplicação) possui uma equação específica para a determinação dos PF ajustados.

A figura 2 apresenta os elementos que compõem o Processo de Contagem de PF.

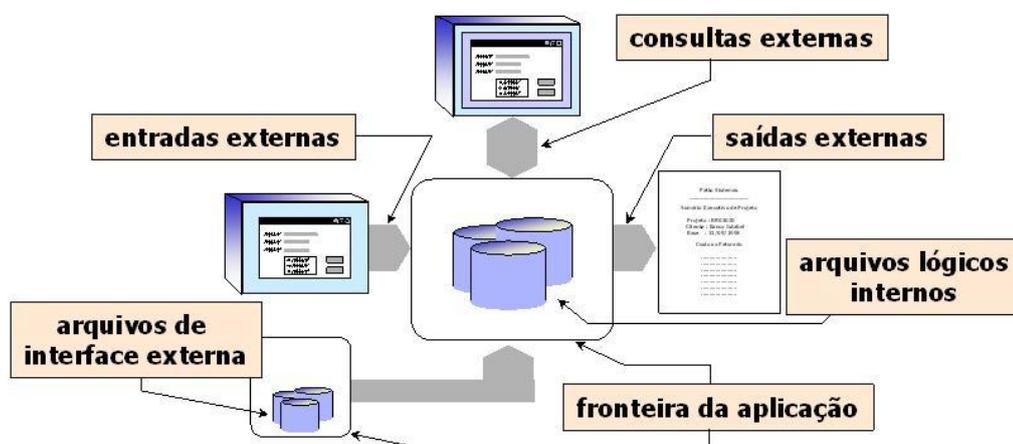


FIGURA 2 - Elementos da Contagem de Pontos de Função
(FONTE: VAZQUEZ, SIMÕES e ALBERT, 2006, p. 46)

Através da figura 2, pode-se ter um entendimento sobre os elementos que compõem o Processo de Contagem de PF. Esses elementos são chamados de tipos de funcionalidades, ou seja, Arquivos Lógicos Internos (ALI) e os Arquivos de Interface Externa (AIE) que compõem as funções do tipo dados; e as Entradas Externas (EE), Saídas Externas (SE) e Consultas Externas (CE) fazem parte das funções do tipo transação e por fim, a fronteira da aplicação que delimita o *software* que está sendo medido e o mundo exterior (seus usuários).

3.5 Processo de contagem

Como mencionado no item 3.1, o Processo de Contagem pode ser aplicado tanto em *softwares* concluídos quanto em projetos em desenvolvimento.

Para efetivar o Processo de Contagem é necessário seguir alguns passos sugeridos na APF. Abaixo segue a seqüência de passos.

3.5.1 Passo 1

O Processo de Contagem inicia-se com a determinação do tipo de contagem, como observado na figura 1.

3.5.2 Passo 2

Identificar o escopo da contagem e a fronteira da aplicação, como apresentado no item 3.4.

3.5.3 Passo 3

Este passo é caracterizado pela contagem dos PF não-ajustados que, combinada com a determinação do valor do fator de ajuste, pode-se obter o cálculo, bem como o resultado dos pontos de função ajustados, objetivo principal do Processo de Contagem.

A contagem dos PF não-ajustados é realizada a partir de dois procedimentos:

a contagem das funções do tipo dados (Arquivo Lógico Interno - ALI; Arquivo de Interface Externa - AIE) e a contagem das funções do tipo transação (Entrada Externa - EE; Saída Externa - SE e Consulta Externa - CE), conforme observado na figura 2.

A contagem destas funções exige a aplicação de um conjunto de etapas e regras, referente a cada um dos elementos. Pode-se considerar um dos passos mais complexos do projeto, pois exige grande atenção da pessoa responsável pela contagem.

3.5.3.1 Contagem das funções do tipo dados

Duas regras básicas são exigidas para a contagem das funções de dados (VAVASSORI, 2002):

- Regras de Identificação
 - Identificação de ALI
 - Identificação de AIE
- Regras de Complexidade
 - Identificação dos tipos de registros
 - Identificação dos tipos de dados

- Regras de Identificação

De acordo com BRAGA⁴ (*apud* VAVASSORI, 2002), para que uma função seja determinada como ALI todas as regras seguintes devem ser válidas:

- O grupo de dados ou informação de controle é identificável pelo usuário como sendo requisito da aplicação;
- O grupo de dados é mantido dentro da fronteira da aplicação sendo contada.

No que diz respeito aos AIE, deve-se verificar a validade das regras:

- O grupo de dados ou informação de controle é identificável pelo

⁴ BRAGA, A. Análise de Pontos de Função. IBPI Press, 1996. ISBN 85-733-1002-2

usuário como sendo um requisito da aplicação;

- O grupo de dados é referenciado pela aplicação sendo contada, porém é externo a ela;
- O grupo de dados não é mantido pela aplicação que está sendo contada;
- O grupo de dados é mantido em um ALI de outra aplicação.

A complexidade é classificada com base no número de tipos de dados e no número de tipos de registros e é fornecida pela tabela 1.

Um *tipo dado* é um campo único, reconhecido pelo usuário, não repetido, já um *tipo registro*, é um subgrupo de dados, reconhecido pelo usuário, componente de um ALI ou AIE.

De acordo com VAZQUEZ, SIMÕES e ALBERT (2006), existem regras bem definidas encontradas no anexo 1, que determinam quais elementos devem ser contabilizados como tipo de dados e tipo de registros.

TABELA 1 - Tabela de complexidade funcional dos ALI e AIE

		<i>Tipos de Dados</i>		
		< 20	20 – 50	> 50
<i>Tipo de Registros</i>	1	Baixa	Baixa	Média
	2 – 5	Baixa	Média	Alta
	>5	Média	Alta	Alta

(FONTE: VAZQUEZ, SIMÕES e ALBERT, 2006, p. 77)

Por exemplo, um ALI contendo 45 tipos de dados e um tipo de registro é de complexidade baixa e outro AIE contendo 55 tipos de dados e um tipo de registro é de complexidade média, como se pode observar na tabela 1.

Após a determinação da complexidade dos arquivos, deve-se calcular sua contribuição de acordo com a tabela 2.

TABELA 2 - Tabela de contribuição dos PF não-ajustados das funções do tipo dado

<i>Tipo de Função</i>	<i>Baixa</i>	<i>Média</i>	<i>Alta</i>
Arquivo lógico interno	7 PF	10 PF	15 PF
Arquivo de interface interna	5 PF	7 PF	10 PF

(FONTE: VAZQUEZ, SIMÕES e ALBERT, 2006, p. 80)

Por exemplo, um ALI de complexidade alta contribui com 15 PF não-ajustados e um AIE também de complexidade alta contribui com 10 PF não-ajustados como se pode verificar na tabela 2.

3.5.3.2 Contagem das funções do tipo transação

O passo seguinte após a contagem das funções do tipo dados é contar as funções do tipo transação: EE, SE e CE. A contagem destas é baseada no número de *arquivos referenciados*, que são um ALI ou um AIE e *tipos de dados*, que são campos únicos reconhecidos pelo usuário e não repetidos. Assim como as funções do tipo dados, as funções do tipo transação também possuem regras bem definidas para determinar os seus elementos (EE, SE e CE), pois estes fazem parte do cálculo dos PF. As regras consideradas neste trabalho são encontradas em VAZQUEZ, SIMÕES e ALBERT (2006) e podem ser visualizadas no anexo 1.

Após a determinação da quantidade de arquivos referenciados e tipos de dados utilizam-se as tabelas 3 e 4 para a classificação da complexidade das EE, SE e CE.

TABELA 3 - Tabela de complexidade funcional para EE

		<i>Tipos de Dados</i>		
		< 5	5 – 15	> 15
<i>Arquivos Referenciados</i>	< 2	Baixa	Baixa	Média
	2	Baixa	Média	Alta
	>2	Média	Alta	Alta

(FONTE: VAZQUEZ, SIMÕES e ALBERT, 2006, p. 109)

TABELA 4 - Tabela de complexidade funcional para SE e CE

		<i>Tipos de Dados</i>		
		< 6	6 – 19	> 19
<i>Arquivos Referenciados</i>	< 2	Baixa	Baixa	Média
	2 - 3	Baixa	Média	Alta
	>3	Média	Alta	Alta

(FONTE: VAZQUEZ, SIMÕES e ALBERT, 2006, p. 109)

Como exemplo, uma EE com 16 tipos de dados e 2 arquivos referenciados é de complexidade alta, como mostra a tabela 3. Já uma SE com 16 tipos de dados e 2 arquivos referenciados é de complexidade média e uma CE com 19 tipos de dados e um arquivo referenciado é de complexidade baixa, como observado na tabela 4.

Após a determinação da complexidade das funções do tipo transação deve-se determinar a sua contribuição utilizando a tabela 5.

TABELA 5 - Tabela de contribuição dos PF não-ajustados das funções do tipo transação

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

(FONTE: VAZQUEZ, SIMÕES e ALBERT, 2006, p. 112)

Exemplos: De acordo com a tabela 5, uma EE de complexidade alta contribui com 6 PF não-ajustados, uma SE também de complexidade alta contribui com 7 PF não-ajustados e uma CE também de complexidade alta contribui com 6 PF não-ajustados.

3.5.4 Passo 4

Depois de identificadas a complexidade e a contribuição de cada tipo de função, reúnem-se esses valores na tabela 6, na qual o número de funções de determinada complexidade é multiplicado pelo valor de sua contribuição.

TABELA 6 - Tabela para o cálculo dos PF não-ajustados ou brutos

<i>Tipo de função</i>	<i>Complexidade funcional</i>	<i>Total por complexidade</i>	<i>Total tipo função</i>
<i>Funções do tipo dados</i>			
ALI	Número funções baixa * 7		
	Número funções média * 10		
	Número funções alta * 15		
AIE	Número funções baixa * 5		
	Número funções média * 7		
	Número funções alta * 10		
<i>Funções do tipo transação</i>			
EE	Número funções baixa * 3		
	Número funções média * 4		
	Número funções alta * 6		
SE	Número funções baixa * 4		
	Número funções média * 5		
	Número funções alta * 7		
CE	Número funções baixa * 3		
	Número funções média * 4		
	Número funções alta * 6		
Total de PF não-ajustados			

(FONTE: adaptado de DIAS, 2003)

Posteriormente, para preenchimento da tabela 6, somam-se os valores para

obter o total, que corresponde aos PF não-ajustados ou brutos.

3.5.5 Passo 5

Prosseguindo, calcula-se o valor do fator de ajuste, que é baseado na influência de 14 CGS, citadas no item 3.4. O valor do fator de ajuste influencia os PF não-ajustados em +/- 35% para obterem-se os PF ajustados.

Assim, verifica-se o nível de influência de cada uma das 14 CGS, variando de 0 a 5, de acordo com o quadro 1.

QUADRO 1 - Escala de nível de influência das CGS

0 - Nenhuma Influência
1 - Influência Mínima
2 - Influência Moderada
3 - Influência Média
4 - Influência Significativa
5 - Grande Influência

(FONTE: HAZAN, 2000)

Após, soma-se os níveis de influência de cada uma das 14 CGS para obtenção do somatório dos níveis de influência (TDI).

Por fim, emprega-se a equação 1 (VAZQUEZ, SIMÕES e ALBERT, 2006):

$$\mathbf{VAF = (TDI * 0,01) + 0,65} \qquad \mathbf{Eq.1}$$

Onde,

- TDI: Corresponde ao somatório dos níveis de influência;
- VAF: Valor do fator de ajuste.

3.5.6 Passo 6

O último passo do Processo de Contagem de PF abrange o cálculo final para os três tipos de contagem: projeto de desenvolvimento, projeto de melhoria e aplicação, tratados no item 3.4. Neste cálculo, considera-se o valor dos pontos de função não-ajustados e o fator de ajuste, obtidos através dos cálculos apresentados anteriormente.

3.5.6.1 Projeto de desenvolvimento

O cálculo consiste em três componentes:

- Funcionalidade da aplicação requisitada pelo usuário para o projeto: corresponde às funções utilizadas após a instalação do *software* para atender as necessidades do usuário;
- Funcionalidade de conversão requisitada pelo usuário para o projeto: corresponde às funções disponíveis no momento da instalação da aplicação para converter dados ou fornecer outros requisitos de conversão especificados pelo usuário. Após a instalação essas funções são descartadas;
- Valor do fator de ajuste da aplicação: corresponde ao valor obtido conforme descrição anterior.

