

**CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DO ESPÍRITO SANTO
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS**

**FERRAMENTA DE ESTIMATIVAS DE PROJETOS DE SOFTWARE
BASEADA EM PONTOS DE FUNÇÃO**

**CIRO XAVIER MARETTO
WILLIAM DANIEL LESSA JUNGER**

**SERRA/ES
2008**

CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DO ESPÍRITO SANTO
CURSO SUPERIOR DE TECNOLOGIA EM SISTEMAS DE INFORMAÇÃO

CIRO XAVIER MARETTO
WILLAM DANIEL LESSA JUNGER

FERRAMENTA DE ESTIMATIVAS DE PROJETOS DE SOFTWARE
BASEADA EM PONTOS DE FUNÇÃO

Trabalho apresentado à Coordenadoria de Sistemas de Informação do Centro Federal de Educação Tecnológica do Espírito Santo - Unidade Descentralizada de Serra como requisito parcial para a obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientadora: Prof^a. M.Sc Vanessa Battestin Nunes

Co-orientador: Prof^o. M.Sc Elton Siqueira Moura

**CIRO XAVIER MARETTO
WILLIAM DANIEL LESSA JUNGER**

**FERRAMENTA DE ESTIMATIVAS DE PROJETOS DE
SOFTWARE BASEADA EM PONTOS DE FUNÇÃO**

Trabalho apresentado à Coordenadoria do Curso de Sistemas de Informação do Centro Federal de Educação Tecnológica do Espírito Santo - Unidade Descentralizada de Serra como requisito parcial para a obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Aprovado em 02 de dezembro de 2008.

COMISSÃO EXAMINADORA

Prof^a. M.Sc Vanessa Battestin
Centro Federal de Educação Tecnologia do Espírito Santo
Orientadora

Prof^o. M.Sc Elton Siqueira Moura
Centro Federal de Educação Tecnologia do Espírito Santo
Co-Orientador

Prof^a. M.Sc Fabiano Borges Ruy
Centro Federal de Educação Tecnologia do Espírito Santo

Guilherme Siqueira Simões
Fatto Consultoria e Sistemas

DECLARAÇÃO DOS AUTORES

Declaramos, para fins de pesquisa acadêmica, didática e técnico-científica, que o presente Trabalho de Conclusão de Curso pode ser parcialmente utilizado desde que se faça referência à fonte e aos autores.

Serra, 02 de Dezembro de 2008.

CIRO XAVIER MARETTO
WILLIAM DANIEL LESSA JUNGER

*A Deus e todos que fazem parte
da família e rede de amigos, vocês
estão no meu coração.*

Ciro Xavier Maretto

*Aos familiares, amigos, professores
e a Deus que colocou essas pessoas
maravilhosas em minha vida.
Amo todos vocês.*

William Daniel Lessa Junger

É com grande satisfação que chego ao final do meu curso de graduação. Por isso agradeço aos que me apoiaram para o término desse trabalho. Certamente quero agradecer primeiramente a Deus, por ter permitido que eu pudesse chegar até aqui, e ter me protegido e abençoado durante toda essa jornada.

Agradeço a minha mãe, meu pai e minha irmã, pois eles me deram todo o suporte necessário e foram grandes incentivadores para minha grande dedicação aos estudos.

Quero agradecer aos colegas que apresentaram paciência, atenção e confiança em todos os trabalhos e momentos que passamos juntos. Novos laços de amizade e profissionais puderam ser feitos nesse curso, dentre eles o William, um cara inteligente e responsável que consegue encarar trabalho e duas faculdades ao mesmo tempo, um real exemplo de dedicação; o Henrique, que sempre dispôs apoio e bons conselhos; e muitos outros colegas, como o Matheus, Cintya e seu namorado João Marcos, que também é meu primo e colega de faculdade.

Os professores do CEFET-ES tiveram um importante papel para a formação de meu conhecimento, e por isso venho agradecê-los, em especial aos professores e orientadores Vanessa e Elton, que foram importantes presenças e nos ajudaram a traçar um bom caminho para a execução deste trabalho. Agradeço também ao professor Fabiano, que esteve sempre disposto a tirar dúvidas e sugerir novas idéias.

Não teria como deixar de agradecer, e muito, a empresa Fatto que teve uma importante cooperação na realização deste trabalho. Desde o início contamos com o apoio do Guilherme, sempre com muita boa vontade e grande experiência, sanou muitas de nossas dúvidas, além de ter nos disponibilizado um curso de capacitação referente ao tema do trabalho.

Meu muito obrigado a todos os que de alguma forma influenciaram para que esse trabalho se tornasse realidade, desejo a vocês tudo de bom.

Ciro Xavier Maretto

Primeiramente a Deus, que me deu a vida, força e sabedoria os quais permitiram que eu chegasse à conclusão deste curso.

À minha mãe, Marina, que me ensinou os valores da vida e me incentivou em todos os momentos dessa caminhada. Você é o motivo de todo o meu esforço e a fortaleza de onde busco minhas forças para prosseguir. Agradeço ao meu pai, William, que tenho certeza ter me dado forças lá de cima e que estaria orgulhoso dessa vitória. À minha irmã, Danielle, pelo amor e compreensão durante todo esse período do curso. Obrigado por me deixar utilizar o computador nos momentos de maior dificuldade. Aos demais familiares, pelo incentivo e ajuda. Agradeço a Deus por ter essa família maravilhosa. Amo todos vocês.

À minha futura esposa, Roberta, pela paciência e companheirismo. Obrigado por me incentivar e me entender nos momentos de estudo. Sem você ao lado não conseguiria vencer esse desafio. A você dedico todo o meu amor.

Ao Ciro, pela amizade e responsabilidade demonstrada durante todo o desenvolvimento deste trabalho. Continue sendo esse cara humilde e brincalhão, com uma inteligência fora do comum e um caráter digno de respeito. Ao Luke que esteve presente em todas as reuniões pelo skype, com os seus latidos de incentivo. Aos meus amigos de curso Henrique, Felipe, Gabriel, Renata, Matheus e todos os outros que nos acompanharam desde o início, agradeço pela cooperação e incentivo.

Aos orientadores, Vanessa e Elton, que demonstraram os grandes profissionais que são através de suas contribuições e responsabilidade. Ao professor Fabiano pela ajuda nessa reta final. Aos demais professores pela contribuição acadêmica e profissional proporcionada.

Ao Guilherme, pela presteza e cooperação durante todo o desenvolvimento deste trabalho. Obrigado por enriquecer o meu conhecimento e contribuir para o meu crescimento profissional.

A todos àqueles que direta ou indiretamente colaboraram para o alcance desta conquista.

William Daniel Lessa Junger

*"O degrau de uma escada não serve
simplesmente para que alguém
permaneça em cima dele, destina-se a
sustentar o pé de um homem pelo tempo
suficiente para que ele coloque o outro um
pouco mais alto."*

Thomas Huxley

*A vida só pode ser entendida
olhando-se para trás.
Mas só pode ser vivida
olhando-se para frente.*

S. Kierkegaard

RESUMO

O objetivo deste trabalho foi a criação de uma ferramenta livre capaz de realizar contagens e estimativas com a técnica de Análise de Pontos por Função, gerando uma base histórica de projetos já realizados. Estes dados históricos por sua vez, podem ser utilizados no sistema para agilizar e aumentar a precisão das estimativas dos projetos de software de uma organização. A necessidade surgiu com a dificuldade de equipes de desenvolvimento de software realizarem estimativas confiáveis. O trabalho foi desenvolvido seguindo metodologias e técnicas estudadas durante o curso.

Palavras-Chave: estimativas, Análise de Pontos de Função, desenvolvimento de software, dados históricos.

ABSTRACT

This work goal was the creation of a free tool able to perform counts and estimates with the technique of Function Points Analysis, in addition to generating a historical basis for projects already undertaken. These historical data in turn, can be used in the system to streamline and increase the accuracy of estimates of software projects within an organization. The need arose with the difficulty of software development teams to achieve reliable estimates. The study was conducted using methods and techniques studied during the course.

Keywords: estimates, Function Points Analysis, software development, historical data.

SUMÁRIO

CAPÍTULO 1	13
INTRODUÇÃO	13
1.1 CONTEXTO E OBJETIVOS DO TRABALHO	14
1.2 METODOLOGIA.....	15
1.2.1 <i>O Paradigma Orientado a Objetos.....</i>	<i>15</i>
1.2.2 <i>Desenvolvimento Iterativo</i>	<i>16</i>
1.2.3.1 Levantamento e Especificação dos Requisitos.....	18
1.2.3.2 Análise Orientada a Objetos	18
1.2.3.3 Projeto Orientado a Objetos.....	20
1.2.3.4 Implementação e Testes	21
1.3 ORGANIZAÇÃO DO TEXTO	21
CAPÍTULO 2	23
ESTIMATIVAS DE PROJETOS DE SOFTWARE	23
2.1 TÉCNICAS DE ESTIMATIVAS DE TAMANHO	23
2.1.1 <i>Histórico da Análise de Pontos de Função.....</i>	<i>24</i>
2.1.2 <i>Análise de Pontos de Função no Brasil.....</i>	<i>26</i>
2.1.3 <i>Outras Técnicas.....</i>	<i>27</i>
2.1.4 <i>Vantagens e Desvantagens da Análise de Pontos de Função.....</i>	<i>28</i>
2.2 PROCESSO DA CONTAGEM DE PONTOS DE FUNÇÃO	29
2.2.1 <i>Definir o Tipo de Contagem.....</i>	<i>30</i>
2.2.1.1 Projeto de Desenvolvimento	30
2.2.1.2 Projeto de Melhoria	30
2.2.1.3 Aplicação	30
2.2.2 <i>Identificar o Escopo da Contagem e a Fronteira de Aplicação</i>	<i>30</i>
2.2.3 <i>Identificar as Funções do sistema</i>	<i>31</i>
2.2.3.1 Funções do Tipo dados	31
2.2.3.2 Funções do Tipo Transação	31
2.2.3.3 Determinação da Complexidade das Funções	32
2.2.4 <i>Pontos de Função não ajustados.....</i>	<i>33</i>
2.2.5 <i>Determinar os pontos de função ajustados.....</i>	<i>33</i>
2.3 DIFERENTES TIPOS DE ABORDAGEM.....	34
2.3.1 <i>Contagem Indicativa.....</i>	<i>35</i>
2.3.2 <i>Contagem Estimativa.....</i>	<i>35</i>
2.4 BASE HISTÓRICA DOS PROJETOS.....	35
2.5 INDICADORES COM BASE EM PONTOS DE FUNÇÃO	36
2.5.1 <i>Produtividade e Taxa de Entrega.....</i>	<i>36</i>
2.5.2 <i>Esforço.....</i>	<i>38</i>
2.5.3 <i>Qualidade</i>	<i>38</i>
2.5.4 <i>Duração.....</i>	<i>38</i>
2.5.5 <i>Custo.....</i>	<i>39</i>
2.6 FERRAMENTAS SIMILARES	40
2.6.1 <i>Scope - Project Sizing Software.....</i>	<i>40</i>
2.6.2 <i>FPRecorder</i>	<i>42</i>
2.6.3 <i>APFPlus.....</i>	<i>43</i>
2.6.4 <i>WinFPA</i>	<i>44</i>
2.6.5 <i>Function Point Modeler.....</i>	<i>45</i>
2.6.6 <i>Outras ferramentas de estimativa.....</i>	<i>46</i>
2.7 CONCLUSÕES DO CAPÍTULO	48
CAPÍTULO 3	50
ESPECIFICAÇÃO DE REQUISITOS.....	50
3.1 ESPECIFICAÇÃO DE REQUISITOS FUNCIONAIS	50
3.1.1 <i>Descrição do Mini-mundo.....</i>	<i>50</i>

3.1.2	<i>Modelagem dos Requisitos</i>	53
3.1.2.1	<i>Diagrama de Pacotes</i>	53
3.1.2.2	<i>Pacote de Registros</i>	54
3.1.2.3	<i>Pacote de Projeto</i>	56
3.1.2.4	<i>Pacote de Estimativa</i>	59
3.2	ESPECIFICAÇÃO DE REQUISITOS NÃO FUNCIONAIS	61
3.2.1	<i>Requisitos de Processo</i>	61
3.2.2	<i>Requisitos de Produto</i>	62
3.2.2.1	<i>Segurança</i>	62
3.2.2.2	<i>Usabilidade</i>	62
3.3	CONCLUSÕES DO CAPÍTULO	62
CAPÍTULO 4		63
ANÁLISE DO SISTEMA.....		63
4.1	MODELAGEM ESTÁTICA.....	63
4.2	MODELAGEM DO COMPORTAMENTO.....	68
4.2.1	<i>Diagrama de Atividade do sistema</i>	69
4.2.2	<i>Diagrama de Seqüência – Incluir Contagem</i>	70
4.2.3	<i>Diagrama de Seqüência – Incluir Estimativa</i>	71
4.3	CONCLUSÕES DO CAPÍTULO	72
CAPÍTULO 5		73
PROJETO DO SISTEMA		73
5.1	PLATAFORMA DE IMPLEMENTAÇÃO	73
5.2	ARQUITETURA DO SISTEMA.....	74
5.3	CAMADAS DA ARQUITETURA	76
5.3.1	<i>Camada de Domínio do Problema</i>	76
5.3.2	<i>Camada de Gerência de Dados</i>	78
5.3.3	<i>Camada de Gerência de Tarefas</i>	79
5.3.4	<i>Camada de Interface</i>	82
5.4	PADRÕES DE PROJETO	83
5.5	CONCLUSÕES DO CAPÍTULO.....	84
CAPÍTULO 6		85
IMPLEMENTAÇÃO E TESTES		85
6.1	IMPLEMENTAÇÃO DO PROTÓTIPO	85
6.1.1	<i>Tela Inicial</i>	86
6.1.2	<i>Telas de Registro</i>	87
6.1.3	<i>Telas de Aplicação</i>	91
6.1.4	<i>Telas de Projeto</i>	92
6.1.5	<i>Telas de Contagens</i>	94
6.1.6	<i>Telas de Estimativas</i>	98
6.2	CONCLUSÕES DO CAPÍTULO.....	100
CAPÍTULO 7		101
CONSIDERAÇÕES FINAIS.....		101
REFERÊNCIAS BIBLIOGRÁFICAS		103
ANEXO A.....		106
DESCRIÇÃO DOS CASOS DE USO.....		106
ANEXO B.....		107
QUESTIONÁRIO DO ISBSG.....		107

Capítulo 1

Introdução

Na última década, houve um grande aumento no interesse das empresas de desenvolvimento de software por modelos de qualidade, tais como ISO ou CMM, para garantir destaque no mercado. Com isso, cresceu a necessidade de ter uma técnica que permita a mensuração do projeto, geração de estimativas de custo, prazo, recursos e produtividade, para auxiliar o gerente de projetos de desenvolvimento de softwares em suas tomadas de decisão.

Medir o tamanho de um software é uma alternativa comumente utilizada, tendo em vista que essa técnica está fortemente relacionada com o esforço despendido em um projeto, sendo assim um ponto de partida para o planejamento.

Segundo (PRESSMAN, 2002), a medição é fundamental para qualquer atividade de engenharia e a engenharia de software não é exceção. A medição permite obter um mecanismo para avaliação objetiva. Como Lord Kelvin disse (apud PRESSMAN, 2002):

“Você tem algum conhecimento sobre o que você fala, quando pode medir e expressar isso em números; mas quando você não pode medir, quando não pode expressar isso em números, seu conhecimento é fraco e insatisfatório: pode ser o princípio do conhecimento, mas, na sua opinião, você provavelmente está no estágio inicial de uma ciência.”

Faz-se necessário realizar medições para atingir melhores estimativas em projetos de desenvolvimento de software, auxiliando o gerente de projetos com dados mais próximos da realidade. Através de estimativas precisas o gerente pode tomar decisões importantes para o projeto, conseguindo aperfeiçoamentos reais ao longo do tempo.

Porém, realizar estimativas não é uma tarefa fácil de ser executada. Para que ela seja confiável, é preciso que a medição seja bem feita. No caso de métricas orientadas a função, as funcionalidades do sistema precisam estar bem definidas para que o resultado seja satisfatório.

Atualmente existem diferentes técnicas para medir o tamanho de um software, destacando-se entre elas a quantidade de LOC (Lines of Code), a UCP (Pontos por Caso de Uso) e a APF (Análise de Pontos de Função).

A técnica de Análise de Pontos de Função possui algumas vantagens quanto às demais, destacando entre elas o fato da técnica poder ser aplicada nas fases iniciais de um projeto bastando apenas uma boa definição dos requisitos do software, sendo que tais requisitos podem ser expressos de diferentes formas. Além disso, a medição da APF sempre chegará ao mesmo resultado, diferentemente da UCP e a LOC, tendo em vista que não existe um padrão único para escrita dos casos de uso e a quantidade de linhas de código difere de acordo com a linguagem (FATTO, 2008).

Este trabalho propõe a construção de uma ferramenta que auxilie equipes de desenvolvimento de software a fazer estimativas de seus projetos através da análise de pontos de função, produzindo indicadores de produtividade, tempo, custo, qualidade e possibilitando ainda a geração de uma base de dados históricos.

1.1 Contexto e Objetivos do Trabalho

Construir uma ferramenta livre que auxilie gerentes de projetos a realizar estimativas com base em pontos por função em seus projetos de construção de software, gerando uma base histórica de projetos já realizados. Dessa forma, as estimativas poderão ser cada vez mais precisas, auxiliando-os em suas tomadas de decisão. Para se atingir este objetivo geral, os seguintes objetivos específicos foram definidos:

- Auxiliar a tomada de decisão por parte do gerente da equipe de desenvolvimento através dos dados históricos dos projetos já realizados;
- Utilizar a análise de pontos de função para estimar e medir o tamanho do software, servindo como base para as demais estimativas;
- Adotar o manual de práticas de contagem de análise de pontos de função estabelecido pelo órgão internacional (IFPUG).
- Incentivar a utilização da técnica de APF para realizar estimativas de projetos de software;

- Auxiliar equipes de desenvolvimento na realização dos cálculos de estimativa de seus projetos;
- Fornecer mecanismo de utilização da base de dados histórica, para auxílio em novas estimativas;
- Calcular a produtividade de equipes de desenvolvimento de software, através da utilização da base de dados histórica de projetos;
- Obter indicador de qualidade do software, através do cálculo da quantidade de defeitos por pontos de função.

1.2 Metodologia

A abordagem para a construção de um software, que segue um conjunto determinado de tarefas é conhecida como processo de desenvolvimento. Neste trabalho foram usados os conceitos de paradigma OO, ciclo de vida iterativo, envolvendo as atividades de levantamento e especificação de requisitos, análise, projeto, implementação e testes.

1.2.1 O Paradigma Orientado a Objetos

Um paradigma de desenvolvimento de software agrupa métodos e técnicas que seguem um mesmo conjunto de princípios. O paradigma Orientado a Objetos, também chamado de paradigma OO, procura modelar os objetos e conceitos do mundo real envolvidos no domínio do problema, de forma abstrata em modelos computacionais, chamados de Classes. As classes são moldes para os objetos, e estes quando criados possuem os mesmos atributos e métodos (comportamentos) da classe (SOUZA, 2006). Segue abaixo alguns benefícios desse paradigma:

- Apoio na reutilização de código;
- Possui recursos como herança e polimorfismo;
- Facilita a abstração do código e conceitos;
- Maior consistência na fase de Análise;
- Guia toda a tarefa de modelagem;
- Auxilia a administrar a complexidade.

A Orientação a Objetos não é mágica. Para uma boa utilização é preciso que seja aplicada com disciplina e em conjunto com outras técnicas da Engenharia de Software (SOUZA, 2006).

1.2.2 Desenvolvimento Iterativo

O desenvolvimento de um sistema envolve diversas etapas, onde em cada uma, diversas atividades devem ser desenvolvidas. O modelo de ciclo de vida do desenvolvimento de software consiste na forma em como essas atividades e etapas estão organizadas. Há diversos modelos de ciclo de vida, a diferença entre um e outro está basicamente na maneira como as diversas etapas são encadeadas. Dentre os modelos existentes é destacado neste trabalho o Modelo de Desenvolvimento Iterativo.

O ciclo de vida iterativo é baseado em incrementos e refinamentos de um sistema por meio de várias iterações, até que um sistema adequado tenha sido desenvolvido. O sistema cresce incrementalmente ao longo do tempo, iteração por iteração, razão pela qual esta abordagem também é conhecida como desenvolvimento iterativo e incremental, ou ainda como desenvolvimento evolucionário em espiral, como era inicialmente conhecida (LARMAN, 2003).

Como podemos observar na figura 1.1, no desenvolvimento iterativo cada uma das fases inclui suas próprias atividades de levantamento e especificação de requisitos, análise, projeto, implementação e testes, tendo como resultado ao fim de cada fase um sistema executável, mas incompleto, até que se chegue ao último ciclo.