Utiliza-se a equação 2:

$$\mathbf{DFP = (UFP + CFP) * VAF} \qquad \mathbf{Eq.2}$$

Onde,

- DFP: Número de PF do projeto de desenvolvimento;
- UFP: Número de PF não-ajustados das funções disponíveis após instalação;
- CFP: Número de PF não-ajustados das funções de conversão;
- VAF: Valor do fator de ajuste.

3.5.6.2 Projeto de aplicação

Estabelece a contagem inicial de PF para uma aplicação. Reflete as novas funcionalidades recebidas pelo usuário, após um Projeto de Desenvolvimento, Não inclui requisitos de conversão. Utiliza-se a equação 3 (VAVASSORI, 2002).

$$\mathbf{PFA = PF_NÃO_AJUSTADO * VAF} \qquad \mathbf{Eq.3}$$

Onde,

- PFA: Número de PF do projeto de aplicação;
- VAF: Valor do fator de ajuste.

3.5.6.3 Projeto de melhoria

Este tipo de contagem implica um sistema já desenvolvido, cujo objetivo consiste em quantificar um projeto de manutenção. Utiliza-se a equação 4 (VAVASSORI, 2002).

$$\mathbf{PFM = [(INC + ALT + PFC) * FAD] + (EXC * FAA)} \quad \mathbf{Eq.4}$$

Onde,

- PFM: Número de PF do projeto de melhoria;
- INC: PF brutos que foram incluídos na aplicação pelo projeto de melhoria. Refere-se as funções que foram adicionadas à aplicação;
- ALT: PF que foram alterados na aplicação pelo projeto de melhoria. Refere-se as funções que sofreram alteração;
- PFC: São os PF que foram adicionados pelo processo de conversão;
- FAD: Fator de ajuste da aplicação depois do projeto de melhoria;
- EXC: PF brutos que foram excluídos da aplicação pelo projeto de melhoria. Refletem as funções que foram excluídas da aplicação;
- FAA: Fator de ajuste da aplicação antes do projeto de melhoria.

3.6 Trabalhos correlatos

Esta seção aborda alguns trabalhos correlatos onde os resultados dos mesmos justificam a aplicação da APF.

3.6.1 *Medição de PF a partir da especificação de requisitos*

TAVARES, CARVALHO e CASTRO (2002) apresentam uma proposta para medição de PF a partir da especificação de requisitos mostradas em casos de uso, notação *Unified Modeling Language* (UML). O objetivo deste trabalho é enfatizar a importância da especificação de requisitos para o trabalho de medição de *softwares*, diminuindo o esforço dos gerentes de projeto, obtendo assim o porte do projeto em termos de PF de maneira simples e dinâmica.

De acordo com o relato deste trabalho, pode-se concluir que para o alcance do sucesso no gerenciamento e controle dos projetos, a especificação de requisitos, a notação UML e a medição de PF podem ser utilizadas juntamente. APF pode ser aplicada ao método de casos de uso.

No entanto, um dos problemas que podem interferir no cálculo utilizado para encontrar o valor dos pontos de função é a identificação dos requisitos, ou seja, funcionalidades do sistema, pois a forma como tais requisitos estão definidos pode ocasionar numa certa inconsistência desses requisitos. Uma proposta apresentada para prevenir tal problema é a adoção de ambos os métodos: caso de uso e APF. Com a nova proposta pode-se considerar que a qualidade do documento de requisitos pode ser melhorada, conseqüentemente a medição e a estimativa como um todo são melhoradas.

3.6.2 *Análise e melhoria de um processo de estimativas de tamanho de projetos de software*

HAZAN e STAA (2004) têm como objetivo analisar um processo de estimativas do Serviço Federal de Processamento de Dados (SERPRO) e implantar melhorias através de uma metodologia de estimativas de tamanho de projetos de desenvolvimento e de manutenção evolutiva de *software*, inserindo os conceitos da Análise de Pontos de Função ao conhecimento inicial do domínio do projeto.

Para realizar o estudo foi feito uma análise de métodos utilizados para

geração de estimativas de tamanho de projetos de *software* baseadas em PF, a partir de uma perspectiva baseada em *Capability Maturity Model Integration* (CMMI). Apresenta também, um processo simplificado para a geração e a documentação de estimativas de tamanho, esforço, custo, cronograma e recursos computacionais críticos.

A elaboração do plano de projeto de *software* tem como base as estimativas. Assim, torna-se fundamental a existência de um procedimento para a geração e documentação das estimativas a serem utilizadas no acompanhamento do projeto de *software*. A definição do procedimento de estimativas descrito no trabalho significou um passo importante na implantação da área chave de planejamento de projeto no SERPRO. Vale ressaltar que o modelo CMMI não apresenta procedimentos para o “Como fazer”, apenas define o que “que deve ser feito”. Este projeto se preocupa em estabelecer o “Como fazer” para aplicar as práticas associadas às estimativas previstas no CMMI.

3.6.3 Roteiro para aplicação de métricas e qualidade em subcontratação de *software*

CHAVES (2003) apresenta, a partir da apuração dos resultados dos PF não-ajustados e a da realização dos testes de caixa preta, um roteiro para identificação da medida e qualidade de *softwares* terceirizados.

Considerando que a Prefeitura de Belo Horizonte tem adotado o critério de subcontratação de serviços de *software*, como ocorreu no Sistema Integrado de Gestão de Materiais, é possível afirmar, com base na simulação apresentada no trabalho, que o roteiro sugerido para medida e avaliação da qualidade do sistema, poderá ser utilizado como recurso pelos técnicos e/ou gerentes responsáveis pela implementação, na avaliação do produto, na sua contratação e durante o ciclo de vida; contribuindo para que os valores despendidos sejam os mais justos, tanto para o fornecedor quanto para a administração pública e que seja garantido um nível de qualidade no mínimo satisfatório que justifique o seu investimento.

3.7 Conclusão

Este capítulo apresentou a definição da Análise de Pontos de Função, bem como a descrição do processo de contagem e seus elementos envolvidos. Além disso, procurou-se aprofundar e proporcionar o conhecimento adequado sobre esta métrica.

O assunto abordado é a base do trabalho, visto que o estudo de caso apresentado no capítulo 4 é baseado nesta técnica de medição.

Com a abordagem dos trabalhos correlatos, procurou-se esclarecer e demonstrar as aplicações da Análise de Pontos de Função, justificando sua utilização.

Assim, pretende-se proporcionar, no decorrer do trabalho, um instrumento de apoio para o entendimento da métrica da Análise de Pontos de Função, bem como sua aplicação.

4 DEMONSTRAÇÃO DA APLICAÇÃO DOS PASSOS DA APF NO ESTUDO DE CASO

Este capítulo trata da demonstração da aplicação da APF no estudo de caso selecionado, bem como a aplicação da estimativa de prazos e custos baseada no resultado em PF.

4.1 Estudo de caso

O estudo de caso deste projeto consiste em um sistema com o objetivo de registrar e gerar informações que precisam constar nos diplomas emitidos aos formandos dos cursos da Universidade do Planalto Catarinense (UNIPLAC). O estudo de caso foi realizado por Scoppel (SCOPPEL, 2004) em uma situação real. O setor beneficiado com a utilização do sistema é o Setor de Registro e Expedição de Diplomas e Certificados (SERED).

As funcionalidades do sistema são (SCOPPEL, 2004):

- Manter cadastro de aluno
 - Inclusão de aluno;
 - Alterações no cadastro de aluno;
 - Exclusões no cadastro de aluno;
 - Consulta aos dados do aluno.
- Manter cadastro de autorização
 - Inclusão de autorização;
 - Alterações no cadastro de autorização;
 - Exclusões no cadastro de autorização;

- Consulta aos dados do cadastro de autorização.
- Manter cadastro de categoria de curso de graduação
 - Inclusão de categoria de curso de graduação;
 - Alterações no cadastro de categoria de curso de graduação;
 - Exclusões no cadastro de categoria de curso de graduação;
 - Consulta aos dados do cadastro de categoria de curso de graduação.
- Manter cadastro de chefe do SERED
 - Inclusão de chefe do SERED;
 - Alterações no cadastro de chefe do SERED;
 - Exclusões no cadastro de chefe do SERED;
 - Consulta aos dados do cadastro de chefe do SERED.
- Manter cadastro de curso de graduação
 - Inclusão de curso de graduação;
 - Alterações no cadastro de graduação;
 - Exclusões no cadastro de graduação;
 - Consulta aos dados de graduação.
- Manter cadastro de habilitação de curso de graduação
 - Inclusão de habilitação de curso de graduação;
 - Alterações no cadastro de habilitação de curso de graduação;
 - Exclusões no cadastro de habilitação de curso de graduação;
 - Consulta aos dados do cadastro de habilitação de curso de graduação.
- Manter cadastro de pró-reitor de ensino
 - Inclusão de pró-reitor de ensino;
 - Alterações no cadastro de pró-reitor de ensino;
 - Exclusões no cadastro de pró-reitor de ensino;
 - Consulta aos dados do pró-reitor de ensino.
- Manter cadastro de reconhecimento

- Inclusão de reconhecimento;
- Alterações no cadastro de reconhecimento;
- Exclusões no cadastro de reconhecimento;
- Consulta aos dados do cadastro de reconhecimento.
- Manter cadastro de registro
 - Inclusão de registro;
 - Alterações no cadastro de registro;
 - Exclusões no cadastro de registro;
 - Consulta aos dados do cadastro de registro.
- Manter cadastro de reitor
 - Inclusão de reitor;
 - Alterações no cadastro de reitor;
 - Exclusões no cadastro de reitor;
 - Consulta aos dados do reitor.
- Manter cadastro de resolução de curso de graduação
 - Inclusão de resolução de curso de graduação;
 - Alterações no cadastro de resolução de curso de graduação;
 - Exclusões no cadastro de resolução de curso de graduação;
 - Consulta aos dados do cadastro de resolução de curso de graduação.
- Manter cadastro de secretário acadêmico
 - Inclusão de secretário acadêmico;
 - Alterações no cadastro de secretário acadêmico;
 - Exclusões no cadastro de secretário acadêmico;
 - Consulta aos dados do secretário acadêmico.
- Gerar diploma

As descrições das funcionalidades podem ser visualizadas no anexo 2 de acordo com o trabalho de SCOPPEL (2004). Destaca-se que neste caso, entendem-se

como arquivos os conceitos aos quais estão associadas às operações cadastrais conforme WAZLAWICK (2004).

De acordo com as informações descritas, os arquivos considerados na contagem são: Aluno, Autorização, Categoria, Chefe do SERED, Curso de Graduação, Habilitação, Pró-Reitor de Ensino, Reconhecimento, Registro, Reitor, Resolução, Secretário Acadêmico, contendo os dados descritos nos quadros 2 a 13 (SCOPPEL, 2004).

O quadro 2 apresenta as informações do arquivo aluno.

QUADRO 2 - Informações do arquivo aluno

<i>Aluno</i>	
<i>Nome do Campo</i>	<i>Descrição</i>
nomeAluno	Nome do aluno
RG	RG do aluno
Naturalidade	Naturalidade do aluno
nomeCursoGrad	Curso de graduação cursado pelo aluno
Nacionalidade	Nacionalidade do aluno
dataConclusao	Data de conclusão do curso de graduação cursado pelo aluno
dataNascimento	Data de nascimento do aluno
sexoAluno	Sexo do aluno
dataEncerramentoProcesso	Data de encerramento do processo do aluno
numProcesso	Número do processo do aluno

O quadro 3 apresenta as informações do arquivo autorização.

QUADRO 3 - Informações do arquivo autorização

<i>Autorização</i>	
<i>Nome do Campo</i>	<i>Descrição</i>
NumAutorizacao	Número da autorização
OrgaoEmissor	Órgão Emissor da autorização
Data	Data de emissão da autorização

O quadro 4 apresenta as informações do arquivo categoria.