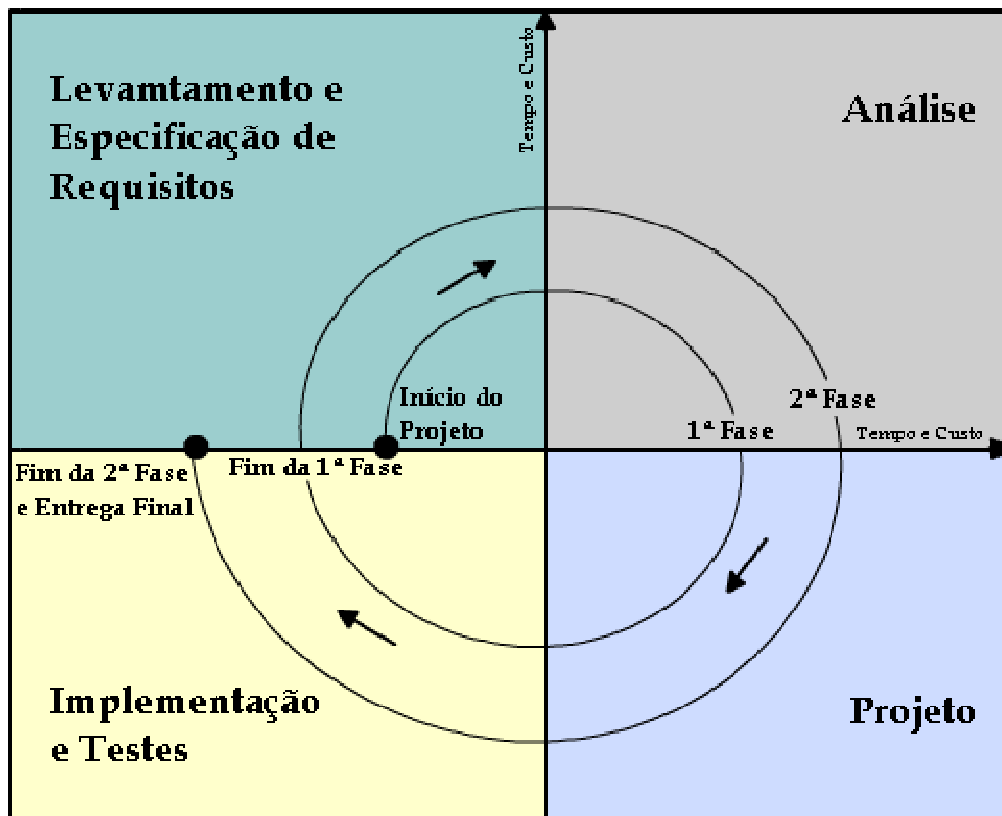


Figura 1.1 – Exemplo Ilustrativo do Modelo de Desenvolvimento Iterativo

Segundo Larman (2003), dentre os benefícios desse modelo, podemos destacar:

- Facilidade para administrar a complexidade e os riscos do sistema;
- O aprendizado obtido em uma iteração pode ser usado para melhorar o próprio processo de desenvolvimento nas próximas iterações;
- Realimentação e envolvimento do usuário mais freqüente, levando a um sistema cada vez mais refinado, que atenda de forma mais adequada às reais necessidades dos interessados no projeto.

Neste trabalho, na 1ª fase o sistema foi desenvolvido com o mínimo de requisitos, mas tendo toda estrutura para poder calcular os diferentes tipos de contagem de pontos de função. Na 2ª fase, foram adicionadas outras funcionalidades para atender aos objetivos deste projeto, que contempla a realização de estimativas com base em dados históricos. Sendo assim, poderão ocorrer novas fases de desenvolvimento em projetos futuros, incentivando a melhoria do software.

Acredita-se que esse modelo é potencialmente a melhor opção para pequenos e médios sistemas com tempo de desenvolvimento razoavelmente curto. Para muitos sistemas de grande porte, naturalmente, não existe apenas um modelo de desenvolvimento que possa ser utilizado. Processos diferentes são utilizados para desenvolver diferentes partes do sistema (SOMMERVILLE, 2003).

1.2.3.1 Levantamento e Especificação dos Requisitos

No que tange às atividades técnicas do desenvolvimento de software, a primeira coisa a ser feita é capturar os requisitos que o sistema a ser desenvolvido tem de tratar. Um completo entendimento dos requisitos do software é essencial para o sucesso de um esforço de desenvolvimento de software (FALBO, 2005).

Nesta fase, os usuários, clientes e especialistas de domínio são identificados e trabalham juntos para descobrir, articular e entender o domínio da aplicação, os processos de negócio específicos, as necessidades que o software deve atender e os problemas e deficiências dos sistemas atuais. Os diferentes pontos de vista dos participantes do processo, bem como as oportunidades de melhoria e restrições existentes, os problemas a serem resolvidos, o desempenho requerido e as restrições também devem ser levantados (FALBO, 2005).

Com o intuito de agrupar as informações levantadas e obter uma visão dos requisitos encontrados foi utilizada a modelagem de casos de uso, que permite definir as funções de aplicação que o sistema irá disponibilizar. Além disso, os casos de uso serão utilizados durante a análise para descrever a funcionalidade do sistema.

No capítulo 03 deste trabalho é apresentada a especificação de requisitos do projeto.

1.2.3.2 Análise Orientada a Objetos

A fase de análise enfatiza uma investigação do problema e dos requisitos. Essa etapa é importantíssima, pois ninguém é capaz de entender com perfeição um

problema usual de sistemas de informação na primeira vez que o olha. Assim, esse tempo de análise deve ser bem aproveitado na investigação do problema (WAZLAWICK, 2004).

Segundo Macoratti (2007), "A análise de sistemas no mundo orientado a objeto é feita analisando-se os objetos e os eventos que interagem com tais objetos". A modelagem é feita através da linguagem UML (*Unified Modeling Language*) que foi aprovada pela OMG (*Object Management Group*) em novembro de 1997. O objetivo da criação dessa linguagem foi criar um padrão para o desenvolvimento orientado a objetos, que torna possível a especificação, visualização, documentação e construção de artefatos que podem ser usados durante todo o processo de desenvolvimento, não sendo imprescindível o uso de uma única tecnologia (MACORATTI, 2008).

Segundo Larman (2003), a notação UML pode ser usada para três perspectivas e tipos de modelos:

1. **Perspectiva Conceitual** – os diagramas são interpretados como descrevendo coisas em uma situação do mundo real ou domínio de interesse.
2. **Perspectiva de Especificação (software)** – os diagramas (usando a mesma notação da perspectiva conceitual) descrevem abstrações de software ou componentes com especificações e interfaces, mas nenhum comprometimento com uma implementação particular (por exemplo, não especificamente uma classe em C# ou Java).
3. **Perspectiva de Implementação (software)** – os diagramas descrevem implementações de software em uma tecnologia particular (tal como Java)

Na fase de análise o mais importante é a perspectiva conceitual. O essencial é o entendimento do domínio do problema.

Na análise deste projeto foi feita a modelagem conceitual do sistema, a partir do levantamento de requisitos. No capítulo 04 está descrito a especificação da análise do sistema, contendo o diagrama de pacotes, os diagramas de classes, o dicionário de dados, os diagramas de seqüência e um diagrama de atividade.

1.2.3.3 Projeto Orientado a Objetos

A fase de projeto indica como fazer o que foi identificado na análise. Durante o projeto orientado a objetos, há uma ênfase na definição dos objetos de software e como eles colaboram para a satisfação dos requisitos (LARMAN, 2003).

A atividade de projeto envolve a modelagem de como o sistema será implementado, com a adição dos requisitos não funcionais aos modelos construídos na análise, como ilustra a figura 1.2. Assim, o objetivo do projeto é incorporar a tecnologia aos requisitos identificados, projetando o que será construído na implementação. Para tal, é necessário conhecer a tecnologia disponível e as facilidades do ambiente de software no qual o sistema será implementado (FALBO, 2005).

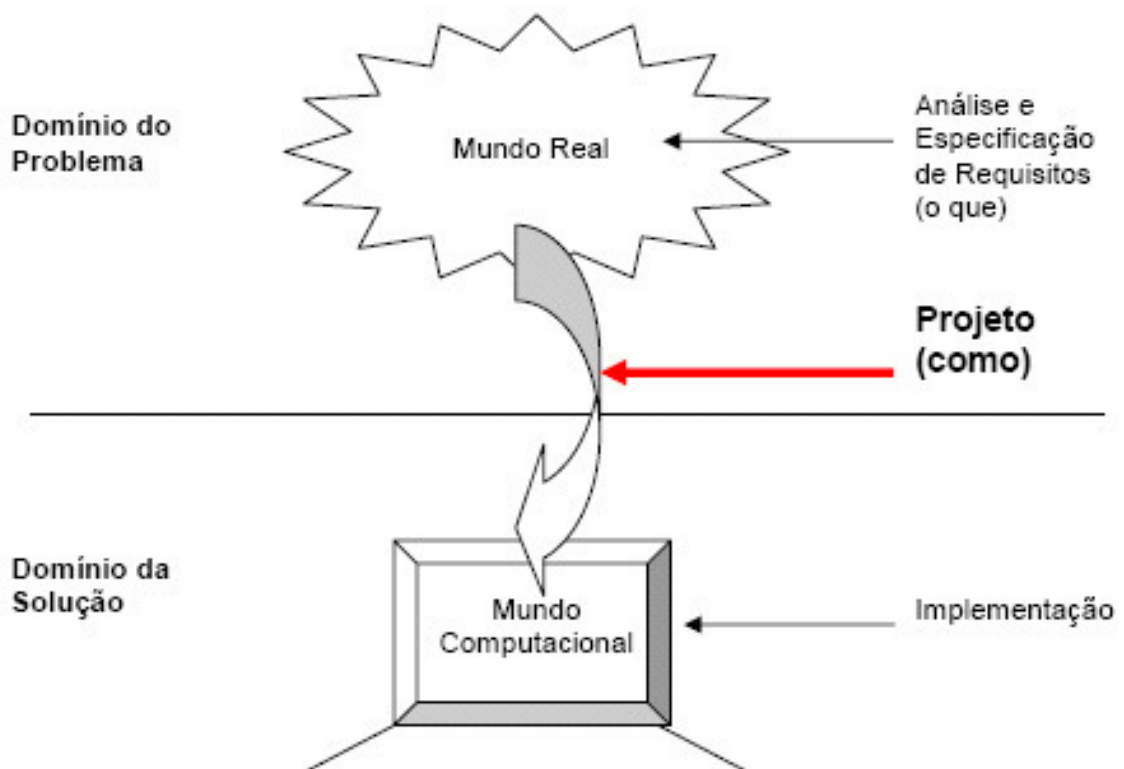


Figura 1.2 – Visão geral da atividade de Projeto

Fonte: Falbo, 2005

Para Falbo (2005), na fase de projeto devem ser observados os seguintes aspectos:

- Características da linguagem de programação a ser utilizada;
- Características do modelo de persistência de dados a ser adotado;

- Características da plataforma de desenvolvimento;
- Características da interface com o usuário.

Na fase de projeto deste sistema foram realizadas as seguintes atividades: definição da arquitetura do sistema; divisão em camadas; caracterização da interface com o usuário; e mecanismos de persistência de dados. Além disso, os modelos de análise foram adaptados para atender aos padrões de projetos. O capítulo 05 demonstra a especificação de projeto do sistema.

1.2.3.4 Implementação e Testes

Tendo em vista que a maior parte do esforço de desenvolvimento de um sistema ocorre nas fases de análise e projeto, a fase de implementação se restringe à codificação do que já foi definido. Nesse caso cabe ao programador dominar as características específicas das linguagens, ferramentas, frameworks e estruturas de dados para adaptar o código gerado aos requisitos indicados (WAZLAWICK, 2004).

Uma vez implementado o código de uma aplicação, o mesmo deve ser testado para descobrir tantos defeitos quanto possível e corrigi-los, antes da entrega do produto de software ao seu cliente (FALBO, 2005).

Para verificar se os componentes gerados atendem à especificação do projeto e aos casos de uso, são realizados testes à medida que o sistema vai sendo desenvolvido, com intuito de minimizar a possibilidade de erros oriundos da implementação. Posteriormente são realizados testes integrados para verificar se o sistema corresponde ao que foi anteriormente especificado, analisado e projetado.

A codificação do sistema e seus respectivos testes estão descritos no capítulo 06 deste trabalho.

1.3 Organização do Texto

Este trabalho apresenta o texto estruturado em sete capítulos e dois anexos. A seguir, segue uma breve descrição sobre o que é abordado em cada um.

Capítulo 1 – *Introdução* – parte inicial que introduz as idéias que serão abordadas no decorrer do trabalho, além da descrição da metodologia utilizada.

Capítulo 2 – *Estimativas de Projeto de Software* – apresenta o histórico e os conceitos referentes a estimativas de projetos de software, especialmente sobre a técnica de Análise de Pontos de Função.

Capítulo 3 – *Especificação de Requisitos* – contém a especificação dos requisitos não funcionais e funcionais, com diagramas de casos de uso e uma breve descrição de cada um.

Capítulo 4 – *Análise* – mostra a modelagem de análise do sistema, utilizando diagramas estáticos e dinâmicos, a fim de representar os conceitos característicos do sistema.

Capítulo 5 – *Projeto* – mostra a representação do sistema que será construído, através de sua arquitetura e a descrição das camadas.

Capítulo 6 – *Implementação e Teste* – descreve as atividades de teste e apresenta o resultado das etapas anteriores exibindo as telas do sistema.

Capítulo 7 – *Considerações Finais* – apresenta as considerações finais sobre o software, bem como sobre as experiências adquiridas e propostas de futuras melhorias.

Anexo A – Descrição dos *Casos de Uso* – possui a descrição detalhada de todos os casos de uso do sistema.

Anexo B – Formulário do ISBSG – apresenta o formulário elaborado pelo ISBSG para submeter projetos à base histórica do órgão.

Capítulo 2

Estimativas de Projetos de Software

A previsibilidade dos projetos constitui uma das grandes preocupações das equipes de desenvolvimento de software. Em muitas organizações, os projetos freqüentemente atrasam excessivamente, gastando mais recursos do que o orçamento planejado. Glass (2003) afirma que "(...) algumas das principais causas para isso são estimativas feitas baseadas em informação insuficiente, estimativas feitas para satisfazer a direção, e projetos com requisitos mal elaborados (...)". A geração de estimativas baseadas em memória pessoal também constitui uma causa comprovada para os projetos que extrapolam os seus orçamentos e cronogramas (HENRY, 2002).

Um ponto de partida para fazer estimativas de esforço, custo e prazo confiáveis é saber o tamanho do software. Segundo Macoratti (2007), existem vários motivos para medir o tamanho de um software, dentre eles:

- Fornecer subsídios para determinar o esforço, os recursos, a duração e os custos de desenvolvimento;
- Avaliar a produtividade do processo de desenvolvimento adotado;
- Formar uma base histórica para embasar estimativas futuras;
- Indicar a qualidade do produto;
- Controlar a evolução do projeto;
- Identificar divergências entre o que foi previsto e o que foi realizado;

2.1 Técnicas de estimativas de tamanho

Até o final da década de 70 era utilizada apenas a técnica de KLOC (quantidade de linhas de código) para estimativas de projetos de software. Segundo Sommerville (2003), essa técnica foi criada quando a maioria dos sistemas eram desenvolvidos em linguagens como FORTRAN, Assembly ou COBOL. O número de linhas de

código era facilmente calculado contando o número de cartões na caixa do programa. Hoje em linguagens como JAVA e C++, os sistemas não são feitos mais com cartões perfurados, e os códigos podem apresentar comentários e muitos comandos por linha, logo não é simples fazer uma relação entre a quantidade de linhas de código e as funcionalidades do sistema.

Com o crescimento das linguagens de programação e o aumento da complexidade dos softwares, foi ficando cada vez mais difícil gerenciar e fazer estimativas apenas com as técnicas por KLOC. Com isso, surgiu a necessidade, por parte dos desenvolvedores, em buscar novas técnicas de estimativa. Uma alternativa foi o método de Medição Funcional de Tamanho (*Function Size Measurement - FSM*), que consiste no termo geral para métodos que dimensionam o tamanho do software a partir das funções requisitadas pelo usuário.

2.1.1 Histórico da Análise de Pontos de Função

O método de Medição Funcional de Tamanho iniciou com a Análise de Pontos de Função (APF) na IBM. Os conceitos foram introduzidos por Allan J. Albrecht, em uma conferência do Grupo de Usuários IBM - Guide/Share, em 1979 (PRESSMAN, 2002). Allan Albrecht deu início a essa técnica, pois tinha sido encarregado de medir a produtividade de vários projetos de software desenvolvidos pela IBM. Mas esses projetos tinham sido desenvolvidos em linguagens diferentes, tornando inviável uma análise conjunta de produtividade com os métodos conhecidos na época. Essa foi a motivação para ele criar uma medida que fosse independente da linguagem de programação utilizada.

A técnica de Análise de Pontos de Função foi refinada em 1984, e a partir desta data houve um aumento considerável na sua utilização, e uma crescente aceitação pelo mercado, levando ao surgimento de diversas variações da proposta apresentada por Allan Albrecht. Surgiu assim, a necessidade de definir um padrão para aplicação da técnica. Com este objetivo foi criado em 1986 o *International Function Point Users Group* (IFPUG) que passou a ser responsável pela definição das regras de contagem, treinamento e certificação dos usuários da técnica. Em 1990, foi lançada a primeira versão do Manual de Práticas de Contagem ou CPM com o objetivo de

padronização da técnica. Atualmente esse manual está na versão 4.2. Embora tenham surgido outras técnicas de medição funcional, como Mark II e COSMIC-FFP, nenhuma delas possui ainda a aceitação pelo mercado quanto aos pontos de função do IFPUG (VAZQUEZ et al, 2003). A partir da figura 2.1 é possível acompanhar o crescimento da técnica.

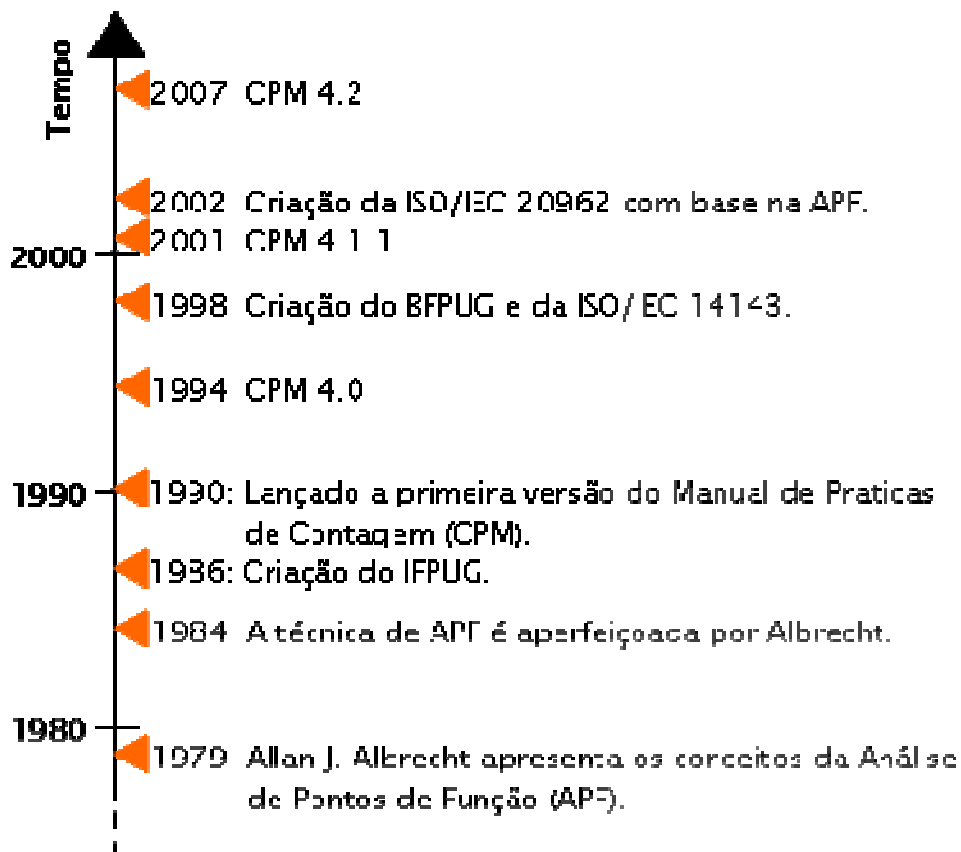


Figura 2.1 – Marcos históricos da Análise de Pontos de Função (APF)

No final de 1992, podiam ser encontrados diversos métodos de Medição Funcional de Tamanho, surgidos a partir de diferentes interpretações do método original da Análise de Pontos de Função proposto por Albrecht. Para tentar solucionar essas inconsistências existentes e estabelecer um método mais rigoroso de medição funcional, os grupos de métricas de software da Austrália, Reino Unido, Holanda e Estados Unidos formaram um grupo de trabalho denominado WG12 (*Working Group 12*). A partir dos resultados dos trabalhos do WG12, foi estabelecido um conjunto de padrões internacionais chamados de norma 14143 (ISO/IEC 14143), com o objetivo de garantir que todos os métodos de medição funcional sejam baseados em

conceitos semelhantes, e possam ser testados para assegurar o comportamento similar (VAZQUEZ et al, 2003).

A Análise de Pontos de Função foi submetida a um PAS - *Publicly Available Specification* a fim de ser considerada uma técnica de Medição Funcional de Tamanho no padrão ISO-14143, e ao final do ano de 2002, foi aprovada sob a denominação ISO/IEC 20926:2002. (VAZQUEZ et al, 2003).

2.1.2 Análise de Pontos de Função no Brasil

No Brasil a técnica de Análise de Pontos de Função (APF) começou a ser usada significativamente no começo da década de 90, com o apoio da empresa Unisys. No período de 1991 a 1994 foram realizados seis ENUPF (Encontro Nacional de Usuários de Pontos de Função), que contava com a participação de palestrantes internacionais.

Porém, o interesse no mercado nacional só começou a ganhar força quando empresas de desenvolvimento de software passaram a se preocupar com modelos de qualidade, como ISO ou CMM, e também quando grandes contratos públicos eram fechados baseados em pontos de função. A criação do BFPUG foi também um importante marco para essa consolidação da técnica no Brasil (VAZQUEZ et al, 2003).

O BFPUG é a representação brasileira oficial (*Chapter*) do IFPUG - *International Function Point Users Group*. Possui o objetivo de estimular e divulgar a utilização de métricas no desenvolvimento de sistemas, em particular a Análise de Pontos de Função - Function Point Analysis, ou FPA. Destina-se aos profissionais interessados em aprender, praticar e divulgar o uso de métricas e de FPA (BFPUG, 2008).

Uma pesquisa realizada pelo Ministério da Ciência e Tecnologia - MCT, em 2006, indicou que a utilização da APF vem sendo utilizada no Brasil, principalmente nas grandes empresas, conforme mostra a tabela 2.1:

Tabela 2.1 - Métricas primitivas utilizadas para medir a produtividade dos processos de software em porcentagem (múltipla escolha)

Categorias	Total	Micro	Pequena	Média	Grande
Linhas de código (<i>LOC</i>)	10,3	7,5	10,3	11,9	14,6
Pontos por função (<i>Function Point</i>)	18,2	10,6	21,4	14,3	28,1
Outras métricas	6,7	3,7	5,5	11,9	11,5
Não utiliza	70,0	81,4	66,9	69,0	55,2

Fonte: MCT, 2006.

Muitas empresas têm contratado, para realizar seus projetos, empresas de softwares que realizam estimativas de Pontos de Função. A FATTO Consultoria e Sistemas é um exemplo de empresa que presta tais serviços, oferecendo consultoria, treinamentos e aplicando a técnica de APF no contexto específico da organização.

2.1.3 Outras Técnicas

Existem outras técnicas de medição do software, destacando entre elas a UCP (*Use Case Points* ou Pontos por Casos de Uso). Essa técnica surgiu como uma adaptação específica da APF e seu criador, Gustav Karner, propôs tal técnica com o intuito de estimar recursos de projetos de software orientados a objeto desenvolvidos utilizando o processo Objectory (FATTO, 2008).

A UCP vem crescendo e ganhando cada vez mais adeptos, pois é uma técnica simples de ser utilizada se a organização possui um padrão bem definido de especificação de casos de uso e traz resultados interessantes dentro do contexto da organização. Mas, deve-se levar em consideração algumas desvantagens, destacando entre elas:

- Somente poder ser aplicado em projetos de software cuja especificação tenha sido expressa por casos de uso;

- Não existe um padrão único para escrita dos casos de uso, podendo levar a diferentes valores na medição;
- Não contempla a medição de projetos de melhoria;
- Falta da existência de um órgão regulador da técnica, tal como o IFPUG;

Segundo a FATTO Consultoria e Sistemas (2008), dentre as desvantagens da UCP em relação à APF, algumas poderiam ser superadas com alguns ajustes simples. No entanto não há benefício adicional do UCP sobre a APF.

2.1.4 Vantagens e Desvantagens da Análise de Pontos de Função

A técnica de Análise de Pontos de Função conta com vários benefícios e vantagens para utilização, tais como:

- Medir a funcionalidade solicitada pelo usuário, antes da fase de projeto de software, de forma a estimar seu tamanho e custo;
- Possibilita medir projetos de desenvolvimento e manutenção de software, independentemente da tecnologia utilizada na implementação;
- É um fator de normalização para comparação de software;
- Diversas organizações, como o ISBSG, disponibilizam um repositório de dados de projetos de software que permitem a realização de benchmarking com projetos similares do mercado;
- Servir de base para indicadores que apóiam na qualidade, como por exemplo, produtividade da equipe e defeitos por tempo;
- Pode ser usada para gerar dados históricos;
- Permite acompanhar a evolução do escopo do projeto;
- É uma ferramenta para fundamentar a negociação de contratos;
- Oferece treinamento e certificação;
- Possui regras padronizadas e mantidas por uma Organização Internacional;
- Pode estimar o tamanho em qualquer fase do ciclo de vida do desenvolvimento de sistema;

Porém, devem-se citar também algumas desvantagens, como:

- Necessidade de um bom nível de experiência na técnica para efetuar uma contagem apurada;
- Necessidade significativa de um maior nível de detalhe de informações do software para uma medição confiável;
- A medição da produtividade é prejudicada caso o processo de desenvolvimento da organização seja caótico (cada projeto é desenvolvido de forma diferente), mesmo que a contagem dos pontos de função do projeto e o registro do esforço tenham sido feitos de forma correta;

2.2 Processo da Contagem de Pontos de Função

Pode-se dividir o processo de contagem nas seguintes etapas:

1. Definir o tipo de contagem.
2. Identificar o escopo de contagem e a fronteira da aplicação.
3. Identificar as funções do sistema.
4. Classificar cada função quanto à complexidade funcional.
5. Calcular os pontos de função não ajustados através da aplicação dos pesos de acordo com a tabela específica.
6. Determinar pontos de função ajustados.

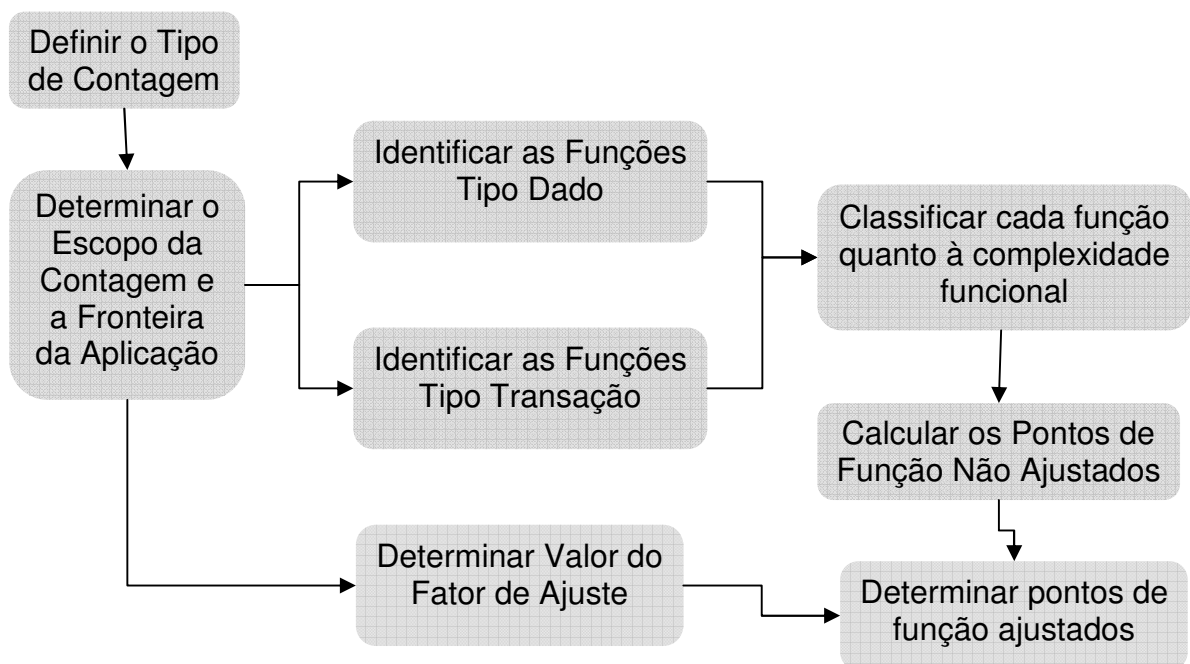


Figura 2.2 – Processo da Contagem de Pontos de Função

2.2.1 Definir o Tipo de Contagem

A análise de pontos de função pode ser aplicada tanto para aplicações prontas quanto para projetos em desenvolvimento ou de melhoria. A contagem pode ser feita através de um dos três tipos seguintes.

2.2.1.1 Projeto de Desenvolvimento

Uma contagem feita antes do término da aplicação, sendo assim uma estimativa para o projeto final, já que é comum adicionar ou modificar funcionalidades no decorrer do desenvolvimento. Esta contagem pode ser feita em qualquer etapa do ciclo de desenvolvimento, auxiliando na monitoração do *scope creep* (variação de tamanho do escopo).

2.2.1.2 Projeto de Melhoria

É comum que sejam requisitadas novas funcionalidades a um sistema depois de ter sido colocado em produção. Para avaliar essas mudanças, este tipo de contagem mede as funções adicionadas, modificadas ou excluídas de um determinado sistema pelo projeto.

2.2.1.3 Aplicação

Este tipo de contagem também é chamado de pontos de função instalados ou *baseline*. Tem como objetivo, medir a atual funcionalidade fornecida aos usuários de sistemas em produção.

2.2.2 Identificar o Escopo da Contagem e a Fronteira de Aplicação

Esta etapa tem como objetivo definir os limites do escopo para a contagem. Inclui saber quais sistemas serão abordados e se a contagem será feita de todas as funcionalidades ou apenas parte delas, por exemplo, apenas os relatórios. A fronteira da aplicação vai indicar o limite entre a aplicação a ser medida e o usuário

ou outro sistema. Esta fronteira é a porta pelo qual dados processados por transações devem passar para chegar ao meio externo da aplicação.

2.2.3 Identificar as Funções do sistema

Um Ponto de Função é a unidade de medida da APF que mede as funções de armazenamento e processamento disponíveis ao usuário. Essas funções são classificadas em cinco tipos diferentes, divididas em dois grupos: funções do tipo dados e funções do tipo transações. A seguir segue uma breve descrição, segundo Vazquez et al (2003) e Dias (2003).

2.2.3.1 Funções do Tipo dados

As funções do tipo dados são funções que representam as funcionalidades fornecidas pelo sistema ao usuário, para atender as suas necessidades de dados. São classificadas em Arquivo Lógico Interno e Arquivo de Interface Externa.

Arquivo Lógico Interno (ALI): Cada ALI é um grupo conceitual de dados, inter-relacionados, requisitado e identificável pelo usuário como necessidades de informação e mantidos dentro da fronteira da aplicação. Sua principal intenção é armazenar dados do sistema, através de um ou mais processos da aplicação sendo contada. Exemplo: tabelas de banco de dados atualizadas pela aplicação.

Arquivo de Interface Externa (AIE): Cada AIE é um grupo conceitual de dados, inter-relacionados, identificável pelo usuário como necessidades de informação e mantidos fora da fronteira da aplicação. Estes arquivos são ALI de outra aplicação, ou seja, não são mantidos pela aplicação que está sendo contada. Exemplo: tabelas de banco de dados atualizadas por outra aplicação.

2.2.3.2 Funções do Tipo Transação

As funções do tipo transação são funções que representam as funcionalidades de processamentos de dados fornecidas pelo sistema ao usuário. São classificadas em Entrada Externa, Saída Externa e Consulta Externa.

Entrada Externa (EE): transações com o objetivo de manter dados ou alterar o comportamento do sistema. Exemplo: incluir cliente, alterar cliente, excluir cliente.

Saída Externa (SE): transações com o objetivo de apresentar dados ao usuário envolvendo cálculo ou geração de dados derivados ou atualização de ALIs ou alteração do comportamento do sistema. Exemplo: relatório de totais de faturamento por cliente.

Consulta Externa (CE): transações com o objetivo de apresentar dados ao usuário envolvendo uma simples recuperação de dados de arquivos lógicos. Exemplo: consulta cadastro clientes.

2.2.3.3 Determinação da Complexidade das Funções

As funções podem ser classificadas de acordo com sua complexidade em três níveis: baixa, média ou alta. Para determinar a complexidade de uma função, o primeiro passo é contar o número de Tipos de Dados (TD) e o número de Tipos de Registros (TR) para funções do tipo dados ou número de Arquivos Referenciados (AR) para funções do tipo transação. O segundo é fazer a classificação de acordo com a tabela correspondente com o seu tipo. Estas tabelas são encontradas abaixo.

A definição para os termos TD, TR e AR segundo Vazquez et al (2003) é:

Tipo de Dado (TD): campo não repetido reconhecido pelo usuário.

Tipo de Registro (TR): subgrupo de ALI ou AIE, reconhecido pelo usuário.

Arquivo Referenciado (AR): Arquivo Lógico Interno lido ou mantido pela função do tipo transação, ou um Arquivo de Interface Externa lido pela função.

Tabela 2.2 - Complexidade funcional dos ALI e AIE

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

Fonte: Vazquez et al (2003)

Tabela 2.3 - Complexidade funcional para EE

Arquivos Referenciado	Tipos de Dados		
	< 5	5 - 15	> 15
< 2	Baixa	Baixa	Média
2	Baixa	Média	Alta
> 2	Média	Alta	Alta

Fonte: Vazquez et al (2003)

Tabela 2.4 - Complexidade funcional para SE e CE

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

Fonte: Vazquez et al (2003)

2.2.4 Pontos de Função não ajustados

Após a identificação e classificação de todas as funções na contagem, o número de pontos de função não ajustados será simplesmente a soma do valor de cada função de acordo com os dados da tabela abaixo. Por exemplo, se na contagem fosse identificado dois SE de complexidade média e um de baixa, então a quantidade de pontos de função não ajustados seria $2 \times 5PF + 4 PF = 14PF$.

Tabela 2.5 - Contribuição das funções para Pontos de Função Não Ajustados

Tipo de função	Baixa	Média	Alta
Arquivo Lógico Interno (ALI)	7 PF	10 PF	15 PF
Arquivo de Interface Externa (AIE)	5 PF	7 PF	10 PF
Entrada Externa (EE)	3 PF	4 PF	6 PF
Saída Externa (SE)	4 PF	5 PF	7 PF
Consulta Externa (CE)	3 PF	4 PF	6 PF

Fonte: Vazquez et al (2003)

2.2.5 Determinar os pontos de função ajustados

Para calcular a contagem de pontos de função ajustada é preciso determinar o valor do fator de ajuste (VFA). Esse fator de ajuste é calculado através da identificação dos níveis de influência de 14 Características Gerais do Sistema (CGS), no qual refletem funções que afetam a aplicação de maneira geral.

Porém, o uso do fator de ajuste tornou-se opcional ao final do ano de 2002, com a adequação ao padrão ISO/IEC de medição funcional no IFPUG, pois várias das CGS contemplam requisitos não-funcionais.

Além disso, antes mesmo do uso do fator de ajuste tornar-se opcional, uma pesquisa pelo IFPUG demonstrou que vários usuários já não o utilizavam, devido a algumas das CGS estarem desatualizadas. Como exemplo, podemos citar um dos modelos de estimativa para softwares mais comuns, o COCOMO II, utiliza apenas os pontos de função não ajustados como entrada do processo (VAZQUEZ et al, 2003).

A contagem de pontos de função ajustada nada mais é do que a contagem de pontos de função não ajustadas multiplicada pelo valor do fator de ajuste, envolvendo o cálculo final para os três tipos de contagem: projeto de desenvolvimento, projeto de melhoria, e aplicação. Mas, como o cálculo do fator de ajuste passou a ser opcional, a contagem ajustada passou a ser cada vez menos utilizada.

2.3 Diferentes Tipos de Abordagem

Todo o processo explicado no item anterior (2.2 Processo da Contagem de Pontos de Função) descreve as etapas para a realização da contagem detalhada de Pontos de Função, que segue as normas definidas pelo IFPUG. Mas a sua utilização requer um alto nível de conhecimento dos requisitos do sistema, o que muitas vezes não acontece quando se está ainda na etapa de planejamento. A fim de poder utilizar a técnica nos estágios iniciais do desenvolvimento, são utilizadas outras duas abordagens: Indicativa e Estimativa.

Essas duas abordagens tiveram suas definições apresentadas pela NESMA (*Netherlands Software Metrics User Association* - Associação de Métricas da Holanda) que possui objetivos semelhantes aos do IFPUG, e possui seu próprio manual de contagem. Ambas as organizações possuem a mesma filosofia, conceitos e regras, mas entre as diferenças, abordam de maneira distinta os projetos de melhoria (VAZQUEZ et al, 2008).

2.3.1 Contagem Indicativa

Para realizar uma contagem indicativa, é necessário saber apenas os Arquivos Lógicos Internos (ALI) e os Arquivos Lógicos Externos (AIE). Para cada ALI é adicionado 35 PF no tamanho do software e 15 PF para cada AIE. Esses valores foram obtidos a partir de uma análise entre vários projetos e chegou-se a conclusão que, em média, para cada ALI do sistema existem uma série de transações relacionadas (EE, CE e SE) que juntos somariam 35 PF, já para o AIE seriam apenas 15 PF.

2.3.2 Contagem Estimativa

Nesta abordagem é necessária a identificação de todas as funções (ALI, AIE, EE, CE e SE), mas ainda não é preciso saber a quantidade de TD e AR/TR, como é feito na Contagem Detalhada. Isso ocorre pois a quantidade de PF de cada função é dada por uma média de valores fornecidos pelo ISBSG. Assim o valor estimado do tamanho é dito pela fórmula que corresponde na soma da multiplicação dos tipos de função e seus respectivos valores médios:

$$\text{Tamanho} = (\text{EE} \times 4) + (\text{SE} \times 5) + (\text{CE} \times 4) + (\text{ALI} \times 7) + (\text{AIE} \times 5)$$

2.4 Base Histórica dos Projetos

Para se obter uma boa utilização da Análise de Pontos de Função faz-se necessário a realização de uma base histórica das contagens de pontos de função realizadas pela equipe, pois através dessa base é possível obter estimativas mais precisas, além de poder calcular indicadores importantes para auxílio na tomada de decisão.

Segundo Galvão (1999), são objetivos do histórico dos projetos de análise de pontos de função:

- Monitorar os indicadores de produtividade baseados em Pontos de função;
- Viabilizar avaliação de índices de qualidade nos projetos em função de seu tamanho medido em pontos de função;
- Transparência dos custos dos projetos referentes ao total de pontos de função;

- Comparação entre projetos a partir de índices de similaridade

A organização sem fins lucrativos, ISBSG (International Software Benchmarking Standards Group), mantém um repositório de métricas de projeto de software a nível mundial. Atualmente possui dados de mais de 4 mil projetos de vários países. São diversas organizações que alimentam o repositório, submetendo seus dados através de um questionário padrão, disponibilizado no site da organização. Com base nas informações coletadas o ISBSG desempenha um papel importante, servindo de local para consulta de projetos similares e disponibilizando dados estatísticos periodicamente.

A utilização da base de dados do ISBSG pode ser uma tarefa difícil, pois as informações obtidas podem fugir um pouco da realidade da organização. Assim, as empresas buscam criar os seus próprios repositórios, o que garante um auxílio na precisão das estimativas. Em geral os dados históricos dos projetos devem conter qualquer informação que possa ajudar numa nova estimativa, por exemplo: plataforma, processo de desenvolvimento, tipo de linguagem, pontos de função não ajustados ou ajustados, os valores estimados e reais para produtividade, custo e esforço.

2.5 Indicadores com Base em Pontos de Função

Os pontos de função passam a ter maior significado quando utilizados como parâmetros na obtenção de estimativas de outras variáveis relevantes para a gerência de projetos, tais como produtividade, esforço, duração, custo e qualidade (VAZQUEZ et al, 2003).

2.5.1 Produtividade e Taxa de Entrega

A produtividade da equipe de desenvolvimento serve como base para o cálculo de diversos indicadores e é calculada encontrando-se a razão entre o Número de pontos de função entregues (F) e a quantidade de unidades de tempo utilizadas para produzi-los (E). Assim, obtém-se a fórmula:

$$P = \frac{F}{E}$$

Pode-se definir a produtividade como, por exemplo, a razão entre o tamanho do projeto em PF e a quantidade total de horas (H) ou de Homem x Mês (HM) gastos no seu desenvolvimento, na unidade PF / H ou PF / HM respectivamente (VAZQUEZ et al, 2003).

Através da produtividade é possível estimar o esforço ou o custo de forma menos empírica. Mas, é importante lembrar que o valor desta informação para fins de estimativa está relacionado com o conhecimento histórico e só deve ser aplicado em novas situações cujas condições sejam semelhantes ao de algum projeto existente no histórico. Na base histórica, os projetos devem estar separados por classes e essas classes devem procurar compreender projetos que tenham características semelhantes quanto a fatores que afetam a relação entre tamanho e esforço. Dessa forma, é possível obter uma produtividade para cada classe de projetos diferente.

Muitas vezes outro indicador é utilizado indistintamente como um medidor de produtividade. Tal indicador é denominado pelo mercado como taxa de entrega e medido numa razão inversa à da produtividade (VAZQUEZ et al, 2003).

A taxa de entrega, geralmente, é medida em H / PF . Assim, uma taxa de entrega de $4 H / PF$ corresponde a uma produtividade de $0,25 PF / H$, da mesma forma que uma produtividade de $0,5 PF / H$ equivale a uma taxa de entrega de $2 H / PF$. Essa medida é mais utilizada para efeito de estimativa de esforço, pois é possível obter uma relação direta entre o aumento da taxa de entrega e o aumento do esforço.

Assim, a produtividade conforme definida é mais utilizada para efeito de monitoramento do desenvolvimento de software, a partir do conhecimento do tamanho e esforço realizados em cada fase de seu ciclo de vida, sendo que um aumento desse indicador teria uma relação direta com o aumento da maturidade e qualidade do processo (VAZQUEZ et al, 2003).