QUADRO 4 - Informações do arquivo categoria

<i>Categoria</i>	
<i>Nome do Campo</i>	<i>Descrição</i>
nomeCategoriaCursoGrad	Nome da categoria

O quadro 5 apresenta as informações do arquivo chefe do SERED.

QUADRO 5 - Informações do arquivo chefe do SERED

<i>Chefe do SERED</i>	
<i>Nome do Campo</i>	<i>Descrição</i>
NomeCSERED	Nome do chefe do SERED
CargoCSERED	Cargo do chefe do SERED
SexoCSERED	Sexo do chefe do SERED
DataInicialCSERED	Data inicial da posse do chefe do SERED
DataFinalCSERED	Data final da posse do chefe do SERED

O quadro 6 apresenta as informações do arquivo curso de graduação.

QUADRO 6 - Informações do arquivo curso de graduação

<i>Curso de Graduação</i>	
<i>Nome do Campo</i>	<i>Descrição</i>
nomeCursoGrad	Nome do curso de graduação
nomeCategoriaCursoGrad	Categoria a que pertence o curso de graduação
nomeHabilitação	Habilitação do curso de graduação
nomeAutorização	Autorização para abertura do curso de graduação
reconhecimentoProvisorio	Reconhecimento provisório do curso de graduação
reconhecimentoOficial	Reconhecimento oficial do curso de graduação
numResolução	Número da resolução a que pertence o curso de graduação

O quadro 7 apresenta as informações do arquivo habilitação.

QUADRO 7 - Informações do arquivo habilitação

<i>Habilitação</i>	
<i>Nome do Campo</i>	<i>Descrição</i>
nomeHabilitacao	Nome da habilitação do curso de graduação

O quadro 8 apresenta as informações do arquivo pró-reitor de ensino.

QUADRO 8 - Informações do arquivo pró-reitor de ensino

<i>Pró-Reitor de Ensino</i>	
<i>Nome do Campo</i>	<i>Descrição</i>
nomePRENS	Nome do pró-reitor de ensino
cargoPRENS	Cargo do pró-reitor de ensino
sexoPRENS	Sexo do pró-reitor de ensino
dataInicialPRENS	Data inicial da posse do pró-reitor de ensino
dataFinalPRENS	Data final da posse do pró-reitor de ensino

O quadro 9 apresenta as informações do arquivo reconhecimento.

QUADRO 9 - Informações do arquivo reconhecimento

<i>Reconhecimento</i>	
<i>Nome do Campo</i>	<i>Descrição</i>
numReconhecimento1	Número do reconhecimento
OrgaoEmissor	Órgão emissor do reconhecimento
Data	Data de emissão do reconhecimento

Tipo	Tipo de reconhecimento
------	------------------------

O quadro 10 apresenta as informações do arquivo registro.

QUADRO 10 - Informações do arquivo registro

<i>Registro</i>	
<i>Nome do Campo</i>	<i>Descrição</i>
numFolha	Número da folha em que consta o registro
numLivro	Número do livro em que consta registro
dataRegistro	Data de emissão do registro
numRegistro	Número do registro

O quadro 11 apresenta as informações do arquivo reitor.

QUADRO 11 - Informações do arquivo reitor

<i>Reitor</i>	
<i>Nome do Campo</i>	<i>Descrição</i>
NomeReitor	Nome do reitor
CargoReitor	Cargo do reitor
SexoReitor	Sexo do reitor
dataInicialReitor	Data inicial da posse do reitor
dataFinalReitor	Data final da posse do reitor

O quadro 12 apresenta as informações do arquivo resolução.

QUADRO 12 - Informações do arquivo resolução

<i>Resolução</i>	
<i>Nome do Campo</i>	<i>Descrição</i>
NumResolucao	Número da resolução
OrgaoEmissor	Órgão emissor da resolução
Data	Data de emissão da resolução

O quadro 13 apresenta as informações do arquivo secretário acadêmico.

QUADRO 13 - Informações do arquivo secretário acadêmico

<i>Secretário Acadêmico</i>	
<i>Nome do Campo</i>	<i>Descrição</i>
nomeSA	Nome do secretário acadêmico
cargoSA	Cargo do secretário acadêmico
sexoSA	Sexo do secretário acadêmico
dataInicialSA	Data inicial da posse do secretário acadêmico
dataFinalSA	Data final da posse secretário acadêmico

Considerando todas as funcionalidades e dados necessários descritos acima, pode-se iniciar o processo de contagem.

4.2 Demonstração

De acordo com os passos apresentados para a aplicação da APF no capítulo 3, apresenta-se o procedimento desta aplicação, representado através do estudo de caso.

4.2.1 Determinação do tipo de contagem

O tipo de contagem utilizado para a medição dos PF, neste estudo de caso, é a contagem do projeto de aplicação, pois toda a fase de projeto do sistema já está concluída.

4.2.2 Identificação da fronteira da aplicação e do escopo da contagem

O sistema não apresenta entradas fora de sua fronteira, pois todas as informações são centralizadas em um único sistema, em um determinado ambiente de trabalho.

Assim, o escopo da contagem abrange apenas uma aplicação e todas as suas funcionalidades.

4.2.3 Contagem dos PF não-ajustados

Dando continuidade a contagem, identificam-se e classificam-se todas as funções envolvidas na contagem. Posteriormente, o número de PF não-ajustados é obtido somando-se o valor de cada uma das funções.

4.2.3.1 Contagem das funções do tipo dados

- a) *Arquivos Lógicos Internos (ALI)*: de acordo com a descrição dos ALI no item 3.4, a tabela 7 mostra a contagem dos ALI envolvidos na aplicação descrita.

TABELA 7 - Identificação dos Arquivos Lógicos Internos (ALI)

<i>Arquivo Lógico Interno (ALI)</i>	<i>Tipos de Dados</i>	<i>Tipos de Registros</i>	<i>Grau da Função</i>
Manter cadastro de aluno	9	1	Baixa
Manter cadastro de autorização	3	1	Baixa

Manter cadastro de categoria de curso de graduação	1	1	Baixa
Manter cadastro de chefe do SERED	5	1	Baixa
Manter cadastro de curso de graduação	3	1	Baixa
Manter cadastro de habilitação de curso de graduação	1	1	Baixa
Manter cadastro de pró-reitor de ensino	5	1	Baixa
Manter cadastro de reconhecimento	4	1	Baixa
Manter cadastro de registro	4	1	Baixa
Manter cadastro de reitor	5	1	Baixa
Manter cadastro de resolução de curso de graduação	3	1	Baixa
Manter cadastro de secretário acadêmico	5	1	Baixa

b) *Arquivos de Interface Externa (AIE)*: conforme definição dos AIE no item 3.4, a aplicação que está sendo contada não apresenta nenhum AIE, pois não apresenta interface com outra aplicação. Sendo assim, $AIE = 0$.

4.2.3.2 Contagem das funções do tipo transação

a) *Entradas Externas (EE)*: de acordo com a descrição das EE no item 3.4, pode-se visualizar na tabela 8 as EE identificadas na aplicação em questão, considerando os tipos de dados, os arquivos referenciados e o grau da função:

TABELA 8 - Identificação das Entradas Externas (EE)

<i>Entrada Externa (EE)</i>	<i>Tipo de Dados</i>	<i>Arquivos Referenciados</i>	<i>Grau da Função</i>
<i>Aluno</i>			
Inclusão de aluno	12 (10 campos, 1 ações e 1 mensagens de erro)	2 (Aluno e Curso de Graduação)	Média
Alteração de aluno	12 (10 campos, 1 ações e 1 mensagens de erro)	2 (Aluno e Curso de Graduação)	Média
Exclusão de aluno	1 (somente é necessário selecionar o aluno a ser excluído)	1	Baixa
<i>Autorização</i>			
Inclusão de autorização	5 (3 campos, 1 ações e 1 mensagens de erro)	1	Baixa
Alteração de autorização	5 (3 campos, 1 ações e 1 mensagens de erro)	1	Baixa
Exclusão de autorização	1 (somente é necessário selecionar a autorização a ser excluída)	1	Baixa
<i>Categoria de curso de graduação</i>			
Inclusão de categoria de curso de graduação	3 (1 campo, 1 ações e 1 mensagens de erro)	1	Baixa
Alteração de categoria de curso de graduação	3 (1 campo, 1 ações e 1 mensagens de erro)	1	Baixa
Exclusão de categoria de curso de graduação	1 (somente é necessário selecionar a categoria de curso de graduação a ser excluída)	1	Baixa
<i>Chefe do SERED</i>			
Inclusão de chefe do SERED	7 (5 campos, 1 ações e 1 mensagens de erro)	1	Baixa
Alteração de chefe do SERED	7 (5 campos, 1 ações e 1 mensagens de erro)	1	Baixa
Exclusão de chefe do SERED	1 (somente é necessário selecionar o chefe do SERED a ser excluído)	1	Baixa
<i>Curso de graduação</i>			
Inclusão de curso de graduação	9 (7 campos, 1 ações e 1 mensagens de erro)	5 (Curso de Graduação, Categoria, Habilitação, Autorização e Resolução)	Alta
Alteração de curso de graduação	9 (7 campos, 1 ações e 1 mensagens de erro)	5 (Curso de Graduação, Categoria, Habilitação, Autorização e Resolução)	Alta
Exclusão de curso de graduação	1 (somente é necessário selecionar o curso de graduação a ser excluído)	1	Baixa
<i>Habilitação de curso de graduação</i>			
Inclusão de habilitação de curso de graduação	3 (1 campo, 1 ações e 1 mensagens de erro)	1	Baixa
Alteração de habilitação de curso de graduação	3 (1 campo, 1 ações e 1 mensagens de erro)	1	Baixa
Exclusão de habilitação de curso de graduação	1 (somente é necessário selecionar a habilitação de curso de graduação a ser excluída)	1	Baixa
<i>Pró-reitor de ensino</i>			
Inclusão de pró-reitor de ensino	7 (5 campos, 1 ações e 1 mensagens de erro)	1	Baixa
Alteração de pró-reitor de ensino	7 (5 campos, 1 ações e 1 mensagens de erro)	1	Baixa

Exclusão de pró-reitor de ensino	1 (somente é necessário selecionar o pró-reitor de ensino a ser excluído)	1	Baixa
<i>Reconhecimento</i>			
Inclusão de reconhecimento	6 (4 campos, 1 ações e 1 mensagens de erro)	1	Baixa
Alteração de reconhecimento	6 (4 campos, 1 ações e 1 mensagens de erro)	1	Baixa
Exclusão de reconhecimento	1 (somente é necessário selecionar o reconhecimento a ser excluído)	1	Baixa
<i>Registro</i>			
Inclusão de registro	6 (4 campos, 1 ações e 1 mensagens de erro)	1	Baixa
Alteração de registro	6 (4 campos, 1 ações e 1 mensagens de erro)	1	Baixa
Exclusão de registro	1 (somente é necessário selecionar o registro a ser excluído)	1	Baixa
<i>Reitor</i>			
Inclusão de reitor	7 (5 campos, 1 ações e 1 mensagens de erro)	1	Baixa
Alteração de reitor	7 (5 campos, 1 ações e 1 mensagens de erro)	1	Baixa
Exclusão de reitor	1 (somente é necessário selecionar o reitor a ser excluído)	1	Baixa
<i>Resolução de curso de graduação</i>			
Inclusão de resolução de curso de graduação	5 (3 campos, 1 ações e 1 mensagens de erro)	1	Baixa
Alteração de resolução de curso de graduação	5 (3 campos, 1 ações e 1 mensagens de erro)	1	Baixa
Exclusão de resolução de curso de graduação	1 (somente é necessário selecionar a resolução de curso de graduação a ser excluída)	1	Baixa
<i>Secretário acadêmico</i>			
Inclusão de secretário acadêmico	7 (5 campos, 1 ações e 1 mensagens de erro)	1	Baixa
Alteração de secretário acadêmico	7 (5 campos, 1 ações e 1 mensagens de erro)	1	Baixa
Exclusão de secretário acadêmico	1 (somente é necessário selecionar o secretário acadêmico a ser excluído)	1	Baixa

b) *Saídas Externas* (SE): conforme descrição apresentada no item 3.4, a aplicação em questão apresenta a SE identificada:

- Relatório com o total de alunos matriculados por curso de graduação.
 - Tipo de dados: 3 (nome do aluno, nome do curso de graduação e quantidade).
 - Arquivos Referenciados: 2 (aluno e curso de graduação).
 - Grau da função: baixa.

c) *Consulta Externa* (CE): de acordo com a descrição no item 3.4, apresenta-se a única CE, pode-se observar a mesma no anexo 3.