Deve-se levar em conta que essas relações lineares merecem atenção ao serem aplicadas em projetos pequenos (menores que 100 PF), pois as particularidades de

cada projeto terão um efeito maior na estimativa final, podendo levar a uma falta de precisão.

2.5.2 Esforço

A melhor forma de chegar à estimativa de esforço a partir da estimativa de tamanho do projeto é utilizar dados internos da própria organização, tais como taxa de entrega ou produtividade. Dessa forma, obtém-se o esforço através da seguinte fórmula:

$$E = F \times T \quad \text{ou} \quad E = \frac{F}{P}$$

Porém, para que o cálculo estimado do esforço seja confiável, é preciso que a taxa de entrega ou a produtividade sejam calculadas para cada classe de projeto existente na base histórica. Dessa forma, deverá ser utilizada a taxa de entrega ou a produtividade da classe em que se encaixa o novo projeto.

2.5.3 Qualidade

Outro ponto importante a ser tratado é a utilização da Análise de Pontos de Função como auxílio na análise da qualidade do software. O cálculo da quantidade de defeitos por pontos de função é utilizado para fazer tal análise e pode servir como base para diversos fatores. A fórmula a seguir demonstra como calcular tal indicador:

$$\text{Qualidade} = \frac{\text{Defeitos}}{\text{PF}}$$

Algumas empresas costumam, inclusive, exigir um valor limite para os defeitos por pontos de função para contratação do software, garantindo a sua qualidade. Vale ressaltar ainda que quanto menor o indicador de qualidade, pois significa que tem poucos defeitos por pontos de função.

2.5.4 Duração

Dentre o planejamento do desenvolvimento de um software, a estimativa de sua duração é de grande importância para o gerente de projetos e, principalmente, o cliente. Ao contratar um serviço, o cliente quer saber basicamente quanto tempo vai

demorar em ser entregue e quanto vai custar. Determinar o cronograma é fundamental também para o gerenciamento do projeto.

A estimativa de duração é principalmente avaliada como a relação entre a estimativa de trabalho envolvido e os recursos disponíveis para a sua execução. Uma vez determinado o esforço em horas, necessário para a realização de uma determinada atividade, para obter sua estimativa de duração, basta dividir esse valor pelo número de horas trabalhadas pela equipe alocada, tendo assim a seguinte fórmula:

$$\text{Prazo} = \frac{\text{esforço}}{\text{recursos}}$$

Ou seja, o prazo previsto será a razão entre o esforço previsto e a quantidade de horas dos recursos alocada na execução do projeto. Essa relação pode parecer lógica, mas a experiência comprova que a relação linear entre prazo e quantidade de recursos não é verdadeira. Ao aumentar o número de recursos é necessário um maior esforço para gerenciá-los. Esse trabalho é consequência de novas atividades para comportar a comunicação, integração e seqüenciamento de atividades geradas pela maior distribuição das tarefas (VAZQUEZ et al, 2003).

Existem outras técnicas que permitem realizar uma estimativa de prazo com mais precisão, por exemplo, o COCOMO (CONstructive COst MOdel) que é um modelo de estimativa que utiliza equações matemáticas para fazer estimativas de esforço, prazo e tamanho da equipe em projetos de software. Suas equações são baseadas em pesquisa e dados históricos e utilizam como entrada a quantidade de linhas de código (ou pontos de função) e a avaliação de outros aspectos relevantes para a estimativa. (FATTO, 2008).

2.5.5 Custo

Outra estimativa de grande importância é a de custo. Nas fases iniciais de um projeto é difícil saber quanto vai custar determinado software e para isso faz-se necessário estimar. Tal estimativa pode ser feita através da multiplicação da estimativa de esforço calculada pelo custo médio da hora da equipe, como na fórmula:

$$\text{Custo} = E \times CM$$

Para calcular o custo de um projeto deve-se utilizar o valor do custo por ponto de função. Algumas organizações utilizam, por exemplo, um valor para projetos de desenvolvimento e outro menor para projetos de melhoria. O custo final será dado pela multiplicação do custo por ponto de função e o tamanho em PF do projeto, conforme a seguinte fórmula:

$$\text{Custo} = (\text{Custo/PF}) \times \text{PF}$$

2.6 Ferramentas Similares

Foram analisadas diferentes ferramentas para Análise de Pontos de Função existentes no mercado, procurando encontrar as funcionalidades propostas nesse projeto. A tabela 2.6 apresenta uma análise comparativa dos softwares encontrados e suas funcionalidades.

Tabela 2.6 - Análise comparativa das ferramentas de Análise de Pontos de Função existentes no mercado

Softwares / Funcionalidades		APF Plus	FPRRecorder	SCOPE	FunctionPoint Modeler	WinFPA	
Plataforma		Desktop/Windows	Desktop/Windows	Desktop/Windows	Desktop/ Multi-plataforma	Desktop/Windows	
Usabilidade		Baixa	Média	Alta	Média	Alta	
Funcionalidades	Indicadores	<i>Produtividade</i>					
		<i>Tempo</i>			X	X	
		<i>Custo</i>	X		X		X
		<i>Qualidade</i>					X
	Gera Histórico			X	X	X	X
	Utiliza dados Históricos para novos Cálculos						
	Gera Relatório		X	X	X	X	X

A seguir, encontra-se uma breve descrição das principais ferramentas encontradas.

2.6.1 Scope - Project Sizing Software

Criado pela empresa australiana Total Metrics, especialista em métricas de softwares, este software é *shareware* e utiliza a técnica de Análise de Pontos de

Função para fazer o controle do escopo dos projetos de software. A figura 2.2 mostra a tela principal do sistema.

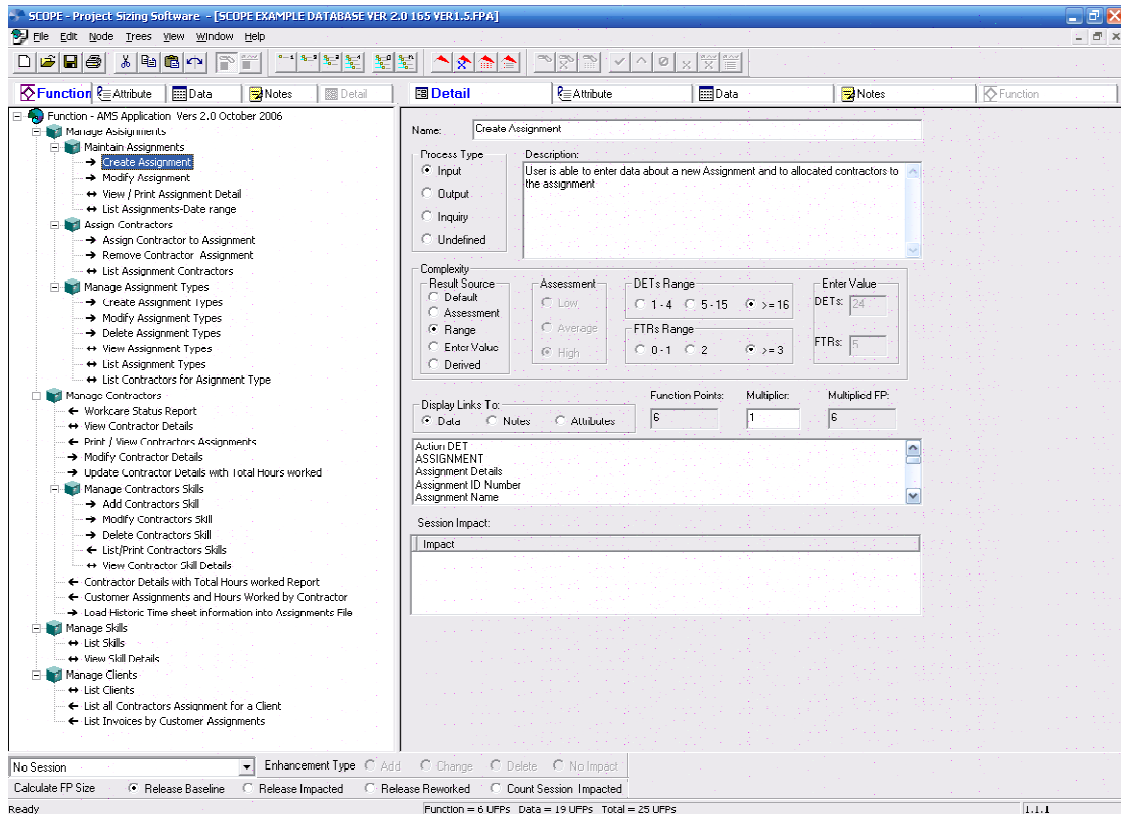


Figura 2.3 – Tela do software Scope

Dentre todas as ferramentas avaliadas, essa foi uma das que apresentaram a melhor estrutura para auxiliar no uso da técnica, já que possui uma boa usabilidade e diversas opções que atendem a todo o processo da contagem. Podemos destacar como pontos fortes da ferramenta:

- Boa navegabilidade;
- Excelente suporte do produto;
- Permite descrever com bastante detalhe o projeto e todo o processo de contagem.

Apesar de toda a sua robusta estrutura para a realização da contagem, ela não aproveita os dados históricos para aprimorar os novos cálculos. Outro recurso interessante que não está presente na ferramenta é o indicador de qualidade do software.

2.6.2 FPRecorder

Ferramenta *shareware* que apóia a Análise de Pontos de Função, fornecendo ao usuário a possibilidade de decompor o sistema em módulos, associar tipos de transação com tipos de dado e visualizar relatórios gráficos. A figura 2.3 apresenta a principal tela do software.

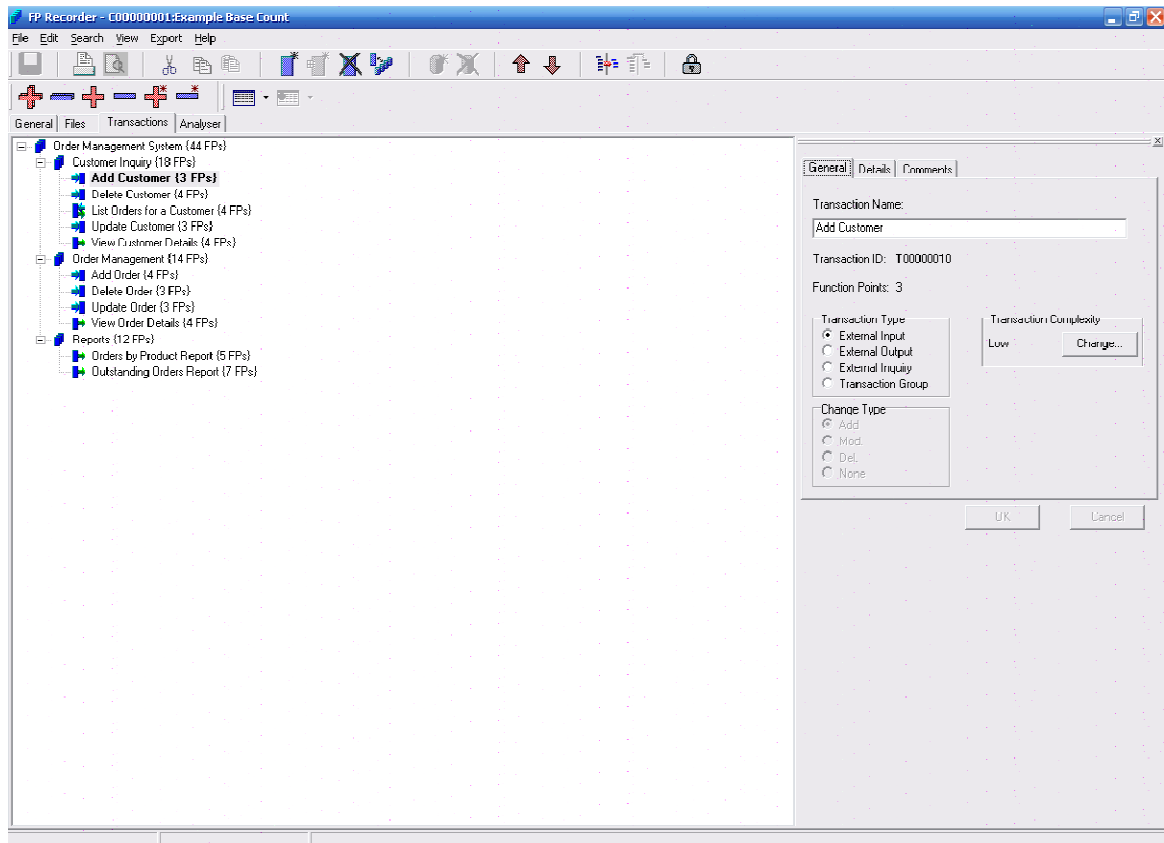


Figura 2.4 – Tela do software FPRecorder

Dentre os pontos fortes podemos destacar:

- Interface simples e funcional.

Porém, é importante ressaltar que não foi encontrada nenhuma forma de suporte da ferramenta além do help ou alguma perspectiva de sua continuidade, tendo em vista que a empresa criadora, a Chispl, não foi encontrada na pesquisa. Outro ponto negativo é que a ferramenta não possui outra estimativa de projeto de software além da de tamanho.

2.6.3 APFPlus

Programa desenvolvido pelo brasileiro Ivan Mecenas, que permite realizar estimativas de software utilizando a Análise de Pontos de Função e COCOMO II. A ferramenta tem como principais funções a contagem dos pontos de função e a estimativa de custo dos projetos, mas possibilita também a geração de relatórios, utilização em rede, envio de e-mails e controle de documentos de determinada pasta do programa.

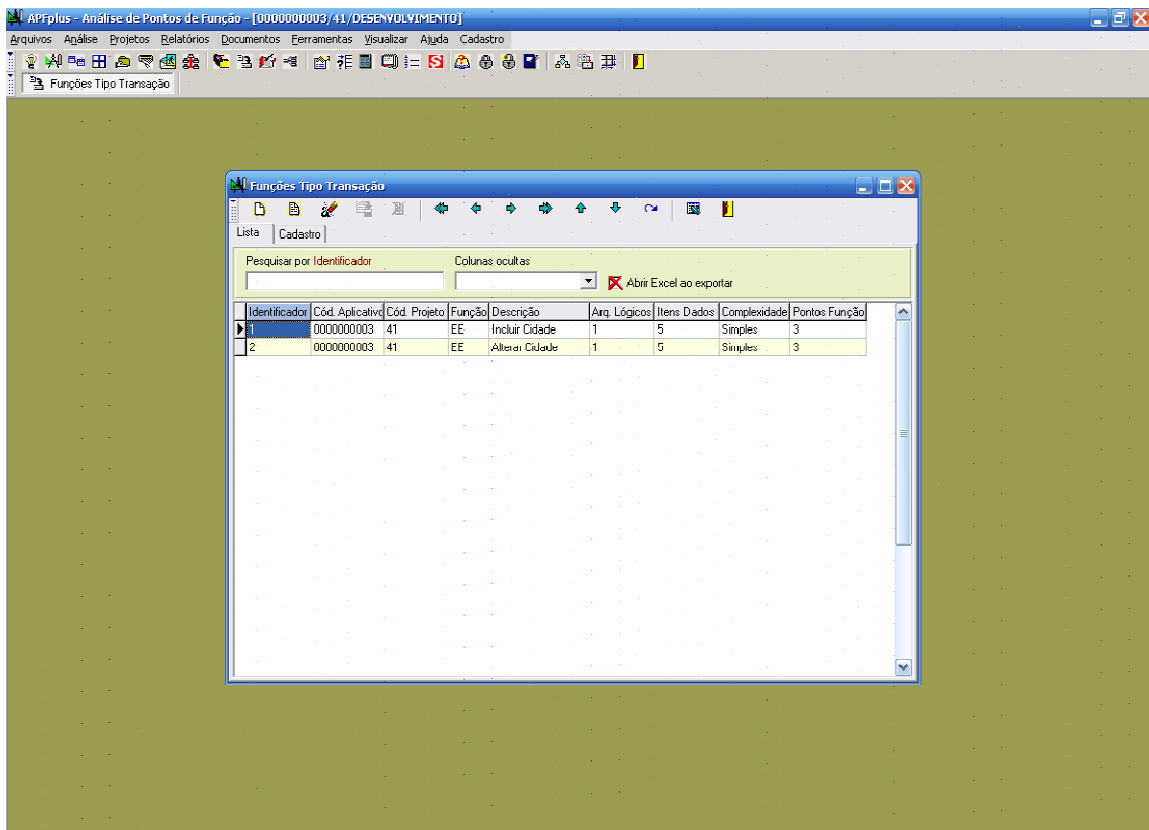


Figura 2.5 – Tela do software APFPlus

Dentre os pontos fortes podemos destacar:

- Utiliza o COCOMO II para estimar o custo.

Mas é possível destacar como ponto negativo da ferramenta sua baixa usabilidade e a falta de outras estimativas como tempo, produtividade ou ainda indicadores de qualidade.

2.6.4 WinFPA

O WinFPA é um software *shareware* especializado em Dimensionamento e Planejamento de Projetos de Desenvolvimento e Manutenção de Sistemas, que utiliza a Análise de Pontos de Função para dimensionar o tamanho dos projetos e para auxiliar no cálculo de diversos indicadores. O WinFPA possui compatibilidade com os sistemas Microsoft Windows. A figura 2.5 apresenta a tela principal do sistema.

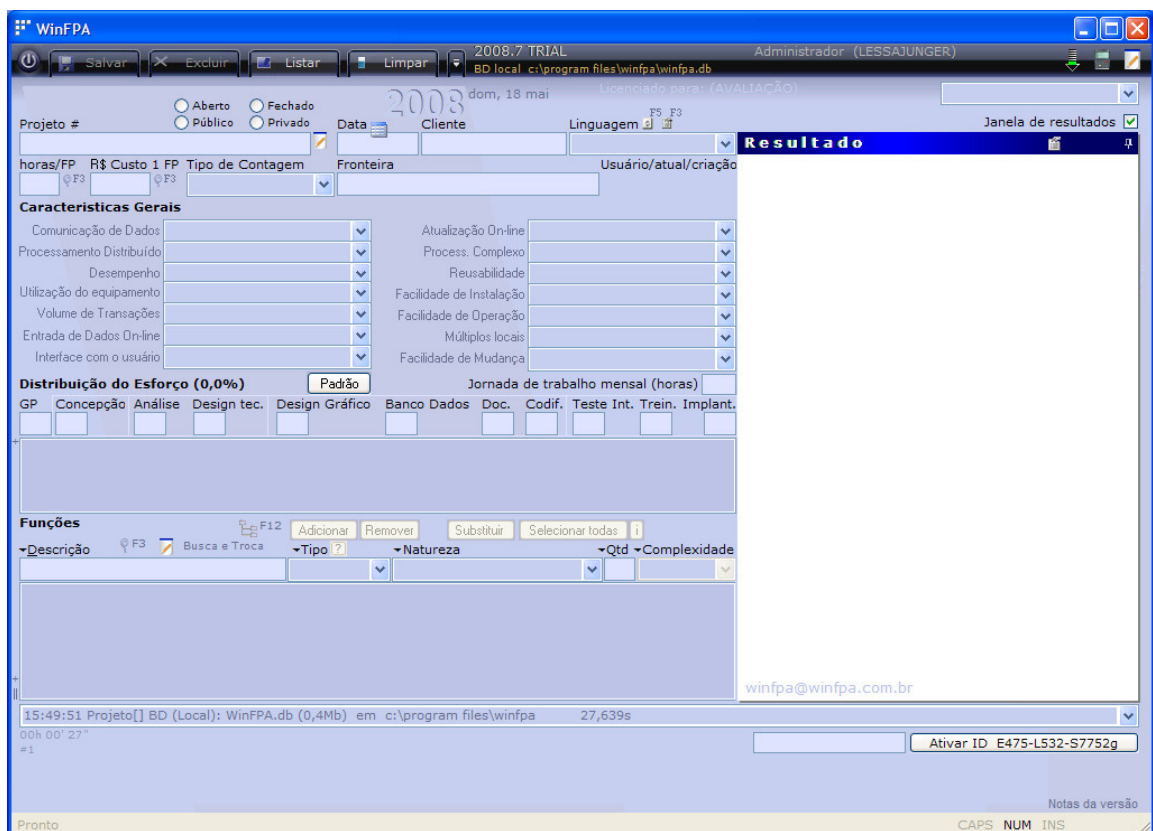


Figura 2.6 – Tela do software WinFPA

Podemos destacar como funcionalidades deste software:

- Geração automática de Cronogramas
- Simulações: Equipe e Duração
- Comparativos entre projetos
- Distribuição do Esforço do Projeto
- Estatística de Taxas de Entrega, Custos, Defeitos e Casos de Testes

O software possui ainda a opção de inserir o indicador de produtividade da empresa, fator esse de grande importância para o cálculo dos demais indicadores, tais como

esforço, duração, custo e qualidade. Mas, esse fator de produtividade não é calculado automaticamente através da utilização da base histórica de projetos da equipe, sendo esse um ponto negativo deste software.

Além disso, podemos citar como outro ponto negativo desta ferramenta o fato dela não ser multi-plataforma, podendo ser utilizada apenas em sistemas operacionais Windows.

2.6.5 Function Point Modeler

O Function Point Modeler é uma nova geração de ferramenta para medir a Análise de Pontos de Função de softwares. Foi desenvolvida por uma empresa alemã e é o primeiro produto que combina o modelo de Pontos de Função com outros modelos tais como *UseCase Model*, *Business Object Class Model* and *Data Model*. Além disso, baseia-se na mais popular plataforma de código aberto Eclipse e pode ser executado tanto como uma ferramenta *stand alone* quanto como um plugin no Eclipse para interagir com outras ferramentas. O Function Point Modeler suporta ainda a sintaxe UML e apóia o ciclo de vida de processos elementares dos sistemas. A Figura 2.6 mostra a tela principal do sistema.

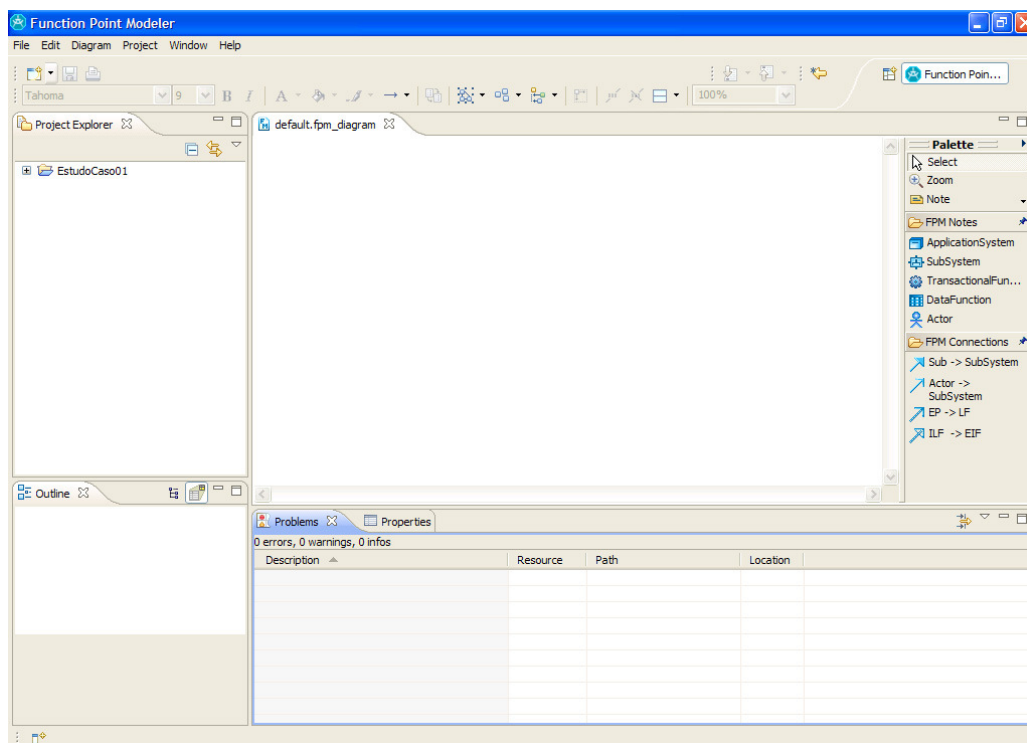


Figura 2.7 – Tela do software Function Point Modeler

Esta ferramenta se diferencia das demais em diversos aspectos, podendo citar como pontos positivos:

- Utilização de uma modelagem para Análise de Pontos de Função;
- Cálculo dos pontos de função através da construção da modelagem.

Porém, é preciso citar a dificuldade de utilização da ferramenta, devido ao fato da aplicação da modelagem ser desconhecida. Para que tal ferramenta seja de grande utilidade é necessário um treinamento da técnica de modelagem.

Além disso, a ferramenta não possui o cálculo de outras estimativas de projetos de software, como produtividade, esforço, duração, custo e etc., restringindo-se ao cálculo de pontos de função.

2.6.6 Outras ferramentas de estimativa

É preciso citar ainda outras ferramentas de estimativa de projetos de software encontradas no mercado, tais como: CustPlan, EstimaODE, Function Point Workbench, Costar, USC-COCOMO, CALICO, Cost Xpert 2.1.

Dentre estas ferramentas podemos destacar o EstimaODE. Tal ferramenta integra-se ao Ambiente de Desenvolvimento ODE, desenvolvido pelo Laboratório de Engenharia de Software (LabES) da Universidade Federal do Espírito Santo (UFES), e possui funcionalidades de grande utilidade, como a utilização de dados históricos para calcular a produtividade da equipe e estimar o esforço de projetos de desenvolvimento de software. Essa funcionalidade torna as estimativas mais precisas, fato esse de grande importância e encontrado apenas no EstimaODE. Além disso, é possível obter a estimativa de tamanho através de duas abordagens: Análise de Pontos de Função e Análise de Pontos de Caso de Uso. Segue abaixo a principal tela do sistema.

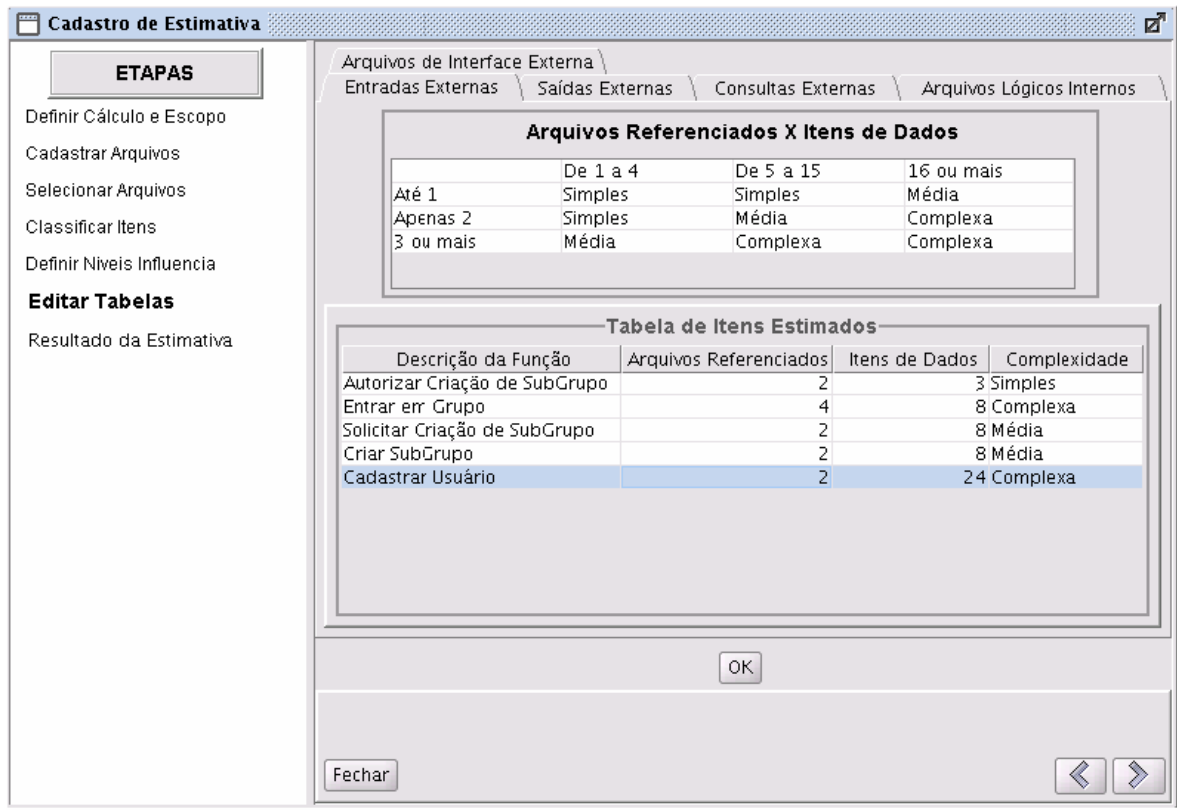


Figura 2.8 – Tela do software EstimaODE

Fonte: LabES, 2005

Porém, o estimaODE somente é utilizado junto ao ambiente de desenvolvimento ODE, sendo esse um ponto negativo da ferramenta. Outra ferramenta similar ao estimaODE é o CustPlan, que também só é utilizado junto ao ambiente de desenvolvimento TABA.

A ferramenta Function Point Workbench calcula os pontos de função, permitindo decompor o sistema em módulos. Porém, a ferramenta restringe-se ao cálculo de pontos de função, sem auxiliar nas estimativas de esforço, duração e custo.

Conforme pesquisa feita por Faria Junior sobre ferramentas CASE para gerenciamento de projetos e métricas de software, a tabela 2.7 apresenta uma análise comparativa dos softwares CALICO, Costar, USC-COCOMO II e Cost Xpert 2.1 (FARIA JUNIOR, 2006).

Tabela 2.7 – Análise Comparativa entre os softwares CALICO, Costar, USC-COCOMO II e Cost Xpert 2.1

	Calico	Costar	USC-COCOMO II	Cost Xpert 2.1
Linguagem por LOC	C, Fortran, Cobol, Basic, Pascal, Pl/I, Ada	C, Fortran, Cobol, Basic, Pascal, Pl/I, Ada	Vide lista apresentada abaixo (linguagens de conversão LOC*)	Inclui todas as demais, excluindo apenas a linguagem "Report Generator" possui mais 500 incluindo variação de versão.
	Permite inclusão de nova Linguagem	Somente na definição de um novo modelo	Não permite inclusão de nova linguagem	Não permite inclusão de nova linguagem
Métricas/Indicadores	FP, COCOMO I, COCOMO II	FP, COCOMO II, 97, Early_Design, COCOMO_85, COCOMO_87, Ada,87, Aprm_88	FP, Linhas de código, COCOMO II	FP, SLOC, Feature Points, GUI Metrics, Object Metrics, Bottom Up, Top Down
Processo	Não Contem	Processo iterativo (4 variações pré-estabelecidas)	Ciclo de vida Clássico (Cascata)	Não Contem
Visualização Gráfica	Não Contem Visualização Gráfica	Contem Visualização Gráfica	Não Contem Visualização Gráfica	Contem Visualização Gráfica
Relatórios	Não emite relatórios	Emite Relatórios com gráficos	Emite relatórios apenas textuais	Emite Relatórios com gráficos
Recursos	Esforço (pessoa-Mês), Duração (pessoa-Mês)	Esforço (pessoa-Mês), Custo, Duração (Mês) ,	Esforço (pessoa-Mês) Custo, Duração (Mês),	Pessoal, Custos, Duração (Meses)
Help	Inexistente	Documentação On-Line (F1)	Manual	Documentação On-Line (F1), sensível ao Contexto

Fonte: Faria Junior, 2006.

Pode-se observar que tais softwares obtêm estimativas de tamanho por PF ou LOC para servir de entrada para a utilização de outras técnicas, como o COCOMO II. Além disso, não abrangem indicadores de produtividade e de qualidade, sendo assim pontos negativos destes softwares.

2.7 Conclusões do Capítulo

É notório que a utilização da APF para medir e estimar o tamanho de um software torna as estimativas mais precisas e confiáveis, tendo em vista a objetividade da técnica. Além disso, deve-se considerar o fato da APF poder ser utilizada em

diferentes etapas do projeto e independente da técnica de especificação de requisitos desenvolvida. Portanto, pode-se considerar a APF como principal técnica de dimensionamento de projetos de software existente atualmente.

Chega-se ainda a conclusão que a maioria dos softwares de estimativas não utilizam dados históricos para auxiliarem no cálculo de pontos de função e no cálculo dos demais indicadores. Além disso, não auxiliam no cálculo da produtividade, um dos principais indicadores para as estimativas e para o acompanhamento do rendimento da equipe.

Devido a todas essas informações, a construção de uma ferramenta de estimativa que atende as deficiências encontradas se mostra necessária e os resultados de seu desenvolvimento são apresentados nos capítulos posteriores.

Capítulo 3

Especificação de Requisitos

Uma completa compreensão dos requisitos é fundamental para o sucesso no desenvolvimento de um sistema. Por isso, há a necessidade de criar um modelo capaz de descrever o sistema sob a perspectiva externa, que seja compreensível tanto para os desenvolvedores do sistema quanto para os clientes e usuários.

Segundo Sommerville (2003), "(...) um requisito é tratado como funcional, quando descreve um serviço ou função que o sistema deve realizar (...)". Paralelamente podem haver requisitos não-funcionais, os quais são restrições impostas ao sistema ou ao desenvolvimento do sistema.

Para identificar os requisitos funcionais do sistema foi utilizada a técnica de modelagem de casos de uso, composta por diagramas e descrições dos casos de uso. Além disso, foi descrito o mini-mundo do sistema.

Neste capítulo, é apresentada a especificação de requisitos utilizada no desenvolvimento deste projeto. A seção 3.1 apresenta os requisitos funcionais, enquanto a seção 3.2 apresenta os requisitos não funcionais, responsáveis pelos requisitos técnicos e de qualidade.

3.1 Especificação de Requisitos Funcionais

A seguir estão descritas as funções do sistema, através da descrição do mini-mundo e a modelagem de casos de uso.

3.1.1 Descrição do Mini-mundo

Para auxiliar gerentes de projetos de desenvolvimento de software a estimar, surgiu a necessidade da construção de um software baseado em pontos de função, o FP Evolution. Tal software segue as características descritas abaixo.

O software tem o intuito de apoiar a gerência de projetos de software de uma organização, através de estimativas baseadas em pontos de função. Para tal, foram utilizados os conceitos definidos pelo IFPUG e NESMA, permitindo gerenciar as contagens de pontos de função dos projetos, e armazenando esses dados para que possam ser aproveitados para aprimorar os resultados futuros.

A ferramenta permitirá que uma organização gerencie várias aplicações, sendo que é desejado saber de cada aplicação o seu nome, descrição, esforço, custo, duração, tamanho, qualidade e produtividade. Cada aplicação pode ter vários projetos, sendo que os projetos devem conter: nome, descrição, tecnologia (LP, SO, etc), normas de qualidade (CMMI, ISO 12207, etc. ou customizado da organização), processos de desenvolvimento utilizados (RUP, XP, etc.), ferramentas (framework, SGBD, IDE, etc.), recursos utilizados, tamanho, esforço, duração, custo, quantidade de defeitos, produtividade, qualidade e custo/PF.

Os projetos que possuem características semelhantes devem ser separados por classes, sendo que uma classe contém: nome, descrição e a produtividade média para os projetos dessa classe. Podemos citar como exemplos de classes: “Sistema Operacional”, “Jogos”, “Softwares para celular”, “Editor de texto”, etc.

Cada projeto pode ter vários recursos humanos a serem utilizados, sendo que cada recurso possui uma carga horária no projeto e uma experiência no assunto daquele projeto. Deseja-se saber ainda os seguintes dados do Recurso: Nome, Data de Nascimento, Telefone, E-mail, Cursos, Observações e Experiência em Desenvolvimento de Software.

Um projeto pode ter várias contagens, que podem ser do tipo aplicação (Contagem Inicial ou Contagem após melhoria), desenvolvimento ou melhoria. Além disso, deve ser escolhida qual das três diferentes abordagens técnicas será utilizada: estimativa, indicativa ou detalhada. Uma contagem contém: versão do CPM, descrição do escopo, propósito da contagem, data, responsável, observações e sua quantidade de PF. No caso de contagens do tipo melhoria é possível ainda utilizar um deflator, que consiste em um valor para influenciar no tamanho de funções adicionadas, alteradas e excluídas.

Para calcular a quantidade de PF de uma contagem é preciso saber suas funcionalidades. As funcionalidades estão separadas em funções do tipo dado e funções do tipo transação. São funções do tipo dado: Arquivo Lógico Interno (ALI) e Arquivo de Interface Externa (AIE). São funções do tipo transação: Entrada Externa (EE), Saída Externa (SE) e Consulta Externa (CE). As funcionalidades do software podem conter várias funções do tipo dado e várias funções do tipo transação.

As funções contêm: Grupo de Dados / Processo Elementar, tipo (ALI, AIE, EE, SE ou CE), quantidade de Tipos de Dados (TD), quantidade de Tipos de Registros (TRs) / Arquivos Referenciados (ARs), complexidade, quantidade de pontos de função.

Para saber o tamanho de um software em pontos de função, além de saber as funcionalidades é preciso calcular um fator de ajuste para se adequar a outras características do software, tais como atualização online, usabilidade, etc. O fator de ajuste contém 14 características gerais do sistema que devem ser preenchidas com valores de 0 a 5 de acordo com a sua influência. Mas, o fator de ajuste é opcional. Se o fator de ajuste for preenchido o tamanho é considerado como pontos de função ajustados, senão é considerado como pontos de função não-ajustados.

Um projeto é composto por várias tecnologias. Podemos citar como exemplos de tecnologia: linguagem de programação; sistema operacional; banco de dados; etc. Essas tecnologias podem estar associadas a vários projetos. Deseja-se armazenar das tecnologias o nome, tipo e observações.

O projeto pode possuir várias normas de qualidade. Devem-se armazenar os seguintes dados da norma: nome, observações e nível (CMMI Nível II, por exemplo).

Existem diversos processos de desenvolvimento que podem ser utilizadas em um projeto, tais como RUP, XP (Xtreme Programming), RAD (Rapid Application Development), etc. Um processo pode estar associado a diversos projetos. São dados do processo: nome, observações.

São utilizadas diversas ferramentas em projetos, tais como: Ambiente de Desenvolvimento; Framework; Ferramenta Case; SGBD; Ferramenta de

Gerenciamento de Projetos; Ferramenta de Documentação; etc. Deve-se armazenar os seguintes dados: nome, tipo, versão e observações.

A produtividade é calculada através de projetos anteriores cadastrados na base histórica e através da produtividade e do tamanho é possível estimar o esforço, a duração e o custo do projeto. Também é possível calcular o indicador de qualidade do projeto através da quantidade de defeitos informada. Deseja-se saber de cada estimativa: Título, Data, Total de PF, Produtividade, Custo/PF, Defeitos, Esforço, Prazo, Custo e Qualidade.

Através da ferramenta será possível ainda gerar três relatórios. O primeiro corresponde ao relatório do projeto, que contém todos os dados gerais do projeto, como também a relação do estimado x real e fator de crescimento. O segundo relatório exibe todos os projetos agrupados pela classe, exibindo as informações: nome; classe; tamanho; produtividade e custo/PF (da classe e do projeto). O terceiro relatório irá exibir uma relação entre as características no projeto e a sua produtividade média agrupados pelos tipos de registros (Ferramentas, Tecnologias, Normas de Qualidade e Processos de Desenvolvimento) para, por exemplo, saber a produtividade média dos projetos que usam Java.

3.1.2 Modelagem dos Requisitos

As descrições de todos os casos de uso encontram-se no Anexo A. Segue abaixo uma descrição sucinta dos casos de uso com os respectivos diagramas, separados por pacote.

3.1.2.1 Diagrama de Pacotes

È conveniente utilizar-se de um elemento para organizar os subsistemas do modelo. Para isto utilizam-se os diagramas de pacote. Segundo Deboni (2008), "(...) um pacote representa um grupo de classes (ou outros elementos) que se relaciona com outros pacotes através de uma relação de dependência (...)". Um diagrama de pacotes pode ser utilizado em qualquer fase do processo de modelagem e visa organizar os modelos.

A figura 3.1 mostra o diagrama de pacotes deste trabalho, que é constituído dos pacotes Projeto, Registro e Estimativa.

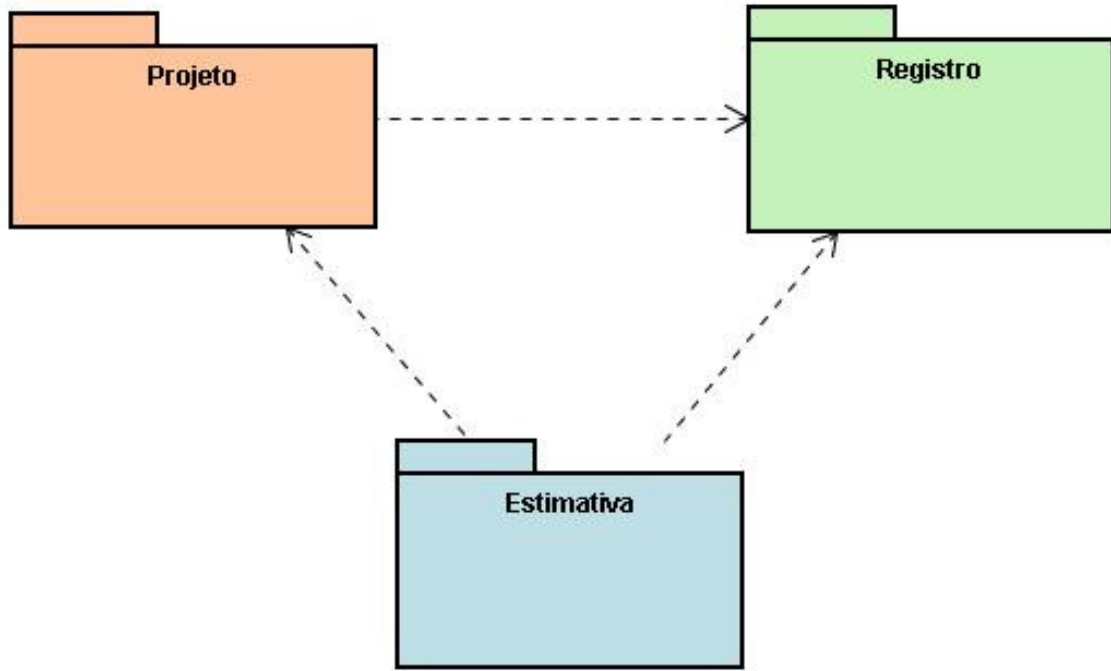


Figura 3.1 - Diagrama de Pacotes

3.1.2.2 Pacote de Registros

Corresponde aos cadastros de apoio do sistema. As informações desses cadastros poderão ser utilizadas para caracterizar um projeto, e assim aproveitá-lo de melhor forma na realização de novas estimativas com base em dados históricos, ou ainda, para geração de dados estatísticos em possíveis relatórios. Esses cadastros têm o intuito de abranger alguns itens do questionário elaborado pelo ISBSG (Anexo B) para submeter projetos à sua base histórica.