- Gerar diploma
 - Entrada:
 - Tipo de dados: 2 (nome do curso de graduação e nome do aluno desejados).
 - Arquivos Referenciados: 2 (curso de graduação e aluno)
 - Saída:
 - Tipo de dados: 32
 - Arquivos Referenciados: 11 (curso de graduação, categoria, habilitação, autorização, resolução, reconhecimento, aluno, registro, reitor, pró-reitor de ensino e secretário acadêmico).
 - Grau da função: alta.

Ressalta-se que as consultas simples, como por exemplo, consulta de cadastro de curso de graduação, consulta de cadastro de alunos, consulta de cadastro de reitor, entre outras, não estão sendo consideradas nesta aplicação, porque a quantidade de tipo de dados das mesmas não justificam a contagem, uma vez que apresentariam na sua maioria valor pouco significativo (VAZQUEZ, SIMÕES e ALBERT, 2006).

4.2.3.3 Cálculo final dos PF não-ajustados

Depois de contabilizadas e identificadas as complexidades das funções do tipo *dados* e das funções do tipo *transação*, o próximo passo consiste em obter o valor dos PF não-ajustados. Assim, a tabela 9 é utilizada para o cálculo dos PF não-ajustados da aplicação em questão.

TABELA 9 - Cálculo dos PF não-ajustados ou brutos

<i>Tipo de função</i>	<i>Complexidade funcional</i>	<i>Total por complexidade</i>	<i>Total tipo função</i>
<i>Funções do tipo dados</i>			
ALI	Número funções baixa * 7	12 * 7 = 84	84
	Número funções média * 10		
	Número funções alta * 15		
AIE	Número funções baixa * 5		0
	Número funções média * 7		
	Número funções alta * 10		
<i>Funções do tipo transação</i>			
EE	Número funções baixa * 3	32 * 3 = 96	116
	Número funções média * 4	2 * 4 = 8	
	Número funções alta * 6	2 * 6 = 12	
SE	Número funções baixa * 4	1 * 4 = 4	4
	Número funções média * 5		
	Número funções alta * 7		
CE	Número funções baixa * 3	1 * 6 = 6	6
	Número funções média * 4		
	Número funções alta * 6		
Total de PF não-ajustados			210

(FONTE: adaptado de DIAS, 2003)

4.2.4 Definição do valor do fator de ajuste

O valor do fator de ajuste é baseado na influência das 14 CGS, apresentadas no item 3.4. O quadro 14 apresenta as respostas e o grau de influência de cada CGS para a aplicação em questão.

QUADRO 14 - Fator de ajuste da aplicação em questão

1. Comunicação de dados: a aplicação é puramente <i>batch</i> ou uma estação de trabalho isolada. Grau de influência: 0
2. Processamento distribuído: a aplicação não participa da transferência de dados ou processamento entre os componentes do sistema. Grau de influência: 0
3. Performance: o usuário não estabeleceu nenhum requisito especial sobre performance. Grau de influência: 0

4. Configuração altamente utilizada: não existem restrições operacionais implícitas ou explícitas nos requisitos. Grau de influência: 0
5. Volume de transações: são antecipados períodos de pico de processamento (por exemplo: mensal, quinzenal, periódico, anual). Grau de influência: 1
6. Entrada de dados <i>on-line</i> : todas as transações são processadas em lote. Grau de influência: 0
7. Eficiência do usuário final: menus, movimentação automática do cursor. Grau de influência: 1
8. Atualização <i>on-line</i> : não há nenhuma atualização <i>on-line</i> Grau de influência: 0
9. Complexidade de processamento: a aplicação não envolve processamento complexo. Grau de influência: 0
10. Reutilização: não houve preocupação com reutilização de código. Grau de influência: 0
11. Facilidade de instalação: o usuário não definiu considerações especiais, mas é necessário <i>setup</i> para instalação. Grau de influência: 1
12. Facilidade de operação: a aplicação minimiza a necessidade da manipulação de papel. Grau de influência: 1
13. Múltiplos locais: necessidade de múltiplos locais foi considerada no projeto, devendo operar nos mesmos ambientes idênticos de <i>software</i> e <i>hardware</i> . Grau de influência: 1
14. Facilidade de mudanças: não houve nenhuma preocupação específica. Grau de influência: 0

(FONTE: adaptado de VAVASSORI, 2002)

O quadro 14 apresentou as CGS identificadas na aplicação em questão e o seu grau de influência. As características que possuem o grau de influência com o valor igual a zero não se manifestam na aplicação, já as que possuem grau de influência com o valor igual a um estão presentes, porém possuem um nível de influência sobre o sistema considerado mínimo.

Após a identificação do grau de influência de cada CGS, aplica-se a equação 1 (apresentada no item 3.5.5):

$$VAF = (TDI * 0,01) + 0,65 \quad \text{Eq.1}$$

$$VAF = (5 * 0,01) + 0,65$$

tem-se o valor **0,70** como fator de ajuste da aplicação em questão.

4.2.5 Contagem dos PF ajustados

O passo final consiste na contagem dos PF ajustados obtida com a aplicação da equação 3 (apresentada no item 3.5.6.2):

$$\mathbf{PFA = PF_N\tilde{A}O_AJUSTADO * VAF} \quad \mathbf{Eq.3}$$

$$PFA = 210 * 0,70$$

tem-se o valor **147** como pontos de função da aplicação em questão.

4.3 Aplicação dos PF na estimativa de prazos e custos

O objetivo deste item é demonstrar o uso do valor dos PF Ajustados, ou seja, como usar este valor para o cálculo da estimativa de prazos e custos, gerando indicadores de desempenho que fornecem subsídios para análises.

Quando se trata do desenvolvimento, manutenção ou aquisição de um *software* dois questionamentos estão sempre presentes: quanto tempo será necessário para a conclusão do projeto? E quanto o projeto vai custar para a organização?

O detalhamento dos requisitos de um projeto de *software*, da equipe responsável e da tecnologia empregada, são fatores que proporcionam dificuldade na obtenção de respostas confiáveis a estas perguntas. Devido a isto, determinar o custo final e a data de conclusão de um projeto é uma tarefa que só pode ser realizada quando o projeto já estiver finalizado. Antes disso, tudo o que pode ser calculado são estimativas.

Para obtenção destas estimativas faz-se necessário a aplicação de métricas específicas, destacando-se a Análise de Pontos de Função, que proporciona a obtenção do tamanho do *software* através de sua aplicação.

A Estimativa é a obtenção de um valor, através da aplicação de cálculos que têm como base a fase de análise e projeto do sistema. Tem como objetivo a utilização deste valor como indicativo de custo e data final de um determinado projeto de *software*.

O processo de estimativas de um projeto de *software* envolve quatro atividades:

1. Estimar o tamanho do produto a ser gerado;
2. Estimar o esforço empregado na execução do projeto;
3. Estimar a duração do projeto;
4. Estimar o custo do projeto.

Primeiramente, define-se a unidade de medida utilizada. De acordo com as categorias de medição, têm-se duas unidades mais comuns, linhas de código (LOC) e pontos de função (PF). Optou-se por PF, pois a obtenção destes depende unicamente da funcionalidade requerida para o *software* e não da tecnologia empregada em seu desenvolvimento.

É possível realizar uma estimativa de tamanho desde a fase inicial do levantamento de requisitos e refiná-la à medida que se avança no entendimento do projeto.

Inicia-se o processo de estimativa a partir dos primeiros níveis de abstração dos requisitos do projeto. Os dados históricos de projetos passados, os recursos disponíveis dentro ou fora da organização, os dados de custo e os fatores de risco que cercam o projeto devem ser considerados para que o processo seja completo e eficiente. As estimativas devem passar por um processo de aprovação antes de serem registradas na base histórica. Durante todo o ciclo de vida do desenvolvimento do *software* deve-se reestimar. Com o produto concluído, na etapa final do processo, deve-se registrar na base histórica as medidas reais de tamanho, esforço, duração e custo.

Após a obtenção do tamanho do produto que neste trabalho foi obtido através da aplicação da APF, pode-se estimar o esforço empregado na execução do projeto, a duração e custo do projeto.

4.3.1 Estimativa de esforço empregado na execução do projeto

O esforço, para desenvolvimento do produto, é gerado em homens_horas (HH), baseando-se em dois fatores (HAZAN e STAA, 2004):

- Tamanho do projeto, estimado em pontos de função (PF);
- Fator de conversão HH/PF: quantidade de horas empregadas para

implantar um PF, recuperado de bases históricas.

A produtividade depende de diversos fatores, tais como tamanho, plataforma de desenvolvimento, complexidade da aplicação e experiência da equipe. Pode ser obtida de bases históricas.

No caso dos projetos em que não há dados históricos de projetos similares disponíveis, pode-se utilizar dados de produtividade pesquisados por meio de *benchmarking*. A tabela 10 apresenta alguns dados pesquisados de nível de linguagem da empresa *Software Productivity Research* e *benchmarking* com outras empresas (HAZAN e STAA, 2004).

TABELA 10 - Produtividade em horas/PF por linguagem

<i>Ambiente/Linguagem</i>	<i>Produtividade (horas/PF)</i>		
	<i>Baixa</i>	<i>Média</i>	<i>Alta</i>
<i>Mainframe</i>			
Cobol	24,6 h	17,6 h	13,2 h
Natural	13,2 h	8,8 h	6,6 h
<i>Micro e Cliente/Servidor</i>			
Visual Basic	8,8 h	6,8 h	5,7 h
Delphi	8,8 h	6,8 h	5,7 h
Oracle	13,2 h	8,8 h	6,6 h
<i>Web/Documentos</i>			
ASP	12 h	10 h	8 h
Java	17 h	15 h	12 h
<i>Lotus Notes</i>	5,5 h	3,9 h	3,1 h

(FONTE: HAZAN e STAA, 2004)

Ressalta-se que o nível de produtividade baixa está associado às aplicações complexas ou equipes inexperientes na plataforma de desenvolvimento. Já o nível de produtividade alta está associado às aplicações simples, ou a equipes experientes na plataforma. As estimativas de prazos são diretamente influenciadas pelo nível de produtividade da equipe desenvolvedora. A produtividade alta implica em prazos de desenvolvimento menores (HAZAN e STAA, 2004).

Caso, a linguagem de desenvolvimento utilizada não seja encontrada na tabela de linguagens, seleciona-se uma linguagem com produtividade similar. Caso o projeto seja desenvolvido em mais de uma linguagem de programação, define-se um fator de conversão HH, contemplando o percentual de cada linguagem (HAZAN e STAA, 2004).

Ressalta-se que a tabela 10 está sendo utilizada, pois o estudo de caso utilizado neste trabalho não possui uma base de dados históricos. A produtividade por linguagem pode ser bastante variável. A acurácia das estimativas de esforço é fortemente dependente da existência de uma base de dados histórico, contendo dados de esforço, tamanho e atributos de produtividade de projetos concluídos (HAZAN e STAA, 2004).

Aplicando a estimativa de esforço no estudo de caso, constatou-se que a produtividade é de nível alto, pois a linguagem utilizada no desenvolvimento é *Delphi* e o ambiente *micro*. Então, tem-se o valor **837,9** ($5,7 \cdot 147$) horas como estimativa de esforço para o desenvolvimento da aplicação em questão, conforme equação 5. Onde, **5,7** corresponde a produtividade em horas empregadas no desenvolvimento da aplicação na linguagem *Delphi*, de acordo com a tabela 10 e **147** corresponde ao tamanho do *software* estimado em PF.

$$\text{Esforço} = \text{Horas desenvolvimento Delphi} * \text{PFA} \quad \text{Eq.5}$$

$$\text{Esforço} = 5,7 * 147$$

4.3.2 Estimativa de duração do projeto

O próximo passo após a estimativa de esforço necessário para o desenvolvimento do projeto é a estimativa de prazo. Esta estimativa leva em consideração o tamanho e o esforço estimados do projeto e o tamanho da equipe desenvolvedora do projeto.

Devem-se considerar os aspectos (HAZAN e STAA, 2004):

- Um mês possui 22 dias úteis;
- Com uma jornada de trabalho de 8 horas/dia, a produtividade média do profissional no Brasil é de 6 horas/dia.

Assim, utiliza-se a equação 6 (HAZAN e STAA, 2004):

$$\text{Prazo (em dias)} = \text{Esforço (em horas)} / (\text{Tamanho da equipe} * 6) \quad \text{Eq. 6}$$

$$\text{Prazo (em dias)} = 837,9 / (1 * 6) \quad \text{(a)}$$

$$\text{Prazo (em dias)} = 837,9 / (3 * 6) \quad (\text{b})$$

No estudo de caso utilizado neste trabalho, há duas considerações para o tamanho da equipe. A primeira é considerar apenas a aluna que desenvolveu o projeto, desta forma se obtém o valor **139,65** dias (a).

A segunda considera a aluna, o orientador e o co-orientador, neste caso se obtém o valor **46,55** dias (b) como estimativa de duração do desenvolvimento da aplicação em questão.

4.3.3 *Estimativa de custo do projeto*

A estimativa de custo, derivada da estimativa de tamanho, leva em consideração alguns atributos: estimativa de mão-de-obra, recursos computacionais, treinamento, consultoria, viagens e custos indiretos. Ressalta-se que o esforço (HH/PF) e os custos indiretos influenciam fortemente no custo/PF.

Destaca-se que para sistemas terceirizados as estimativas de mão-de-obra são aferidas com base no número de PF estimados e o preço por PF especificado no contrato firmado entre a empresa contratada e a empresa contratante. Já o custo dos sistemas desenvolvidos por profissionais da empresa, é estimado considerando o esforço e o custo dos profissionais alocados para o desenvolvimento do projeto. É importante lembrar que o custo por PF pode ser bastante variável. O esforço (HH/PF) e outros custos indiretos influenciam fortemente no custo/PF (HAZAN e STAA, 2004).

A maioria das empresas, não tem o costume de aplicar métricas para obtenção das estimativas de prazos e custos dos seus projetos, conseqüentemente, baseiam-se em experiências anteriores. Isto não garante a eficácia das estimativas, que uma vez obtidas a partir de métricas, como APF tem maior precisão (VAZQUEZ, SIMÕES e ALBERT, 2006).

Para obtenção da estimativa de custo, deve-se definir o valor do PF, o que varia de uma empresa para outra, pois o preço irá variar de acordo com o trabalho requerido para a construção de um PF e dos subprodutos a serem também entregues.

Por exemplo, ao se contratar uma empresa apenas para o trabalho de codificação e testes de unidade de um sistema espera-se que o preço do PF seja inferior ao caso da contratação da mesma empresa para a realização de todo o ciclo de desenvolvimento do sistema (BFPUG, 2007).

Nesta aplicação, o valor utilizado é o preço médio cobrado por PF, pois não se dispõem de bases históricas para estabelecer o valor real. Conforme, SERPRO (2007), o valor considerado pode ser R\$ 500,00, mas conforme a tecnologia e a complexidade do sistema, esse valor pode variar de R\$ 300,00 a R\$ 1 mil.

Ainda, de acordo com BFPUG (2007), um exemplo para o valor do PF pode ser U\$ 10 (dez dólares).

Por isto, neste projeto é apresentado o valor recomendado por SERPRO (2007) e por BFPUG (2007).

Então, calcula-se o custo por PF baseado no tamanho do projeto, e utiliza-se a equação 7.

$$\begin{aligned} \text{Custo projeto} &= \text{Tamanho (PF)} * \text{Custo(R\$/PF)} && \text{Eq. 7} \\ \text{Custo} &= 147 * 10 && \text{(a)} \\ \text{Custo} &= 147 * 500 && \text{(b)} \end{aligned}$$

Considerando o valor de U\$10/PF tem-se o valor de **U\$ 1.470,00** (um mil quatrocentos e setenta dólares) (a) como custo total do projeto. Convertendo este valor para a moeda brasileira, o Real, obtém-se **R\$ 2.558,68** (dois mil quinhentos e cinquenta e oito reais e sessenta e oito centavos). A conversão é de acordo com a cotação do Banco Central do Brasil da data de 16/11/2007 com taxa de “1,7406 Real/Brasil = 1 Dólar” dos Estados Unidos (BANCO CENTRAL DO BRASIL, 2007).

Utilizando o valor de R\$ 500,00/PF obtém-se o valor de **R\$ 73.500,00** (setenta e três mil e quinhentos reais) (b) como custo total do projeto em questão.

A diferença entre esses valores pode até ser justificada pela ausência de uma base histórica para estimativas do projeto em estudo, pois na presença da mesma o valor poderia ser mais preciso. Embora estes valores sejam recomendações que deveriam ser precisas, apresentam uma grande diferença entre si, o que não significa

que estejam errados, pois o valor varia de acordo com o tamanho do *software* sendo medido.

Contudo, entende-se que, embora esses valores estejam distantes, pode-se ter uma idéia de como utilizar o valor em PF coletados do processo de contagem da métrica APF.

4.4 Conclusão

Após a apresentação e entendimento dos conceitos envolvidos na APF, foi possível demonstrar sua aplicação em um estudo de caso desenvolvido em um ambiente real, objetivando a obtenção do tamanho do *software*.

Com base na demonstração da aplicação da APF, apresentada neste capítulo, é possível realizar a contagem dos PF, bem como estimativas de esforço, prazo e custo em projetos futuros. Este capítulo serve como um tutorial e teve o objetivo de apresentar e incentivar o uso desta métrica tão sedimentada atualmente, pois mede a funcionalidade do *software*, através do ponto de vista do usuário.

A aplicação da métrica resulta na obtenção de um valor em PF, que pode ser utilizado na obtenção das estimativas, como demonstrou o exemplo apresentado. A partir do valor encontrado nas estimativas, é possível fornecer à gerência um indicador da medição, para que estes possam interpretar e avaliar os resultados obtidos.

É de extrema importância a demonstração da utilização do valor obtido através da aplicação da APF, pois o valor por si só não esclarece sua utilização.

O valor encontrado sempre pode ser combinado com outro cálculo para encontrar a medida necessária.

5 CONSIDERAÇÕES FINAIS

Um problema comum identificado atualmente que merece destaque é o não cumprimento de prazos no desenvolvimento de *software*, o que gera um aumento considerável nos custos previstos para o projeto. Isso ocorre também pela falta de prática e conhecimento sobre o uso de métricas que gera a imprecisão na definição de prazos e custos dos projetos.

Uma solução plausível para este problema é a aplicação de métricas de *software* no processo de desenvolvimento. Através das métricas é possível não apenas estimar prazos e custos, mas também acompanhar, analisar e controlar o progresso dos projetos, permitindo identificar aspectos que necessitem de melhoria.

Na tentativa de minimizar o problema da imprecisão e desconhecimento, destaca-se a Análise de Pontos de Função, que é utilizada na coleta de métricas para as estimativas de prazos e custos e possui uma aceitação boa na gerência de projetos.

Baseando-se no exposto, este trabalho teve como objetivos fornecer um referencial teórico sobre a importância da utilização de Métricas de *Software* como também da Análise de Pontos de Função e proporcionar uma orientação do uso da APF, bem como a demonstração de sua utilização na obtenção das estimativas de prazos e custos no estudo de caso. Para a efetivação de tais objetivos, foi demonstrada e aplicada à métrica Análise de Pontos de Função em um estudo de caso, como forma de orientar a utilização dos PF encontrados.

Destaca-se a Análise de Pontos de Função, que proporciona a obtenção da medida do tamanho do *software* em PF, desconsiderando a tecnologia empregada no desenvolvimento do mesmo. Desta forma, entende-se que os objetivos foram alcançados em sua totalidade. Através deste trabalho, foi possível verificar a real

importância da aplicação de uma métrica no processo de desenvolvimento de *software* para um maior controle do mesmo.

Ressalta-se que a demonstração do processo da APF no estudo de caso, bem como seu uso em estimativas de prazos e custos contribui orientando como trabalhar e contar os PF.

Assim, este trabalho contribui para a aplicação da Análise de Pontos de Função no processo de desenvolvimento de *software*, pois descreve um tutorial de como aplicar a métrica e posteriormente como utilizar o valor encontrado através da aplicação da mesma.

Uma das dificuldades vivenciadas foi em encontrar material bibliográfico completo, pois os existentes não são claros e também não proporcionam o entendimento adequado para a aplicação da métrica, que apresenta um alto grau de complexidade.

Como indicação para trabalhos futuros sugere-se a análise do valor encontrado neste trabalho através de uma avaliação e comparação com o valor real. Também, pode ser feita uma comparação entre Análise de Pontos de Função e Análise de Casos de Uso, verificando a melhor alternativa. Outra sugestão interessante é o desenvolvimento de uma ferramenta simples para trabalhar com a contagem dos PF e estimativas.

REFERÊNCIAS BIBLIOGRÁFICAS

BANCO CENTRAL DO BRASIL – Site do Banco Central do Brasil. Disponível em: <[http:// www.bcb.gov.br/](http://www.bcb.gov.br/)>. Acessado em: 17 nov. 2007.

BARTIÉ, A. **Garantia da Qualidade de Software**. Rio de Janeiro: Campus, 2002. 292 p.

BFPUG. **Brazilian Function Point Users Group**. Divulga a utilização das métricas no desenvolvimento de sistemas, em particular a Análise de Pontos de Função. Disponível em: <<http://www.bfpug.com.br/>>. Acessado em: 02 mar. 2007.

CALAZANS, A. T. S. Medição de Tamanho para Sistemas de Data Mart. In: SIMPÓSIO BRASILEIRO DE QUALIDADE DE SOFTWARE – SBQS, 3., 2004, Brasília. **Anais...** Brasília: UCB, 2004. Disponível em: <<http://www.mct.gov.br/index.php/content/view/14975.html>>. Acessado em: 21 abr. 2007.

CAMPÊLO, G. M. C. **A Utilização de Métricas na Gerência de Projetos de Software uma Abordagem Focada no CMM Nível 2**. 2002. 135 f. Dissertação (Mestrado em Ciência da Computação) – Centro de Informática, Universidade Federal de Pernambuco, Recife.

CHAVES, F. F. **Roteiro para Aplicação de Métricas e Qualidade em Subcontratação de Software**. 2003. 46 f. Monografia (Especialização em Informática Pública) - Pontifícia Universidade Católica de Minas Gerais, Belo Horizonte.

DIAS, R. Análise por Pontos de Função: Uma Técnica para Dimensionamento de Sistemas de Informação. **RESI - Revista Eletrônica de Sistemas de Informação**, Campo Largo, 3. ed., 2003. Seção Artigos. Disponível em: <<http://www.inf.ufsc.br/resi>>. Acessado em: 24 abr. 2007.

GRANDCHAMP, R. E. **Gerenciamento de Projetos de Software**. 2002. 82 f. Monografia (Especialização em MBA – Gerência de Produção e Tecnologia) – Departamento de Economia, Contabilidade, Administração e Secretariado, Universidade de Taubaté, Taubaté.

HAZAN, C. Análise de Pontos de Função: Uma Abordagem Gerencial. In: JORNADA DE ATUALIZAÇÃO EM INFORMÁTICA – JAI, 19., 2000, Curitiba. **Anais do XX Congresso da Sociedade Brasileira de Computação**. Curitiba: SBC, 2000. Disponível em: <[http:// www.niee.ufrgs.br/SBC2000/eventos/ jai.htm](http://www.niee.ufrgs.br/SBC2000/eventos/jai.htm) >. Acessado em:

23 abr. 2007.

HAZAN, C; STAA, A. V. Análise e Melhoria de um Processo de Estimativas de Tamanho de Projetos de Software. In: SIMPÓSIO INTERNACIONAL DE MELHORIA DE PROCESSO DE SOFTWARE – SIMPROS, 6., 2004, São Paulo. **Anais...** São Paulo: SENAC, 2004. Disponível em: <<http://www.lbd.dcc.ufmg.br/bdbcomp/servlet/Evento?id=101>>. Acessado em: 21 mai. 2007.