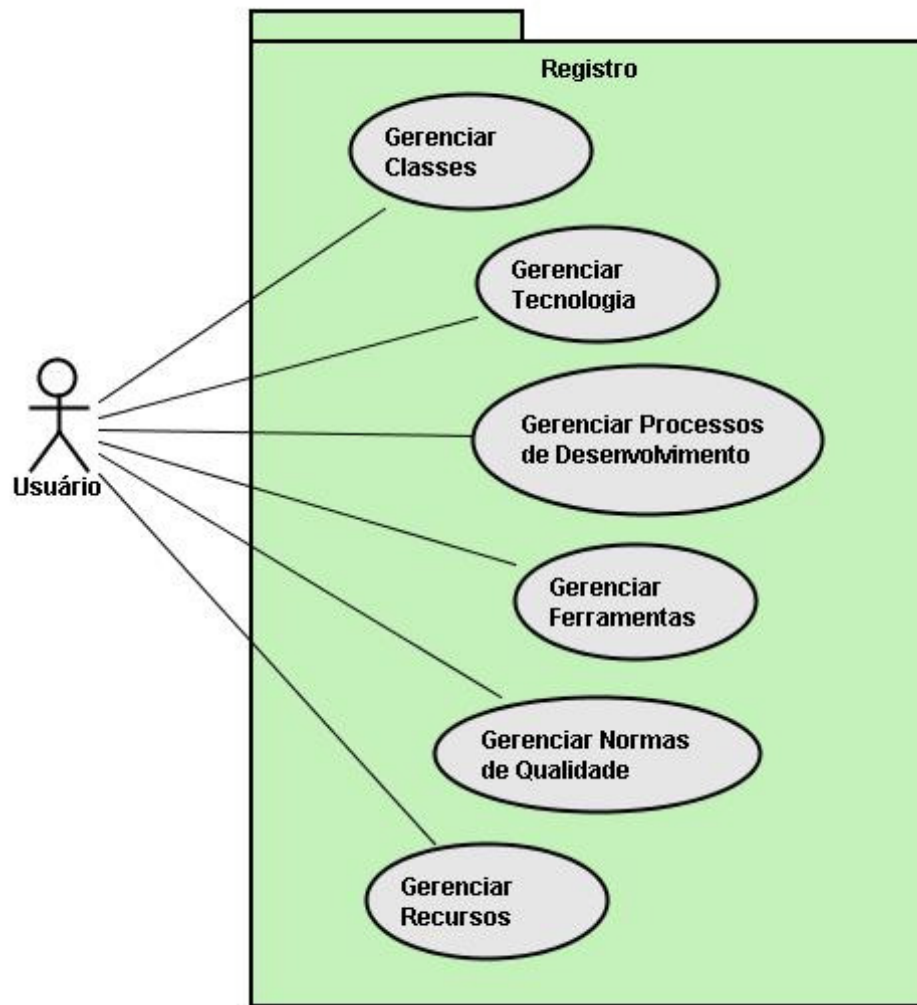


Figura 3.2 – Diagrama de Casos de Uso do pacote Registro

Caso de Uso	Descrição
Gerenciar Classes	Descreve as operações do sistema para incluir, alterar, excluir classes de projetos no sistema. Uma classe é como uma categoria de um conjunto de projetos com características semelhantes. Por exemplo: Sistema Operacional, Editor de Texto, Jogos e outros.
Gerenciar Tecnologia	Descreve as operações do sistema para incluir, alterar, excluir registros de tecnologias no sistema. Exemplos: Linguagem de Programação: JAVA; Banco de Dados: Postgre.

Gerenciar Ferramentas	Descreve as operações do sistema para incluir, alterar e excluir as ferramentas a serem utilizadas nos projetos do sistema. Exemplos: Framework: Hibernate; IDE: Visual Studio 2008; SGBD: IBExpert.
Gerenciar Processos de Desenvolvimento	Descreve as operações do sistema para incluir, alterar, excluir registros de processos de desenvolvimento no sistema. Exemplos de processos de desenvolvimento: RUP, XP, RAD entre outros.
Gerenciar Normas de Qualidade	Descreve as operações do sistema para incluir, alterar e excluir Normas de Qualidade de projetos no sistema. Exemplos: CMMI nível 1; MPS.BR nível F.
Gerenciar Recursos	Esse caso de uso descreve as operações do sistema para incluir, alterar e excluir recursos humanos no sistema, para que possam ser utilizados em projetos.

3.1.2.3 Pacote de Projeto

Neste pacote estão incluídos os casos de uso diretamente ligados à gerência dos projetos, como também a realização de estimativas e contagens dos projetos.

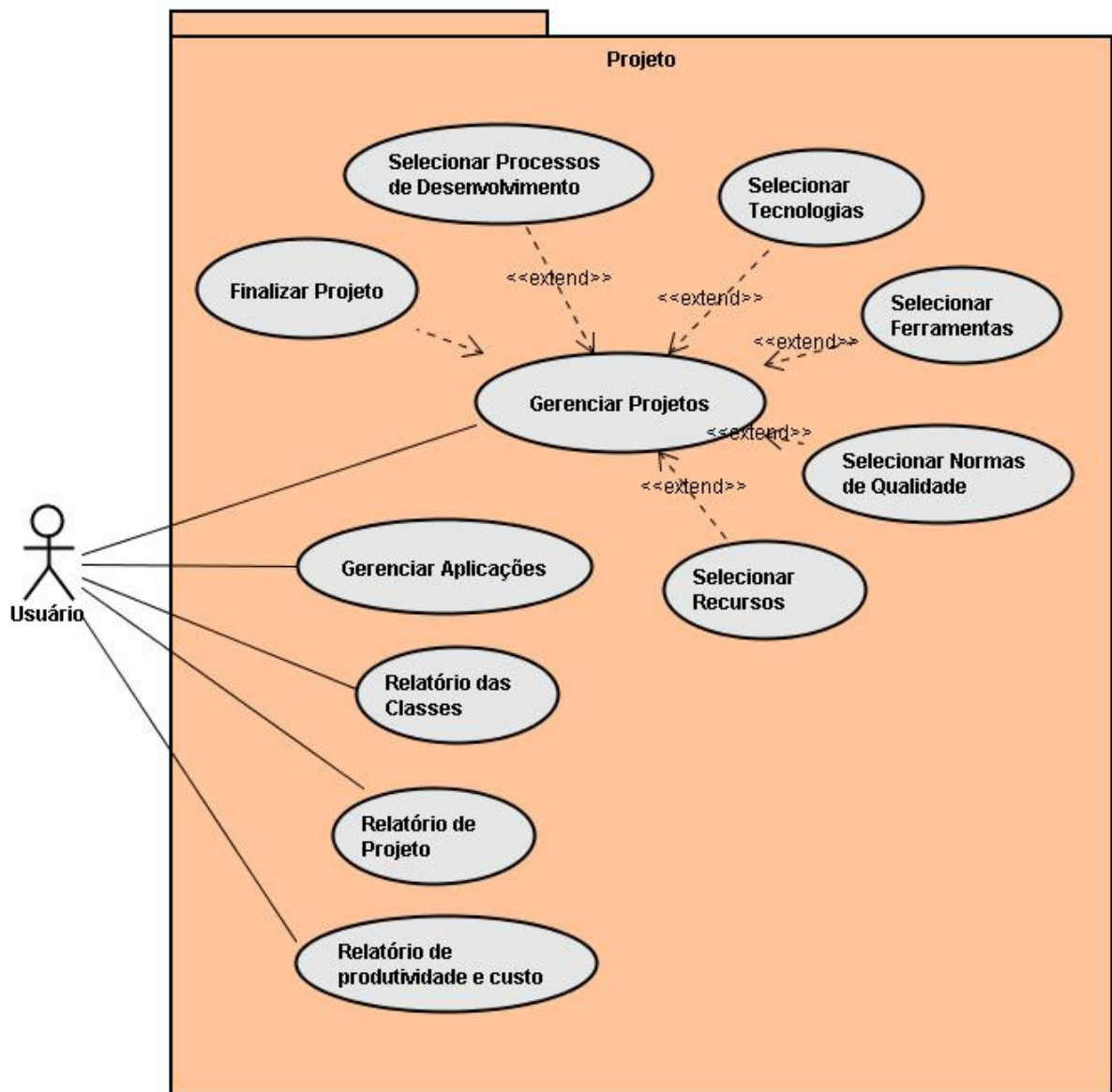


Figura 3.3 – Diagrama de Casos de Uso do pacote Projeto

Caso de Uso	Descrição
Gerenciar Projetos	Descreve as operações para incluir, alterar e excluir projetos no sistema.
Selecionar Tecnologia	Descreve as operações para adicionar e remover tecnologias no projeto.
Selecionar Ferramentas	Descreve as operações para adicionar e remover ferramentas no projeto.

Selecionar Processos de Desenvolvimento	Descreve as operações para adicionar e remover processos de desenvolvimento no projeto.
Selecionar Normas de Qualidade	Descreve as operações para adicionar e remover normas de qualidade no projeto.
Selecionar Recursos	Descreve as operações para adicionar e remover recursos humanos no projeto, informando a experiência do recurso no contexto do projeto e a quantidade de horas de trabalho por dia.
Finalizar Projeto	Descreve as operações para indicar que um projeto da aplicação está finalizado, ou seja, já possui valores reais que possam ser usados em estimativas.
Gerenciar Aplicações	Descreve as operações do sistema para incluir, alterar, excluir aplicações no sistema. Inclui informações gerais de uma aplicação.
Gerar Relatório de Projeto	Descreve as operações do sistema para gerar o Relatório de Projeto, que consiste em mostrar os dados gerais de todos os projetos, como também a relação do valor estimado e real, indicando ainda o fator de crescimento.
Gerar Relatório das Classes	Descreve as operações do sistema para gerar o Relatório das Classes, que consiste em exibir todos os projetos agrupados pela classe.
Gerar Relatório de Produtividade e Custo	Descreve as operações do sistema para gerar o Relatório de Produtividade e Custo, que consiste em exibir uma relação entre as características do projeto (Ferramentas, tecnologias, Normas de Qualidade e Processos de Desenvolvimento) e a sua produtividade média e custo.

3.1.2.4 Pacote de Estimativa

Neste pacote estão incluídos os casos de uso referentes às contagens e estimativas. Mostram como devem ser feitos os diferentes cálculos de pontos de função dependendo do tipo e abordagem da contagem. Além disso, descreve como são feitas as estimativas.

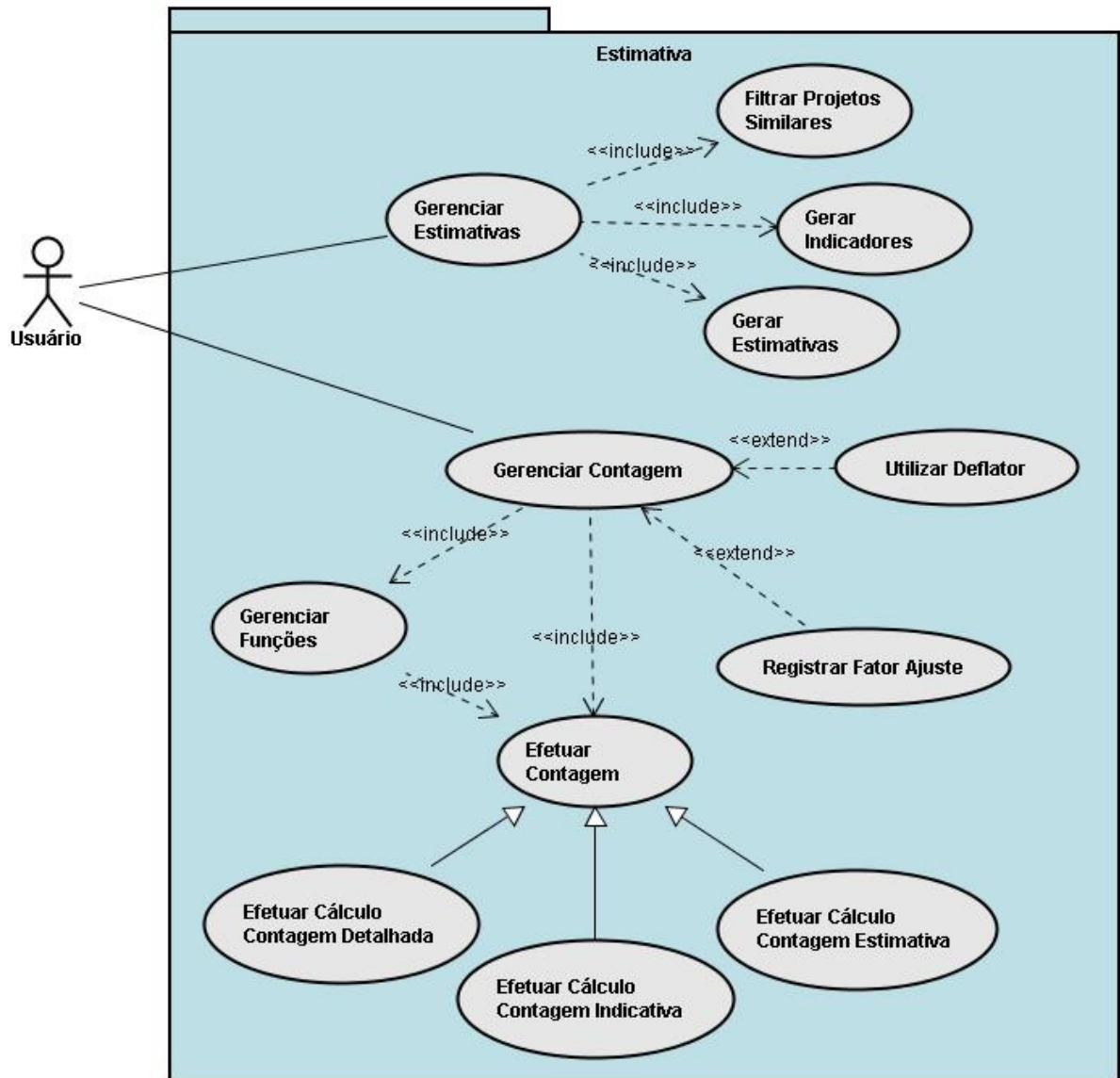


Figura 3.4 – Diagrama de Casos de uso do pacote Estimativa

Caso de Uso	Descrição
Gerenciar Contagem	Descreve as operações para incluir, alterar e excluir contagens no sistema.

Gerenciar Funções	Descreve as operações do sistema para incluir, alterar e excluir funções nas contagens de projetos do sistema. Essas funções são a base para o cálculo de pontos de função.
Efetuar Contagem	Descreve as operações gerais do sistema para calcular a quantidade de pontos de função de uma contagem.
Efetuar Cálculo da Contagem Detalhada	Descreve as operações do sistema para calcular a quantidade de pontos de função de uma contagem, a partir da abordagem detalhada.
Efetuar Cálculo da Contagem Indicativa	Descreve as operações do sistema para calcular a quantidade de pontos de função de uma contagem, a partir da abordagem indicativa.
Efetuar Cálculo da Contagem Estimativa	Descreve as operações do sistema para calcular a quantidade de pontos de função de uma contagem, a partir da abordagem estimativa.
Registrar Fator de Ajuste	Descreve as operações do sistema para registrar o fator de ajuste em uma contagem de pontos de função.
Utilizar Deflador	Descreve as operações para utilizar o deflador numa contagem de melhoria, influenciando assim no tamanho da contagem.
Gerenciar Estimativas	Descreve as operações para incluir, alterar e excluir estimativas no sistema.
Filtrar Projetos Similares	Descreve as operações do sistema para filtrar projetos similares que serão utilizados para a geração de indicadores de uma estimativa.

Gerar Indicadores	Descreve as operações do sistema para calcular os indicadores de produtividade, custo/PF e qualidade de uma estimativa e juntamente com o tamanho definirá as estimativas de esforço, prazo, custo e defeitos.
Gerar Estimativas	Descreve as operações do sistema para calcular as estimativas de esforço, custo, prazo e quantidade de defeitos de uma estimativa.

3.2 Especificação de Requisitos Não Funcionais

Os requisitos não-funcionais representam as propriedades ou requisitos de qualidade que um sistema precisa atender. Eles têm papel relevante por servir de critério na seleção e/ou composição de uma arquitetura de software. Descrevem não o que o software fará, mas como fará. Pode-se citar, por exemplo, requisitos de desempenho, da interface externa, de projeto e da qualidade (THAYER, 1990).

O sistema apresenta algumas particularidades quanto a esses fatores que são apresentadas a seguir e estão divididas nas categorias de requisitos de processo e de produto.

3.2.1 Requisitos de Processo

Os requisitos de processo estão relacionados ao processo de desenvolvimento do sistema.

Identificação	Descrição
Linguagem de Programação	O sistema será desenvolvido utilizando a linguagem de programação Java.
Modelagem	Todo o sistema deverá ser modelado utilizando a linguagem UML.

3.2.2 Requisitos de Produto

Os requisitos de produto estão relacionados às características desejadas para o sistema.

3.2.2.1 Segurança

Identificação	Descrição
Integridade	Os dados armazenados e consultados deverão estar corretos em relação aos dados fornecidos ao sistema.

3.2.2.2 Usabilidade

Identificação	Descrição
Manutenibilidade	A manutenção do sistema deve ser fácil, visando gastar o menor tempo possível com alterações. Essa preocupação se dá devido às possíveis mudanças no Manual de Práticas de Contagem feito pelo IFPUG.
Portabilidade	O sistema deve poder ser implantado independente do sistema operacional.

3.3 Conclusões do Capítulo

Neste capítulo foram definidos e especificados os requisitos funcionais e não funcionais do projeto. A definição dos requisitos do sistema é fundamental para as fases seguintes do trabalho. Qualquer ponto que não tenha ficado claro implicará em transtornos no futuro. O próximo passo é a fase da análise, em que é feita a modelagem do sistema, conforme apresentado no capítulo seguinte.

Capítulo 4

Análise do Sistema

Segundo Larman (2003), “(...) a análise orientada a objetos se preocupa com a criação de uma descrição do domínio, a partir da perspectiva dos objetos (...)”, ou seja, tem como intuito desenvolver um modelo que descreva o funcionamento do sistema de modo a atender um conjunto de requisitos pré-definidos. Durante a análise, a ênfase está em encontrar e descrever objetos que estejam no domínio do problema e que sejam relevantes para o sistema que se pretende construir. Por meio dela representa-se um sistema real de forma abstrata e simplificada.

De posse dos requisitos levantados para o sistema, descritos no capítulo anterior, a fase de análise é iniciada. Nesta fase são elaboradas tanto a modelagem estática quanto a modelagem comportamental do sistema. A seção 4.1 mostra a modelagem estática, através de um modelo de classes e a seção 4.2 mostra a modelagem de comportamento, através do diagrama de atividades e dos diagramas de seqüência dos principais casos de uso.

4.1 Modelagem Estática

A modelagem estática visa identificar os conceitos do mundo real relevantes para o sistema. No caso do paradigma Orientado a Objetos, o modelo estático é suportado principalmente pelo diagrama de classes que envolve a identificação de classes, atributos e associações, bem como o agrupamento de classes em subsistemas ou pacotes. A seguir, são apresentados os resultados da análise, no que tange aos aspectos de informação.

A figura 4.1 apresenta, a seguir, o diagrama de classes do projeto, diferenciando as classes de cada pacote através das cores.