IPFUG. **International Function Point Users Group**. Apresenta o Manual de Práticas de Contagem da Análise de Pontos de Função. Disponível em: <<http://www.ifpug.org/publications>>. Acessado em: 02 mar. 2007.

LEMKE, A. P. **Solução Computacional para um Sistema de Avaliação da Qualidade dos Serviços em Instituições de Ensino Superior, baseado nos Parâmetros do PQSP**. 2005. 94 f. Monografia (Bacharelado em Ciência da Computação) – Instituto de Física e Matemática, Universidade Federal de Pelotas, Pelotas.

PAULA FILHO, W. P. **Engenharia de Software: Fundamentos, Métodos e Padrões**. 2. ed. Rio de Janeiro: Livros Técnicos e Científicos, 2003. 602 p.

PETERS, J. F.; PEDRYCZ, W. **Engenharia de Software: Teoria e Prática**. Rio de Janeiro: Campus, 2001. 602 p.

PFLEEGER, S. L. **Engenharia de Software: Teoria e Prática**. 2. ed. São Paulo: Prentice Hall, 2004. 535 p.

PRESSMAN, R. S. **Software Engineering: A Practitioner's Approach**. 5th ed. Nova York: McGraw-Hill, 2001. 860 p.

ROCHA, A. R. C. da; MALDONADO, J. C.; WEBER, K. C. **Qualidade de Software: Teoria e Prática**. São Paulo: Prentice Hall, 2001. 303 p.

SCOPPEL, L. **Aplicações do Processo Unificado na Informatização do Setor SERED**. 2004. 54 f. Trabalho de Conclusão de Curso (Bacharelado em Informática) – Departamento de Ciências Exatas e Tecnológicas, Universidade do Planalto Catarinense – UNIPLAC, Lages.

SERPRO – **Serviço Federal de Processamento de Dados**. Empresa pública de prestação de serviços em tecnologia da informação do Brasil. Disponível em: <<http://www.serpro.gov.br/>>. Acessado em: 14 nov. 2007.

SOMMERVILLE, I. **Engenharia de Software**. 6. ed. São Paulo: Addison Wesley, 2003. 592 p.

TAVARES, H. C. A. B.; CARVALHO, A. E. S.; CASTRO, J. F. B. Medição de Pontos por Função a Partir da Análise de Requisitos. In: WORKSHOP DE ENGENHARIA DE REQUISITOS – WER, 5., 2002, Valencia. **Anais...** Valencia: Universidad Politecnica de Valencia, 2002. Disponível em <<http://wer.inf.puc-rio.br/wer02/Anais02.html>>. Acessado em: 20 mai. 2007.

VAVASSORI, F. B. **Metodologia para o Gerenciamento Distribuído de Projetos e Métrica de Software**. 2002. 211 f. Tese (Doutorado em Engenharia de Produção) – Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina - UFSC, Florianópolis.

VAZQUEZ, C. E.; SIMÕES, G. S.; ALBERT, R. M. **Análise de Pontos de Função: Medição, Estimativas e Gerenciamento de Projetos de Software**. 5. ed. São Paulo: Érica, 2006. 230 p.

WAZLAWICK, R. S. **Análise e Projeto de Sistemas de Informação Orientados a Objetos**. Rio de Janeiro: Elsevier, 2004. 298 p.

APÊNDICE - TUTORIAL PARA APLICAÇÃO DA ANÁLISE DE PONTOS DE FUNÇÃO NO PROCESSO DE DESENVOLVIMENTO DE SOFTWARE

Mônica Pagno da Silva¹, Viviane Duarte Bonfim¹

¹Departamento de Ciências Exatas e Tecnológicas da Universidade do Planalto Catarinense (UNIPLAC) – Lages – SC – Brasil

{monicap,vivi}@uniplac.net

***Abstract.** Some of the reasons why using Software Metrics can be highlighted, such as: providing subsidies to determinate efforts, resources, timing and cost of development; managing software contracts; making an estimate base; evaluating the process productivity. With that in mind, the major concern for the companies is the constant imprecision on timing and costs definitions for projects. Trying to minimize that imprecision, there is the Analysis of Points of Function (APF), which is a metric used to collect indicators (amounts) that refer to the timing and costs estimates. This way, this article has the main purpose to demonstrate the counting process of APF through a case of study.*

***Resumo.** Podem-se destacar motivos para a utilização de Métricas de Software, tais como: prover subsídios para determinar esforço, recursos, duração e custo do desenvolvimento; gerenciar contratos de software; formar uma base para estimativa; avaliar a produtividade do processo. Diante disto, destaca-se que o mais preocupante para as empresas é a constante imprecisão na definição de prazos e custos dos projetos. Na tentativa de minimizar tal imprecisão, destaca-se a Análise de Pontos de Função (APF), que é uma métrica utilizada para coletar indicadores (valores) referentes às estimativas de custos e prazos. Assim, aborda-se como objetivo principal deste artigo a demonstração do processo de contagem da APF através de um estudo de caso.*

1. Introdução

Atualmente o maior problema encontrado nas empresas desenvolvedoras de software está no estabelecimento e cumprimento de prazos e custos precisos para seus projetos.

A resolução deste problema está na aplicação de Métricas de Software, que consiste na mensuração dos indicadores quantitativos do tamanho e complexidade de um sistema. Esses indicadores são utilizados para confrontar os desempenhos observados no passado, com o propósito de derivar precisão de desempenho futuro (estimativa).

A idéia de estimar prazos e custos envolvidos em um projeto é algo bastante decisivo no desenvolvimento e, por isso, empresas que prezam pela qualidade estão cada vez mais aderindo a essa prática. No entanto, existe uma dificuldade em estimar, devido à complexidade das métricas, o que acarreta no desinteresse muitas vezes em trabalhar com métricas ocasionando desconhecimento sobre o mesmo.

Nesse contexto, destaca-se a Análise de Pontos de Função (APF), uma métrica que permite medir o tamanho de um software sob a visão do usuário e diante dos requisitos descritos pelo mesmo. A unidade de medida utilizada é Pontos de Função (PF). É uma das métricas de estimativa de tamanho mais sedimentadas no mercado e que proporciona resultados cada vez mais precisos à medida que artefatos das fases de análise e projeto são gerados (Vazquez, Simões e Albert, 2006).

Diante disso, é demonstrada a métrica e a aplicação da APF em um estudo de caso, especificamente na determinação de prazos e custos, como forma de orientação e auxílio para o uso da métrica. Neste enfoque, a medida obtida da aplicação da APF é utilizada para orientar a obtenção da estimativa de prazos e custos. Isto pode auxiliar gerentes de projetos em um controle mais preciso sobre o cronograma e custos estabelecidos no início do projeto de software e, também, no esclarecimento da importância do uso de Métricas de Software, especificamente APF.

2. Estudo de Caso

O estudo de caso utilizado neste tutorial consiste em um sistema capaz de registrar informações que precisam constar nos diplomas emitidos aos formandos dos cursos da UNIPLAC e disponibilizar um ambiente prático para gerá-los. A partir disto, deve ser possível realizar consultas rápidas ao sistema, facilitando, assim, o processo de registro e geração de diplomas. O setor em questão é o Setor de Expedição de Certificados e Diplomas (SERED). O estudo de caso foi desenvolvido por Scoppel, 2004.

3. Demonstração da Aplicação da APF

A finalidade do processo de contagem é a obtenção de um valor numérico, cuja unidade são os pontos de função (PF). São apresentados neste item os passos para a obtenção do valor em PF que, posteriormente pode ser utilizado no cálculo de estimativas. Para um melhor entendimento, esses passos são aplicados no estudo de caso descrito no item 2.

Para iniciar o processo de contagem dos pontos de função no estudo de caso devem ser seguidos os seguintes passos:

Passo 1: Primeiramente deve-se determinar o tipo de contagem. Têm-se três tipos: projeto de desenvolvimento, projeto de melhoria e projeto de aplicação. Neste caso aplica-se o projeto de aplicação, pois toda a fase de projeto do sistema já está concluída.

Após determinar o tipo de contagem, inicia-se o segundo passo.

Passo 2: Consiste na identificação do escopo da contagem e da fronteira da aplicação. O *escopo da contagem* define se a contagem abrange um ou mais sistemas ou apenas parte de um sistema. A *fronteira da aplicação* é a interface conceitual que delimita o *software* a ser medido e o mundo exterior (seus usuários). Desta forma, o sistema não apresenta entradas fora de sua fronteira, pois todas as informações são centralizadas em um único sistema, em um determinado ambiente de trabalho. Assim, o escopo da contagem abrange apenas uma aplicação e todas as suas funcionalidades.

Passo 3: Neste passo são contados os pontos de função não-ajustados que, combinados com o valor do fator de ajuste, resultam no valor dos pontos de função ajustados. A contagem dos pontos de função não-ajustados é realizada através de dois procedimentos:

Contagem das funções do tipo dados: as funções do tipo dados representam as funcionalidades fornecidas pelo sistema ao usuário, para atender às suas necessidades. Dividem-se em (Vazquez, Simões e Albert, 2006):

Arquivo Lógico Interno – ALI: grupo logicamente relacionado de dados ou informações de controle, mantido dentro da fronteira da aplicação que está sendo contada e é identificado pelo usuário. Exemplo: tabelas de banco de dados atualizadas pela aplicação.

Arquivo de Interface Externa – AIE: grupo logicamente relacionado de dados ou informações de controle, mantidos fora da fronteira da aplicação sendo contada, também identificado pelo usuário. Exemplo: tabelas de banco de dados lidas pela aplicação, mas atualizadas por outra aplicação.

Depois de contadas, deve-se classificar a sua complexidade, esta por sua vez é feita com base no número de dados e número de tipos de registros e é fornecida pela Tabela 1.

Tabela 1 - Tabela de complexidade funcional dos ALI e AIE

Tipos de Dados				
		< 20	20 – 50	> 50
Tipo de Registros	1	Baixa	Baixa	Média
	2 – 5	Baixa	Média	Alta
	>5	Média	Alta	Alta

(Fonte: VAZQUEZ, SIMÕES e ALBERT, 2006, p. 77)

Após determinada a complexidade dos arquivos, deve-se calcular sua contribuição de acordo com a Tabela 2.

Tabela 2 - Tabela de contribuição dos PF não-ajustados das funções do tipo dado

Tipo de Função	Baixa	Média	Alta
Arquivo lógico interno	7 PF	10 PF	15 PF
Arquivo de interface interna	5 PF	7 PF	10 PF

(Fonte: VAZQUEZ, SIMÕES e ALBERT, 2006, p. 80)

Contagem das funções do tipo transação: as funções do tipo transação representam as funcionalidades de processamento de dados fornecidas pelo sistema ao usuário, dividem-se em (Vazquez, Simões e Albert, 2006):

Entrada Externa – EE: tem como função processar dados ou informações de controle originados fora da fronteira da aplicação. Sua finalidade é manter um ou mais ALI ou alterar o comportamento do sistema. Exemplos: incluir cliente, alterar cliente, excluir cliente.

Saída Externa – SE: tem como função emitir dados ou informações de controle para fora da fronteira da aplicação. Sua finalidade é apresentar informações calculadas ao usuário ou outras aplicações. Exemplo: relatório de totais de faturamento de cliente.

Consulta Externa – CE: tem como função enviar dados ou informações de controle para fora da fronteira da aplicação. Sua finalidade é apresentar informações não calculadas ao usuário ou outras aplicações. Exemplo: consulta de cadastro de clientes.

Após determinada a quantidade de arquivos referenciados e tipos de dados, utilizam-se as Tabelas 3 e 4 para classificação da complexidade das EE, SE e CE.

Tabela 3 - Tabela de complexidade funcional para EE

Tipos de Dados				
		< 5	5 – 15	> 15
Arquivos Referenciados	< 2	Baixa	Baixa	Média
	2	Baixa	Média	Alta
	>2	Média	Alta	Alta

(Fonte: VAZQUEZ, SIMÕES e ALBERT, 2006, p. 109)

Tabela 4 - Tabela de complexidade funcional para SE e CE

Tipos de Dados				
		< 6	6 – 19	> 19
Arquivos Referenciados	< 2	Baixa	Baixa	Média
	2 - 3	Baixa	Média	Alta
	>3	Média	Alta	Alta

(Fonte: VAZQUEZ, SIMÕES e ALBERT, 2006, p. 109)

Depois de determinada a complexidade dos arquivos, deve-se determinar sua contribuição utilizando a Tabela 5.

Tabela 5 - Tabela de contribuição dos PF não-ajustados das funções do tipo transação

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

(Fonte: VAZQUEZ, SIMÕES e ALBERT, 2006, p. 112)

Passo 4: Este passo consiste na obtenção do valor dos PF não-ajustados. Para isso reúnem-se na Tabela 6 os valores obtidos no passo 3. Nesta tabela, o número de determinada complexidade é multiplicado pelo valor de sua contribuição.

Tabela 6 - Tabela para o cálculo dos PF não-ajustados ou brutos

Tipo de função	Complexidade funcional	Total por complexidade	Total tipo função
Funções do tipo dados			

ALI	Número funções baixa * 7 Número funções média * 10 Número funções alta * 15	12 * 7 = 84	84
AIE	Número funções baixa * 5 Número funções média * 7 Número funções alta * 10		0
<i>Funções do tipo transação</i>			
EE	Número funções baixa * 3 Número funções média * 4 Número funções alta * 6	32 * 3 = 96 2 * 4 = 8 2 * 6 = 12	116
SE	Número funções baixa * 4 Número funções média * 5 Número funções alta * 7	1 * 4 = 4	4
CE	Número funções baixa * 3 Número funções média * 4 Número funções alta * 6	1 * 6 = 6	6
Total de PF não-ajustados			210

(Fonte: adaptado de DIAS, 2003)

Após o preenchimento da tabela, somam-se os valores para obter-se o total, correspondente aos PF não-ajustados.

Passo 5: Continuando, calcula-se o fator de ajuste, este valor é baseado na influência de 14 CGS. Assim, verifica-se o nível de influência de cada uma das 14 CGS, variando de 0 a 5, de acordo com o Quadro 1.

Quadro 1 - Escala de influência

0 - Nenhuma Influência
1 - Influência Mínima
2 - Influência Moderada
3 - Influência Média
4 - Influência Significativa
5 - Grande Influência

(Fonte: HAZAN, 2000)

O Quadro 2 apresenta as CGS identificadas na aplicação em questão e o seu grau de influência. As características que possuem o grau de influência com o valor igual a zero não se manifestam na aplicação, já as que possuem grau de influência com o valor igual a um estão presentes, porém possuem um nível de influência sobre o sistema considerado mínimo.

Quadro 2 – Fator de ajuste da aplicação em questão

<p>1. Comunicação de dados: a aplicação é puramente <i>batch</i> ou uma estação de trabalho isolada.</p> <p>Grau de influência: 0</p>
<p>2. Processamento distribuído: a aplicação não participa da transferência de dados ou processamento entre os componentes do sistema.</p> <p>Grau de influência: 0</p>
<p>3. Performance: o usuário não estabeleceu nenhum requisito especial sobre performance.</p> <p>Grau de influência: 0</p>
<p>4. Configuração altamente utilizada: não existem restrições operacionais implícitas ou explícitas nos requisitos.</p> <p>Grau de influência: 0</p>
<p>5. Volume de transações: são antecipados períodos de pico de processamento (por exemplo: mensal, quinzenal, periódico, anual).</p> <p>Grau de influência: 1</p>
<p>6. Entrada de dados <i>on-line</i>: todas as transações são processadas em lote.</p> <p>Grau de influência: 0</p>
<p>7. Eficiência do usuário final: menus, movimentação automática do cursor.</p> <p>Grau de influência: 1</p>
<p>8. Atualização <i>on-line</i>: não há nenhuma atualização <i>on-line</i></p> <p>Grau de influência: 0</p>
<p>9. Complexidade de processamento: a aplicação não envolve processamento complexo.</p> <p>Grau de influência: 0</p>
<p>10. Reutilização: não houve preocupação com reutilização de código.</p> <p>Grau de influência: 0</p>
<p>11. Facilidade de instalação: o usuário não definiu considerações especiais, mas é necessário <i>setup</i> para instalação.</p> <p>Grau de influência: 1</p>
<p>12. Facilidade de operação: a aplicação minimiza a necessidade da manipulação de papel.</p> <p>Grau de influência: 1</p>
<p>13. Múltiplos locais: necessidade de múltiplos locais foi considerada no projeto, devendo operar nos mesmos ambientes idênticos de <i>software</i> e <i>hardware</i>.</p> <p>Grau de influência: 1</p>
<p>14. Facilidade de mudanças: não houve nenhuma preocupação específica.</p> <p>Grau de influência: 0</p>

(FONTE: adaptado de VAVASSORI, 2002)

Posteriormente, obtém-se o somatório dos níveis de influência (TDI). Logo, aplica-se a equação 1 (Vazquez, Simões e Albert, 2006):

$$\text{Fator de Ajuste} = (\text{TDI} * 0,01) + 0,65 \quad (1)$$

$$\text{Fator de Ajuste} = (5 * 0,01) + 0,65$$

Com a aplicação da equação 1 tem-se o valor **0,70** como fator de ajuste da aplicação em questão.

Passo 6: No último passo calcula-se o número de PF ajustados, para isso efetua-se o cálculo final de acordo com o tipo de contagem aplicado. Considera-se o valor dos PF não-ajustados e o fator de ajuste, obtidos anteriormente.

Para o projeto de aplicação aplica-se a equação 2:

$$\text{PFA} = \text{PF_NÃO_AJUSTADO} * \text{Fator de Ajuste} \quad (2)$$

$$\text{PFA} = 210 * 0,70$$

onde PFA = Número de PF do projeto de aplicação.

Tem-se o valor **147** como PF da aplicação em questão.

4. Considerações finais

Este artigo procurou demonstrar a aplicação da Análise de Pontos de Função, bem como, os passos para aplicação da mesma em um projeto de *software*. Tendo como objetivo principal servir como tutorial para aplicação da APF, pois há uma ausência de material específico e detalhado sobre o assunto.

Para a realização desta abordagem foi necessário além de listar os passos para a aplicação da métrica em questão, conceituar os elementos envolvidos no processo, para um melhor entendimento.

O valor encontrado na aplicação do processo de contagem pode ser utilizado para a obtenção de estimativas de prazos e custos, que proporcionam um melhor controle no desenvolvimento dos projetos.

Finalizando, espera-se ter proporcionado o entendimento e despertado o interesse pela métrica APF abordada neste artigo, como exemplo para ser aplicado em estimativas, sempre que necessário.

5. Referências bibliográficas

- DIAS, R. “Análise por Pontos de Função: Uma Técnica para Dimensionamento de Sistemas de Informação”. RESI - **Revista Eletrônica de Sistemas de Informação**, Florianópolis, 3. ed., 2003. Seção Artigos. Disponível em: <<http://www.inf.ufsc.br/resi>>. Acessado em: 24 abr. 2007.
- HAZAN, C. “Análise de Pontos de Função: Uma Abordagem Gerencial”. In: JORNADA DE ATUALIZAÇÃO EM INFORMÁTICA – JAI, 19., 2000, Curitiba. **Anais do XX Congresso da Sociedade Brasileira de Computação**. Curitiba: SBC, 2000. Disponível em: <<http://www.niee.ufrgs.br/SBC2000/eventos/jai.htm>>. Acessado em: 23 abr. 2007.
- HAZAN, C; STAA, A. V. “Análise e Melhoria de um Processo de Estimativas de Tamanho de Projetos de Software”. In: SIMPÓSIO INTERNACIONAL DE MELHORIA DE PROCESSO DE SOFTWARE – SIMPROS, 6., 2004, São Paulo. **Anais...** São Paulo: SENAC, 2004. Disponível em: <<http://www.lbd.dcc.ufmg.br/bdbcomp/servlet/Evento?id=101>>. Acessado em: 21 mai. 2007.

- SCOPPEL, L. “**Aplicações do Processo Unificado na Informatização do Setor SERED**”. 2004. 54 f. Trabalho de Conclusão de Curso (Bacharelado em Informática) – Departamento de Ciências Exatas e Tecnológicas, Universidade do Planalto Catarinense – UNIPLAC, Lages.
- VAZQUEZ, C. E.; SIMÕES, G. S.; ALBERT, R. M. “**Análise de Pontos de Função: Medição, Estimativas e Gerenciamento de Projetos de Software**”. 5. ed. São Paulo: Érica, 2006. 230 p.

ANEXOS

ANEXO 1 – REGRAS DE CONTAGEM DE TIPOS DE DADOS E DE TIPOS DE REGISTROS	69
ANEXO 2 – REQUISITOS FUNCIONAIS DO SISTEMA PARA REGISTRO E EXPEDIÇÃO DE DIPLOMAS E CERTIFICADOS.....	74
ANEXO 3 – PROJETO GRÁFICO DA CONSULTA EXTERNA GERAR DIPLOMA.....	77

ANEXO 1 – REGRAS DE CONTAGEM DE TIPO DE DADOS E DE TIPOS DE REGISTRO

Regras de contagem de tipos de dados para funções do tipo dado

Um tipo de dado é um campo único, reconhecido pelo usuário, não repetido.

Para que um determinado campo seja contado como um tipo de dado todas as regras de contagem devem ser válidas.

1. Conta-se um tipo de dado para cada campo único reconhecido pelo usuário e não repetido, mantido ou recuperado de um ALI ou AIE por meio da execução de um processo elementar. Exemplos:
 - No agendamento de um recebimento, a data de vencimento poderia estar armazenada em múltiplos campos (dia, mês e ano), mas continuaria a ser contada como um único tipo de dado.
 - Arquivo com várias ocorrências do mesmo campo: valor janeiro, valor fevereiro, ..., e valor dezembro; devem-se contar dois tipos de dados: um para o mês em questão e outro para o valor.
2. Quando duas aplicações mantêm ou referenciam o mesmo ALI/AIE, conte apenas os campos utilizados pela aplicação em análise.
 - Uma aplicação mantém ou referencia os seguintes campos de um arquivo: CPF e nome. Outra aplicação mantém ou referencia os seguintes campos do mesmo arquivo: nome, logradouro, cidade, estado e CEP. Para a primeira aplicação devem ser contados dois tipos de dados referentes ao arquivo e para a segunda aplicação devem ser contados cinco tipos de

dados para o mesmo arquivo.

- Para uma aplicação é necessário identificar cada parte do endereço do cliente, como logradouro, cidade, estado e CEP. Para outra aplicação, o mesmo endereço é relevante apenas no conjunto. A primeira aplicação deve contar quatro tipos de dados para o endereço e a segunda aplicação apenas um.
3. Conte um tipo de dado para cada campo solicitado pelo usuário para estabelecer um relacionamento com outro arquivo lógico (ALI ou AIE). Exemplos:
- Numa determinada aplicação as informações de entrada são mantidas no ALI Apontamento. A identificação da pessoa é parte das informações do Apontamento; ela serve para estabelecer um relacionamento entre o ALI Apontamento com o AIE Pessoa. Assim, são contados quatro tipos de dados no ALI Apontamento: identificação da pessoa (chave estrangeira), data, horário de entrada e horário de saída.
 - Caso a chave estrangeira seja composta por vários campos, todos eles devem ser contados como tipos de dados.

Regras de contagem de tipos de registro para funções do tipo dado

Um tipo de registro é um subgrupo de dados, identificado pelo usuário, componente de um Arquivo Lógico Interno ou Arquivo de Interface Interna.

Existem dois tipos de subgrupos:

- *Opcionais*: são aqueles em que o usuário tem a opção de não informar no processo elementar que cria ou adiciona dados ao arquivo.
- *Obrigatórios*: são aqueles que o usuário requer que sejam sempre utilizados pelo processo elementar que cria ou adiciona dados ao arquivo.

As seguintes regras devem ser utilizadas para determinar o número de tipos de registro de um ALI ou AIE:

- Conta-se um tipo de registro para cada subgrupo, obrigatório ou opcional,

de um ALI ou AIE, ou

- Se não houver nenhum subgrupo, conta-se o próprio ALI ou AIE como um tipo de registro.

Regras de contagem para arquivos referenciados para funções do tipo transação

Um arquivo referenciado é um Arquivo Lógico Interno (ALI) lido ou mantido pela função do tipo transação ou um Arquivo de Interface Externa (AIE) lido pela função do tipo transação.