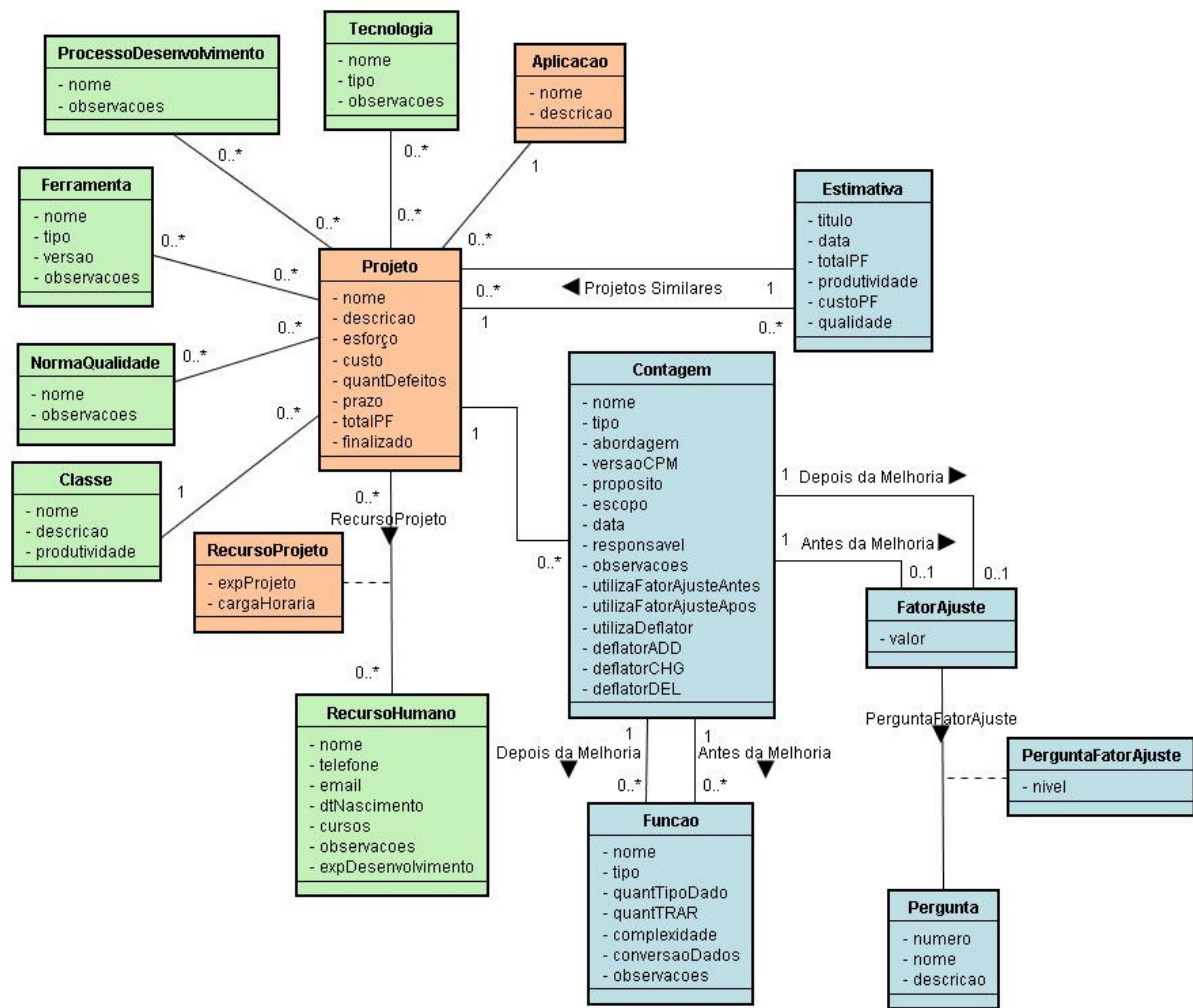


Figura 4.1 – Diagrama de Classes

As seguintes classes com os seus respectivos atributos foram identificadas:

- **Projeto:** Contém todas as informações de um projeto.
 - nome: Nome do projeto;
 - descricao: Descrição do projeto;
 - esforço: Esforço real do projeto;
 - custo: Custo real do projeto;
 - prazo: Prazo real do projeto;
 - quantDefeitos: Quantidade de defeitos do projeto;
 - totalPF: Total de Pontos de Função do projeto;
 - finalizado: Indicador para informar se o projeto está finalizado.

- **Contagem:** Contém todas as informações sobre uma contagem de pontos de função de um determinado projeto.
 - nome: Nome/Título da contagem;
 - tipo: Este atributo contém qual será o tipo da Contagem: Desenvolvimento; Aplicação - Contagem Inicial; Melhoria; Aplicação - Após Melhoria. Cada tipo determina qual fórmula será usada para a realização do cálculo da quantidade total de PF da Contagem;
 - abordagem: Este atributo determina qual das 3 abordagens será usada para calcular a quantidade de pontos de função não ajustados: Indicativa; Estimativa; Detalhada. Cada abordagem determina a fórmula para o cálculo de pontos de função não ajustados e quais valores serão usados para cada tipo de função (ALI, AIE, EE, SE e CE);
 - versaoCPM: Qual a versão da CPM (Manual de Praticas de Contagem) foi usada na Contagem;
 - propósito: Informações relacionadas a um objetivo mais abrangente da contagem, auxiliando na determinação do tipo da contagem, na delimitação do escopo, entre outros dados;
 - escopo: informações que contém quais subsistemas ou funcionalidades estarão incluídas na contagem;
 - data: Data da realização da contagem;
 - responsável: Nome da pessoa que realizou a contagem;
 - observações: Atributo para guardar as observações feitas pelo Usuário;
 - utilizaFatorAjusteAntes: indicador para dizer se utiliza o Fator de Ajuste nas funções de antes da melhoria;
 - utilizaFatorAjusteDepois: indicador para dizer se utiliza o Fator de Ajuste nas funções de depois da melhoria, de aplicação ou de desenvolvimento;
 - utilizaDeflator: indicador para dizer se utiliza um deflator nas contagens de melhoria;
 - deflatorADD: deflator de funções adicionadas;
 - deflatorCHG: deflator de funções alteradas;
 - deflatorDEL: deflator de funções deletadas.

- **Funcao:** Contém os dados gerais de uma função.
 - nome: Nome da Função.

- Tipo: Pode ser uma Função do Tipo Dado ou uma Função do Tipo Transação. Uma Função Tipo Dado pode ser de dois tipos: Arquivo Lógico Interno (ALI); Arquivo de Interface Externa (AIE). Uma Função Tipo Transação pode ser de três tipos: Entrada Externa (EE); Saída Externa (SE); Consulta Externa (CE). Cada tipo determina os valores para a relação entre quantidade de PF e a complexidade da função.
 - quantTipoDado: Número de Tipo de Dados (TD). Um tipo dado é um campo único reconhecido pelo usuário, não repetido;
 - quantTRAR: Quantidade de Tipos de Registros / Arquivos Referenciados.
 - complexidade: Valor da complexidade da Função, pode ser :Baixa, Média ou Alta. Este valor é determinado pela relação da quantidade de PF e o seu Tipo;
 - conversaoDados: indicador se a função é de conversão ou não.
 - observações: observações da função.
- **FatorAjuste:** Contém o fator de ajuste de uma contagem.
 - valor: Valor total do fator de ajuste.
- **Pergunta:** Contém as características gerais do fator de ajuste.
 - numero: Número da pergunta;
 - nome: Nome do item, exemplo: Usabilidade;
 - descricao: Descrição da pergunta e todas as alternativas;
- **PerguntaFatorAjuste:** Contém o nível de influência das características gerais do fator de ajuste.
 - nível: Número do nível de influência do item. Os valores podem ser de 0 a 5.
- **Aplicacao:** Contém todas as informações das aplicações.
 - nome: Nome da aplicação. Ex.: Sistema de Controle de Ponto, etc.;
 - descricao: Descrição da aplicação;
- **Estimativa:** Contém todas as informações das estimativas de projetos.
 - titulo: Título da estimativa;
 - data: Data da estimativa;

- totalPF: Total de Pontos de Função utilizado na estimativa;
 - produtividade: Produtividade utilizada para estimativa;
 - custoPF: Custo por Pontos de Função utilizado na estimativa;
 - qualidade: Qualidade utilizada para a estimativa;
- **RecursoHumano:** Contém todas as informações dos recursos humanos.
 - nome: Nome do recurso;
 - telefone: Telefone do recurso;
 - email: E-mail do recurso;
 - dtNascimento: Data de nascimento;
 - cursos: Cursos do recurso;
 - observações: Observações do recurso;
 - expDesenvolvimento: Experiência em desenvolvimento de software.
- **RecursoProjeto:** Contém todas as informações dos recursos dos projetos.
 - expProjeto: Experiência do recurso no assunto do projeto. Ex.: Experiência em Contabilidade para desenvolver um Sistema de Contabilidade;
 - cargaHoraria: Carga horária do recurso no projeto;
- **Classe:** Contém todas as informações das classes de projetos.
 - nome: Nome da classe. Ex.: Sistema Operacional, Jogos, etc.;
 - descricao: Descrição da classe;
 - produtividade: Produtividade média de todos os projetos da classe.
- **Tecnologia:** Contém todas as informações de tecnologia.
 - nome: Nome da tecnologia. Ex.: Java, Firebird, etc.;
 - tipo: Tipo de tecnologia, sendo que existem os seguintes tipos: Linguagem de Programação, Banco de Dados, Sistema Operacional, Outras;
 - observação: Observações sobre a tecnologia.
- **ProcessoDesenvolvimento:** Contém todas as informações de um processo de desenvolvimento.
 - nome: Nome do processo de desenvolvimento. Ex.: RUP, XP, etc.;

- observação: Observações sobre o processo de desenvolvimento.
- **Ferramenta:** Contém todas as informações de ferramentas.
 - nome: Nome da ferramenta. Ex.: NetBeans, Jude, etc.;
 - tipo: Tipo da ferramenta, sendo que existem os seguintes tipos: Ambiente de Desenvolvimento (IDE), Framework, Ferramenta Case, SGBD, Gerenciamento de Projetos, Documentação, Outra;
 - versão: Versão da ferramenta;
 - observação: Observações sobre a ferramenta.
- **NormaQualidade:** Contém as informações referente à normas de qualidade.
 - nome: Nome da norma de qualidade. Ex.: CMMI, 12207, etc.;
 - observação: Observações sobre a norma de qualidade.

4.2 Modelagem do Comportamento

A modelagem do comportamento dos objetos visa apoiar a identificação de atributos e operações de classes. Nesta seção, são apresentados o Diagrama de Atividades, contendo o fluxo principal do sistema, e os Diagramas de Seqüência dos eventos “Incluir Contagem” e “Incluir Estimativa” dos casos de Uso “Gerenciar Contagens” e “Gerenciar Estimativas”, respectivamente.

Os resultados desta atividade já foram espelhados no Diagrama de Classes mostrados na seção anterior.

4.2.1 Diagrama de Atividade do sistema

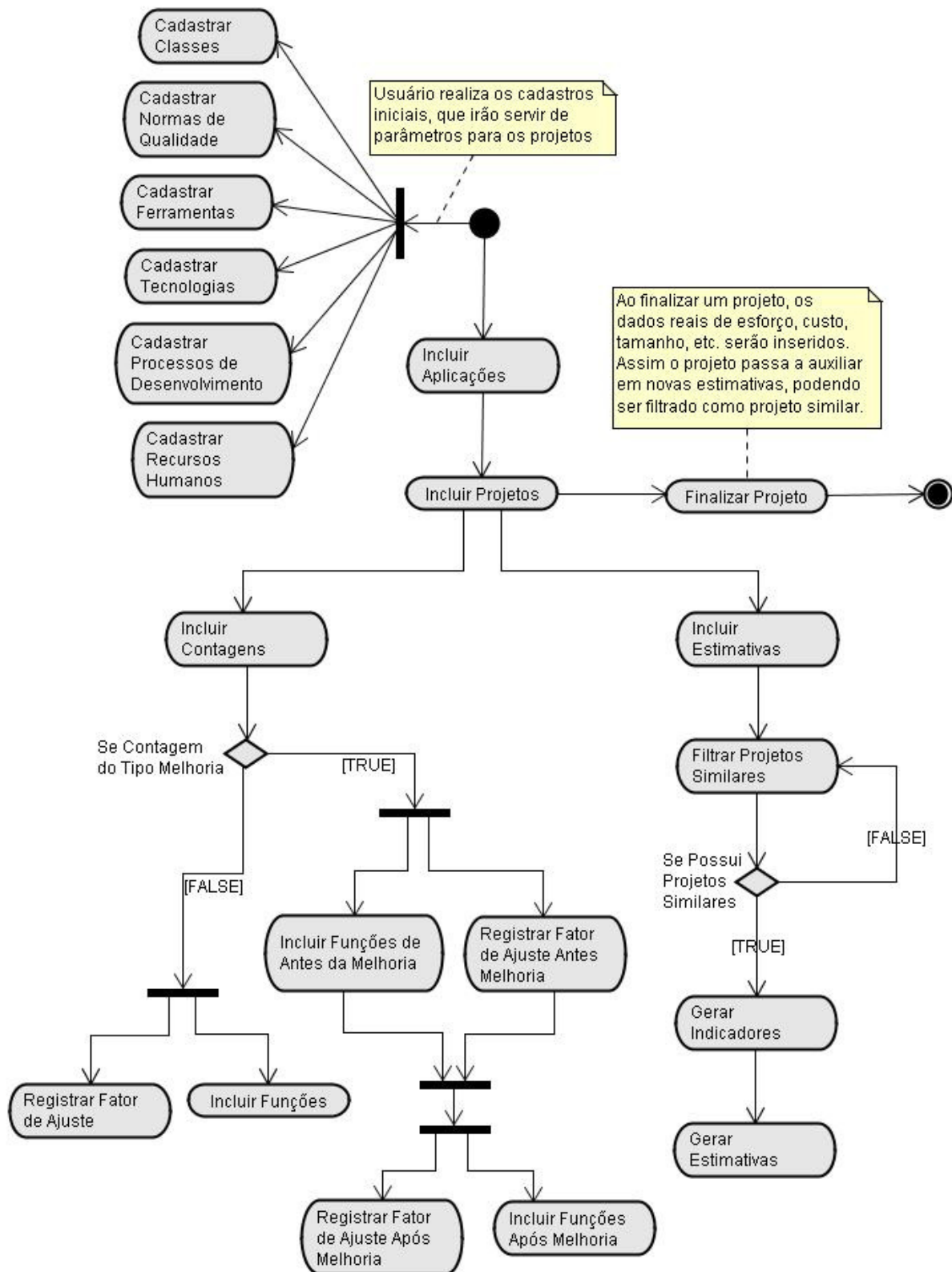


Figura 4.2 – Diagrama de Atividade

4.2.2 Diagrama de Seqüência – Incluir Contagem

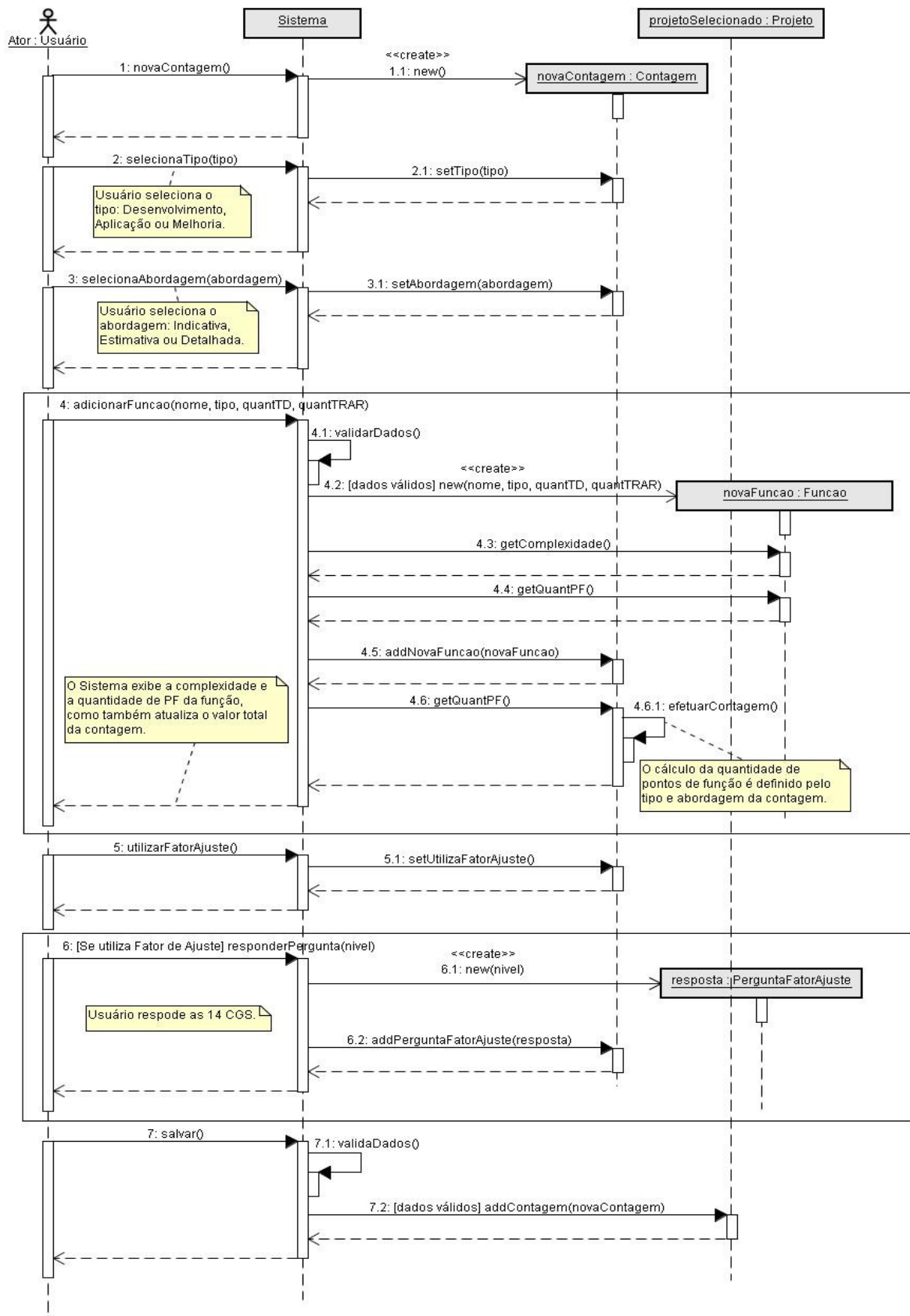


Figura 4.3 – Diagrama de Seqüência - Incluir Contagem

4.2.3 Diagrama de Seqüência – Incluir Estimativa

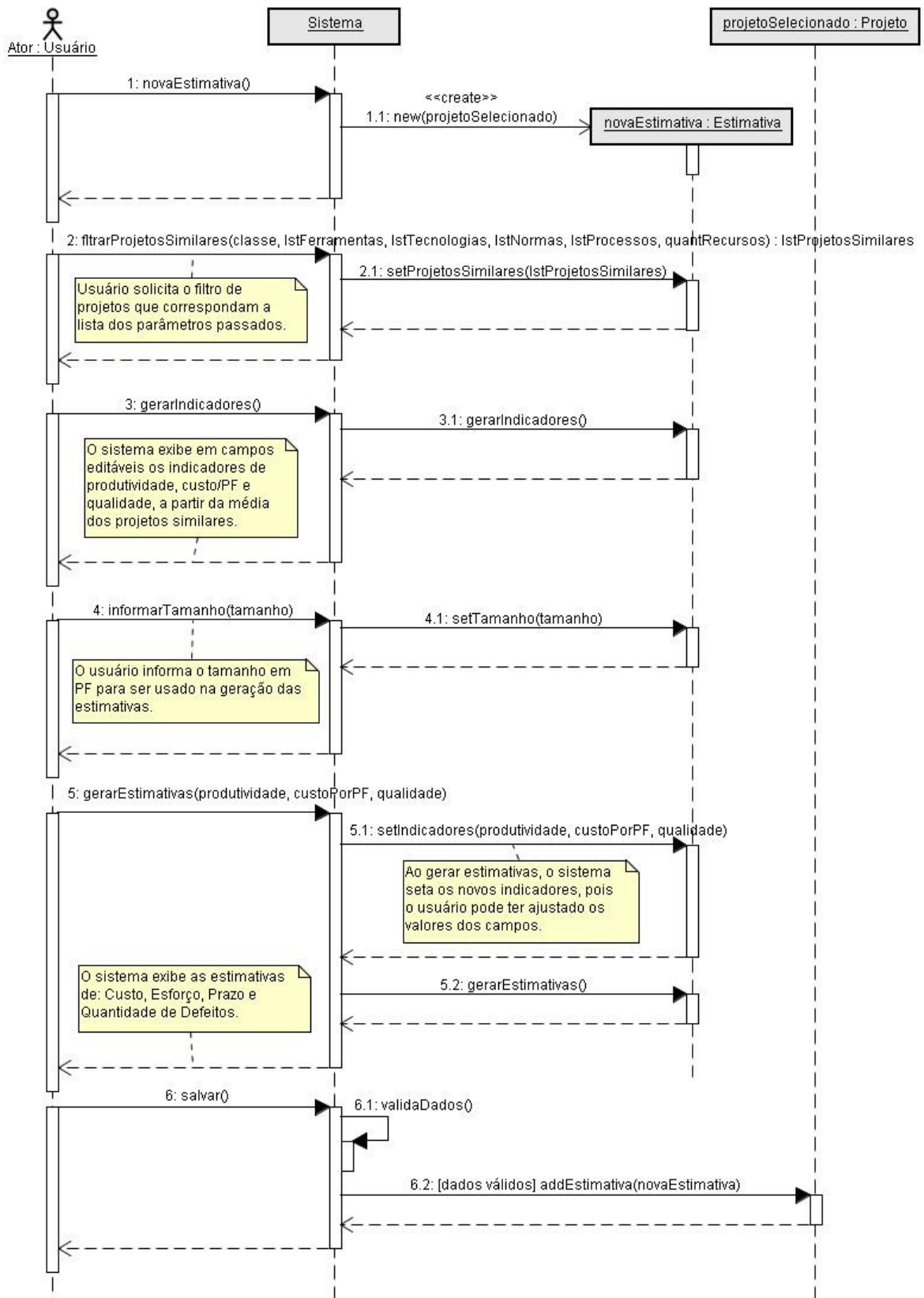


Figura 4.4 – Diagrama de Seqüência - Incluir Estimativa

4.3 Conclusões do Capítulo

A análise do sistema foi realizada através da modelagem estática e de comportamento. Para a modelagem estática foi utilizado um diagrama de classes, enquanto para a modelagem comportamental foram optados os diagramas de seqüências e um diagrama de atividade para informar o fluxo principal do sistema.