As seguintes regras são válidas para a contagem de um arquivo referenciado. As duas primeiras, que tratam de atualização de arquivos, não são aplicáveis para Consultas Externas (CE):

- Conte um arquivo referenciado para cada ALI mantido;
- Conte apenas um arquivo referenciado para cada ALI que seja tanto mantido quanto lido;
- Conte um arquivo referenciado para cada ALI ou AIE lido durante o processamento.

Regras de contagem de tipos de dados para funções do tipo transação

Um tipo dado é um campo único, reconhecido pelo usuário, não repetido.

As seguintes regras devem ser válidas para a contagem de tipos de dados:

- Conte um tipo de dado para cada campo, não repetido e reconhecido pelo usuário, que entra ou sai pela fronteira da aplicação e necessário à conclusão do processo.
 - *Exemplo:* ao adicionar um cliente, o usuário fornece o nome do cliente e sua razão social.
- Se um campo tanto entra quanto sai pela fronteira da aplicação, deve ser

contado uma única vez.

- *Exemplo:* na tela de filtro de um relatório de pedidos o usuário pode informar por qual cliente deseja filtrar. O relatório exibe todos os dados de pedidos, inclusive o código do cliente informado na tela de filtro. Ao contar os tipos de dados desse processo elementar, o código do cliente deve ser contado uma única vez.
- Os campos que durante o processo elementar são recuperados ou derivados pelo sistema e armazenados em um ALI, mas não atravessam a fronteira da aplicação, *não* devem ser contados como tipo de dados.
 - *Exemplos:* quando um usuário agenda um novo recebimento, o sistema calcula um identificador único para esse registro. Esse campo não é apresentado ao usuário, apesar de ser atualizado no respectivo ALI. O código do recebimento não é contado como um tipo de dado para essa EE, uma vez que não atravessa a fronteira da aplicação;
 - Ao emitir uma nota fiscal, o sistema automaticamente atualiza no estoque a nova quantidade de itens do produto. Esse tipo de dado não deve ser contado;
 - Quando um cheque é impresso, um campo de situação no arquivo de compromissos é atualizado. Esse campo não deve ser contado, uma vez que não atravessa a fronteira da aplicação.
- Conte um único tipo de dado para a capacidade de envio para fora da fronteira da aplicação de uma mensagem de resposta do sistema, indicando um erro verificado durante o processamento, a confirmação da sua conclusão ou a verificação de seu prosseguimento.
 - *Exemplo:* ao registrar um compromisso em que o prazo entre a data de emissão e a data de vencimento seja inferior ao negociado com o fornecedor, ou ao tentar registrar um compromisso com valor zero, o sistema emite uma mensagem de erro. Deve-se contar um único tipo de dado para essas mensagens.
- Conte um tipo de dado para a capacidade de especificar uma ação a ser

tomada. Mesmo que haja múltiplos meios de ativar o mesmo processo, deve ser contado apenas um tipo de dado.

- *Exemplos:* a seleção de um parcela para recebimento pode ser feita pela barra de espaço ou utilizado um *check box*;
- Para salvar os dados da tela, o usuário pode clicar no botão Salvar, usar a tecla de atalho CTRL+S ou usar a opção Arquivo>Salvar do menu. Deve-se contar um único tipo de dado para esses comandos.
- *Não* conte literais como tipo de dados.
 - *Exemplos:* títulos de relatórios, identificação de teclas, cabeçalhos de colunas, nomes de campos são exemplos de literais e não devem ser contados.
- *Não* conte variáveis de paginação ou campos automáticos gerados pelo sistema.
 - *Exemplos:* número de página, informações de posicionamento (25 de 102), comandos de paginação como anterior, próximo ou setas de rolagem em aplicações com interface gráfica, campos de data e hora do sistema.

ANEXO 2 – REQUISITOS FUNCIONAIS DO SISTEMA PARA REGISTRO E EXPEDIÇÃO DE DIPLOMAS E CERTIFICADOS

QUADRO 15 - Requisito manter cadastro de curso de graduação

F1 Manter cadastro CursoGrad		Oculto ()		
Descrição: Permitir inserção, alteração, exclusão, e consulta de cursoGrad				
Requisito Não Funcional:				
Nome	Restrição	Categoria	Desejável	Permanente
NF1.1 Informações sobre cursoGrad	*nomeCursoGrad, categoriaCursoGrad, habilitacao, autorizacao, reconhecimentoProvisorio, reconhecimentoOficial, resolucao.	Especificação	Não	Sim

QUADRO 16 - Requisito manter cadastro de aluno

F2 Manter cadastro de Aluno		Oculto ()		
Descrição: Permitir inserção, alteração, exclusão, e consulta de Aluno				
Requisito Não Funcional:				
Nome	Restrição	Categoria	Desejável	Permanente
NF2.1 Informações sobre o aluno	*nomeAluno, RG, naturalidade, cursoGrad, nacionalidade, dataConclusao, dataNascimento, sexoAluno, dataEncerramentoProcesso, numProcesso	Especificação	Não	Não

QUADRO 17 - Requisito manter cadastro de reitor

F3 Manter cadastro de Reitor		Oculto ()		
Descrição: Permitir inserção, alteração, exclusão, e consulta de Reitor				
Requisito Não Funcional:				
Nome	Restrição	Categoria	Desejável	Permanente
NF3.1 Informações sobre o reitor	* nomeReitor, *cargoReitor, *sexoReitor, *dataInicialReitor, *dataFinalReitor	Especificação	Não	Não

QUADRO 18 - Requisito manter cadastro de pró-reitor de ensino

F4 Manter cadastro de ProReitorEnsino		Oculto ()		
Descrição: Permitir inserção, alteração, exclusão, e consulta de ProReitorEnsino				
Requisito Não Funcional:				
Nome	Restrição	Categoria	Desejável	Permanente
NF4.1 Informações sobre o ProReitorEnsino	*nomePRENS, *cargoPRENS, *sexoPRENS, *dataInicialPRENS, *dataFinalPRENS	Especificação	Não	Não

QUADRO 19 - Requisito manter cadastro de secretário acadêmico

F5 Manter cadastro de SecretarioAcademico		Oculto ()		
Descrição: Permitir inserção, alteração, exclusão, e consulta de SecretarioAcademico				
Requisito Não Funcional:				

Nome	Restrição	Categoria	Desejável	Permanente
NF5.1 Informações sobre o SecretarioAcademico	*nomeSA, *cargoSA, *sexoSA, *dataInicialSA, *dataFinalSA	Especificação	Não	Não

QUADRO 20 - Requisito manter cadastro de registro

F6 Manter cadastro de Registro		Oculto ()		
Descrição: Permitir inserção, alteração, exclusão, e consulta de Registro				
Requisito Não Funcional:				
Nome	Restrição	Categoria	Desejável	Permanente
NF6.1 Informações sobre o Registro	*numFolha, *numLivro, *dataRegistro, *numRegistro	Especificação	Não	Não

QUADRO 21 - Requisito manter cadastro de chefe do SERED

F7 Manter cadastro de ChefeSERED		Oculto ()		
Descrição: Permitir inserção, alteração, exclusão, e consulta de ChefeSered				
Requisito Não Funcional:				
Nome	Restrição	Categoria	Desejável	Permanente
NF7.1 Informações sobre o ChefeSERED	*nomeCSERED, *cargoCSERED, *sexoCSERED, *dataInicialCSERED, *dataFinalCSERED	Especificação	Não	Não

QUADRO 22 - Requisito manter cadastro de autorização

F8 Manter cadastro de autorização		Oculto ()		
Descrição: Permitir inserção, alteração, exclusão, e consulta de autorização				
Requisito Não Funcional:				
Nome	Restrição	Categoria	Desejável	Permanente
NF8.1 Informações sobre o autorização	*numAutorizacao, *orgaoEmissor, *data	Especificação	Não	Não

QUADRO 23 - Requisito manter cadastro de reconhecimento

F9 Manter cadastro de reconhecimento		Oculto ()		
Descrição: Permitir inserção, alteração, exclusão, e consulta de reconhecimento				
Requisito Não Funcional:				
Nome	Restrição	Categoria	Desejável	Permanente
NF9.1 Informações sobre o reconhecimento	*NumReconhecimentoI, *orgaoEmissor, *data, *tipo	Especificação	Não	Não

QUADRO 24 - Requisito manter cadastro de categoria de curso de graduação

F11 Manter cadastro de categoria		Oculto ()		
Descrição: Permitir inserção, alteração, exclusão, e consulta de categoria				
Requisito Não Funcional:				
Nome	Restrição	Categoria	Desejável	Permanente
NF11.1 Informações sobre o categoria	*nomeCategoria	Especificação	Não	Não

QUADRO 25 - Requisito manter cadastro de habilitação de curso de graduação

F12 Manter cadastro de habilitação		Oculto ()		
Descrição: Permitir inserção, alteração, exclusão, e consulta de habilitação				
Requisito Não Funcional:				
Nome	Restrição	Categoria	Desejável	Permanente
NF12.1 Informações sobre o habilitação	nomeHabilitacao	Especificação		

QUADRO 26 - Requisito manter cadastro de resolução de curso de graduação

F13 Manter cadastro de resolução		Oculto ()		
Descrição: Permitir inserção, alteração, exclusão, e consulta de resolução				
Requisito Não Funcional:				
Nome	Restrição	Categoria	Desejável	Permanente
NF13.1 Informações sobre o resolução	*NumResolucao, *orgaoEmissor, *data	Especificação		

QUADRO 27 - Requisito gerar diploma

F14 Gerar diploma		Oculto ()		
Descrição: Permitir a elaboração de um diploma associando as informações que devem constar no mesmo				
Requisito Não Funcional:				
Nome	Restrição	Categoria	Desejável	Permanente
NF14.1 Lista de cursoGrad	Deve ser possível selecionar o cursoGrad a partir de uma lista	Interface	Sim	Não
NF14.2 Lista de aluno	Deve ser possível selecionar o aluno a partir de uma lista que seja vinculado ao curso selecionado	Interface	Sim	Não
NF14.3 Visualização de reitor	Deve ser possível visualizar o reitor a partir da data de encerramento do processo (NF 14.1)	Interface	Sim	Não
NF14.4 Visualização de ProReitorEnsino	Deve ser possível visualizar o pró-reitor de ensino a partir da data de registro (NF 14.6)	Interface	Sim	Não
NF14.5 Visualização de SecretarioAcademico	Deve ser possível visualizar o secretário acadêmico a partir da data de encerramento do processo (NF 14.1) Só haverá SecretarioAcademico se o curso tiver habilitação	Interface	Sim	Não
NF14.6 Apresentação de Registro	Deve ser possível gerar novo número de registro, data, folha e livro a cada novo diploma	Interface	Sim	Não

ANEXO 3 – PROJETO GRÁFICO DA CONSULTA EXTERNA GERAR DIPLOMA

The image shows a software window titled "Gerar Diploma" with a blue title bar. The window contains several input fields and sections for data entry:

- Curso Graduação:** A dropdown menu with a "+" button.
- Categoria:** A text input field.
- Habilitação:** A text input field.
- Autorização:** A section containing three input fields: "Número", "Órgão", and "Data".
- Resolução:** A section containing three input fields: "Número", "Órgão", and "Data".
- Reconhecimento Oficial:** A section containing three input fields: "Número", "Órgão", and "Data".
- Reconhecimento Provisório:** A section containing three input fields: "Número", "Órgão", and "Data".
- Aluno:** A dropdown menu with a "+" button.
- Número do Processo:** A text input field.
- Nome do Aluno:** A text input field.
- RG:** A text input field.
- Sexo:** Two radio buttons labeled "M" and "F".
- Naturalidade:** A text input field.
- Nacionalidade:** A text input field.
- Data de Encerramento do Processo:** A text input field.
- Data de Colação de Grau:** A text input field.
- Registro:** A section containing four input fields: "Número do Registro", "Livro", "Folha", and "Data".
- Chefe do SERED:** A text input field with a "+" button.
- Autoridades:** A section containing three input fields: "Reitor", "Pró-Reitor de Ensino", and "Secretário Acadêmico", each with a "+" button.

At the bottom of the window, there are three buttons: "Confirmar", "Cancelar", and "Fechar".

(FONTE: SCOPPEL, 2004)