Como a fase de análise considera a tecnologia perfeita, o próximo passo é definir na fase de projeto a modelagem de como o sistema será implementado e envolver os requisitos tecnológicos aos modelos aqui criados.

Capítulo 5

Projeto do Sistema

Segundo Larman (2003), “(...) o projeto orientado a objetos se preocupa com a definição de objetos de software e suas responsabilidades e colaborações (...)”, ou seja, tem como intuito modelar a maneira como o sistema será implementado, agregando requisitos tecnológicos ao modelo de análise. Sendo assim, o projeto depende da definição de aspectos como a linguagem de programação a ser utilizada, o modelo de persistência de dados, a arquitetura do sistema e a interface com o usuário.

Outro aspecto a ser considerado neste capítulo é a modularização do sistema, através da divisão em camadas, o que facilita seu uso e manutenção. A utilização de padrões de projeto OO também é um assunto a ser tratado, tendo em vista que estes padrões auxiliam para um desenvolvimento mais organizado.

Sendo assim, neste capítulo será apresentado o projeto da plataforma de implementação, a arquitetura do sistema, a divisão em camadas com suas respectivas responsabilidades e, por último, os padrões de projeto OO a serem utilizados.

5.1 Plataforma de Implementação

O sistema proposto será implementado usando a linguagem de programação Java, utilizando Swing para construção das interfaces na plataforma desktop com persistência em arquivo. Foi escolhida essa plataforma de implementação devido ao maior conhecimento técnico em tecnologias Desktop pelos desenvolvedores deste projeto durante o período estipulado para implementação, viabilizando assim o tempo para desenvolvimento. Além disso, o principal objetivo deste trabalho foi implementar o máximo de requisitos funcionais possíveis.

5.2 Arquitetura do Sistema

A organização de classes em pacotes deve ser o ponto de partida para a definição da arquitetura do sistema, já que é um meio de estabelecer níveis de abstração para o modelo. Esses níveis de abstração podem ser organizados em camadas e, assim, tratados separadamente durante a fase de projeto. A organização de classes em pacotes é útil também para permitir a produção de componentes para reuso.

Neste trabalho, foram utilizadas duas formas complementares de agrupamento de classes em pacotes: primeiramente pelos estereótipos definidos pela arquitetura de quatro camadas proposta por Coad e Yourdon, contendo os seguintes estereótipos: Domínio do Problema, Gerência de Dados, Gerência de Tarefas e Interface (apud FALBO, 2003). Podendo posteriormente cada um ser subdividido pelo domínio do problema, aproveitando os pacotes definidos na fase de análise.

O diagrama da figura 5.1 mostra as dependências entre os estereótipos: domínio do problema, gerência de dados, gerência de tarefas e interface. Cada estereótipo é subdividido pelos pacotes definidos na análise, sendo eles: projeto, estimativa e registro.

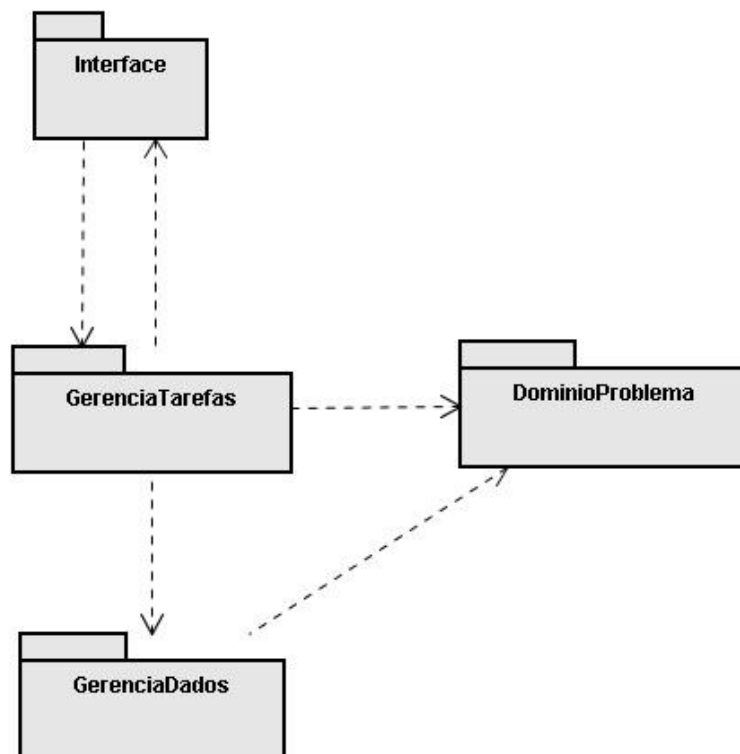


Figura 5.1 – Diagrama de Pacotes por Estereótipo

Nesta abordagem arquitetural, o sistema foi dividido inicialmente em pacotes de acordo com os objetivos de cada camada da arquitetura, onde cada uma destas possui objetivos e responsabilidades específicas. Segue abaixo, a descrição das responsabilidades comuns das classes de cada pacote:

- **Interface:** interagir com o usuário.
- **Gerência de Tarefas:** receber requisições da camada de interface; gerenciar a navegação das telas; e realizar as funcionalidades de casos de uso, invocando métodos do domínio.
- **Domínio do Problema:** representar as classes do domínio do problema.
- **Gerência de Dados:** realizar a persistência dos dados.

Para cada um dos pacotes apresentados haverá outro nível de decomposição baseado na divisão em pacotes de acordo com as classes do domínio do problema, por motivo de organização, conforme apresentado a seguir.

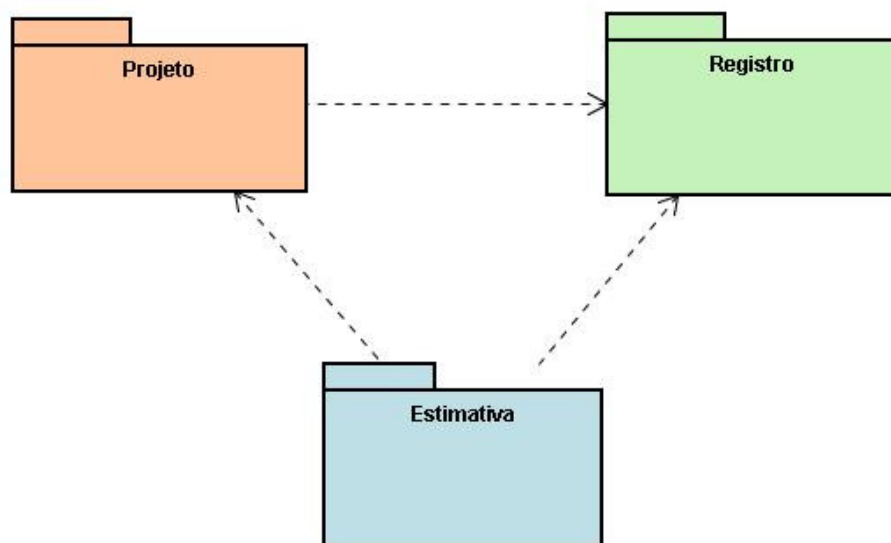


Figura 5.2 – Diagrama de Pacotes do Domínio do Problema

Assim, o sistema estará modularizado de forma a facilitar a manutenção e o reuso do código, incentivando assim a sua evolução. Com essa arquitetura é possível desenvolver, por exemplo, a interface Web modificando a camada de Interface e fazendo alguns ajustes na camada de Gerência de tarefas. Da mesma forma pode ser feita a transposição da camada de gerência de dados de arquivo para banco de dados.

5.3 Camadas da Arquitetura

Nesta seção são apresentadas as quatro camadas do sistema, considerando as classes e arquivos a serem criados. A seção 5.3.1 discute a camada de domínio do problema, com os elementos descritos na fase de requisitos; a seção 5.3.2 mostra a camada de gerência de tarefas, com as aplicações responsáveis por executar os casos de uso e controlar as requisições dos usuários; a seção 5.3.3 apresenta a camada de interface, com os elementos de interação com o usuário e a seção 5.3.4 discute a camada de gerência de dados, com as classes responsáveis pela persistência do sistema.

5.3.1 Camada de Domínio do Problema

Nesta seção são apresentadas as classes do domínio do problema, considerando a modelagem de análise, mais os aspectos de construção, relacionados ao projeto (características da linguagem, notação de projeto, reuso, desempenho, usabilidade etc.).

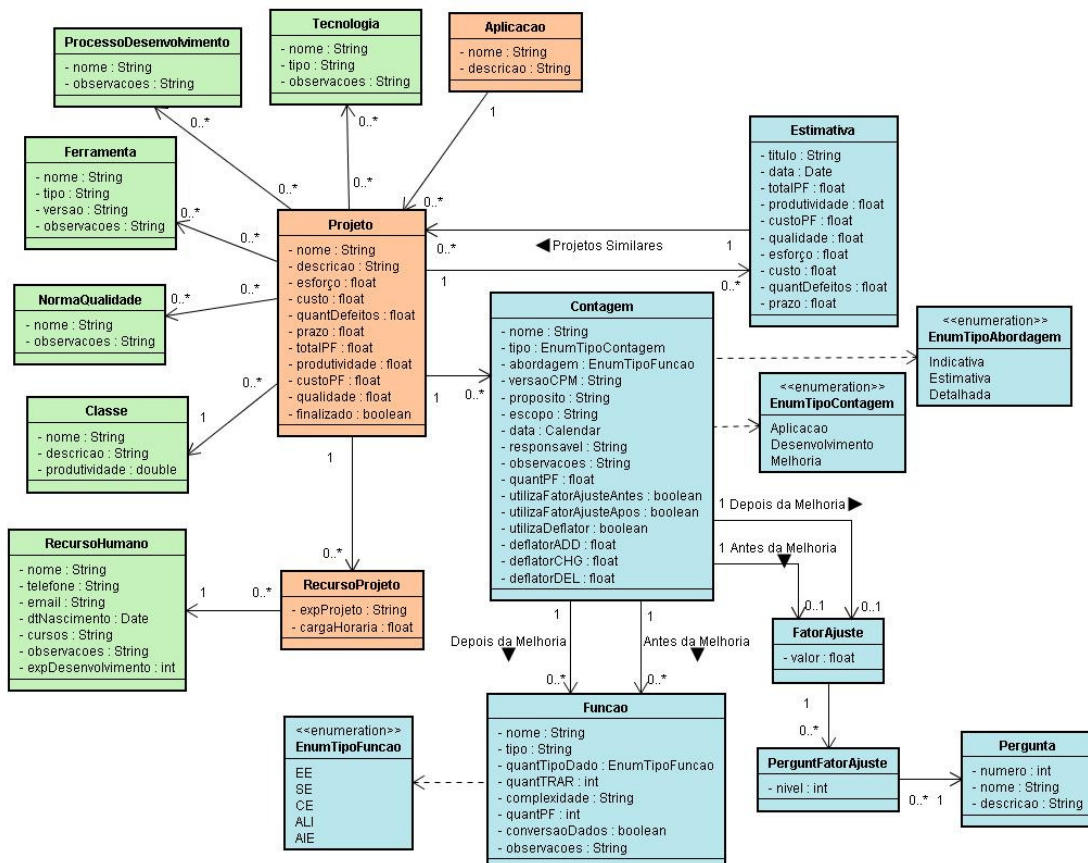


Figura 5.3 – Diagrama de Classes da Camada de Domínio do Problema

Ao atualizar o diagrama de classes do domínio do problema, na etapa de projeto, algumas modificações foram realizadas. Novos campos foram mapeados para serem atributos das classes. E para tratar alguns conjuntos de valores definidos do problema, foram criados enumerations, a fim de padronizar a utilização e agilizar o desenvolvimento.

Segue abaixo uma breve descrição dos atributos adicionados ou modificados na fase de projeto e as suas justificativas:

- **Projeto:**
 - produtividade (total de PF / esforço);
 - custoPF (custo / total de PF);
 - qualidade (quantidade de defeitos / total de PF).

- **Estimativa:**
 - esforço (total de PF / produtividade);
 - custo (custo por PF * total de PF);
 - prazo (esforço / soma do total de horas diárias dos recursos);
 - quantDefeitos (qualidade * total de PF).

Esses atributos das classes de projeto e estimativa, foram adicionados para melhorar a performance e principalmente facilitar a construção de relatórios. Isso evita que o cálculo de dessas informações de projeto e estimativa seja executado toda vez que o dado for solicitado, ou que tenha que ser feito ao gerar um relatório.

- **Contagem:**
 - tipo: Este atributo contém o tipo da Contagem, que pode ser: Desenvolvimento; Aplicação - Contagem Inicial; Melhoria; Aplicação - Após Melhoria. Cada tipo determina qual fórmula será usada para a realização do cálculo da quantidade total de PF da Contagem;
 - abordagem: Este atributo determina qual das 3 abordagens será utilizada para calcular a quantidade de pontos de função não ajustados, sendo elas: Indicativa; Estimativa; Detalhada. Cada abordagem determina a fórmula para o cálculo de pontos de função não ajustados e quais valores serão usados para cada tipo de função (ALI, AIE, EE, SE e CE).

Os três tipos de contagem e abordagem foram mapeados nos enumerations EnumTipoContagem e EnumTipoAbordagem respectivamente, para padronizar e agilizar o uso dos termos.

- quantPF: representa a quantidade de pontos de função (PF) total da contagem.

Foi adicionado para evitar que o cálculo da contagem seja feito toda vez que a quantidade de pontos de função for requisitada, melhorando assim o desempenho do sistema.

- **Função:**

- Tipo: Pode ser uma Função do Tipo Dado ou uma Função do Tipo Transação. Uma Função Tipo Dado pode ser de dois tipos: Arquivo Lógico Interno (ALI); Arquivo de Interface Externa (AIE). Uma Função Tipo Transação pode ser de três tipos: Entrada Externa (EE); Saída Externa (SE); Consulta Externa (CE). Cada tipo determina os valores para a relação entre quantidade de PF e a complexidade da função.

Esses dados foram mapeados em um enumeration (EnumTipoFuncao) para padronizar e agilizar o uso dos termos.

- quantPF: representa a quantidade de pontos de função (PF) da função, o valor pode ser obtido por meio de um calculo que depende do tipo de abordagem da contagem.

Esse atributo foi adicionado com o mesmo intuito do atributo quantPF de Contagem, ou seja, para evitar que esse cálculo tenha que ser feito toda vez que a quantidade de pontos de função de uma função for requisitada.

5.3.2 Camada de Gerência de Dados

Nesta seção são apresentadas as classes de gerência de dados do sistema. Tais classes são responsáveis pela persistência dos dados no sistema.

Mantendo a forma de guardar as informações dos softwares de estimativa analisados, a persistência será feita em arquivo e possuirá um formato padrão

conhecido como XML, que é uma linguagem de marcadores como a HTML e foi modelada para descrever dados. A extensibilidade e flexibilidade da linguagem permitem que sua descrição se aplique a uma grande variedade de aplicações heterogêneas, já que o XML possui em sua própria estrutura a descrição dos dados (MOURA, 2000).

Para auxiliar na manipulação do arquivo foi utilizado a biblioteca gratuita XStream versão 1.3, que permite a serialização de objetos para o formato XML e vice versa. O uso desta biblioteca permite implementar de forma fácil e rápida a persistência em arquivos XML, fator esse de grande importância devido ao limitado tempo para desenvolvimento deste projeto. (XSTREAM, 2008).

Segue abaixo a figura 5.4 que mostra a classe responsável por fazer os tratamentos de recuperar e salvar os dados em arquivos XML.

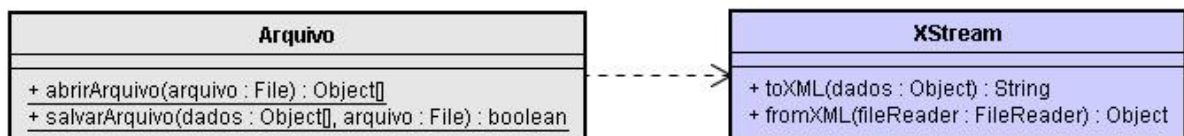


Figura 5.4 – Diagrama de Classes da Camada de Gerência de Dados

5.3.3 Camada de Gerência de Tarefas

Nesta seção são apresentadas as classes de gerência de tarefas do sistema, onde as funcionalidades identificadas são organizadas e distribuídas em classes, criando aplicações modulares, responsáveis por um conjunto coeso de funcionalidades. Tais classes são responsáveis pela execução dos casos de uso, manipulando objetos do modelo. Além disso, são responsáveis por receber requisições da camada de interface e gerenciar a navegação das telas. A figura 5.5 mostra o diagrama das classes de gerência de tarefas do sistema.

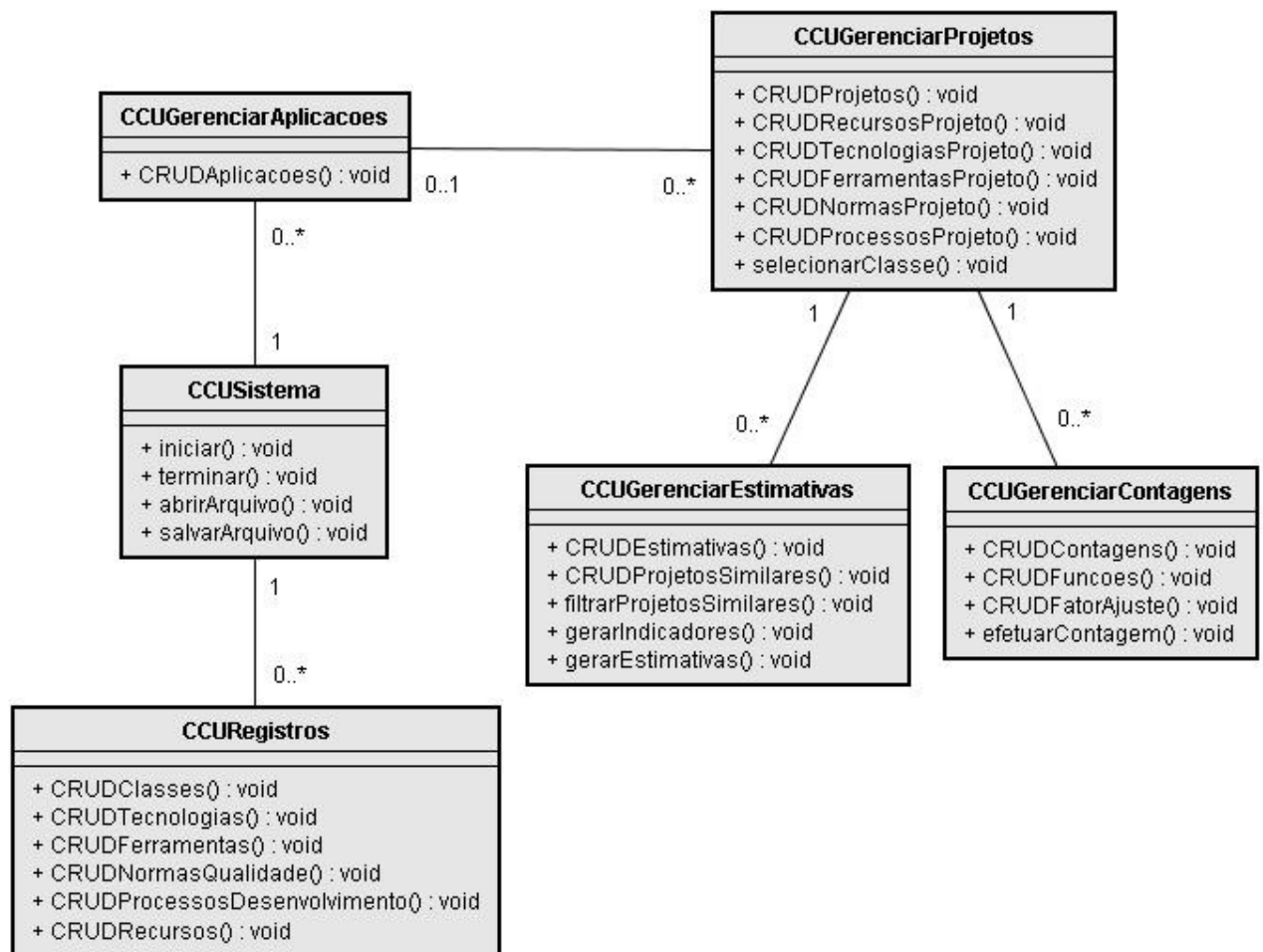


Figura 5.5 – Diagrama de Classes da Camada de Gerência de Tarefas

Segue abaixo uma breve descrição das classes e seus métodos:

- **CCUSistema:** Controlador do sistema. É onde inicia e termina o sistema.
 - iniciar: inicia o sistema através da criação do FramePrincipal;
 - terminar: termina o sistema, finalizando todas as telas abertas;
 - abrirArquivo: abre o arquivo de dados;
 - salvarArquivo: salva os dados em um arquivo XML.

- **CCURegistros:** Controlador de todas as classes de Registro. As classes de Registro foram agrupadas nesse controlador pelo fato de serem parecidas e todas possuírem um mesmo propósito.
 - CRUDClasses: Incluir, alterar, consultar e excluir classes no sistema;
 - CRUDTecnologias: Incluir, alterar, consultar e excluir tecnologias no sistema;

- CRUDFerramentas: Incluir, alterar, consultar e excluir ferramentas no sistema;
 - CRUDNormasQualidade: Incluir, alterar, consultar e excluir normas de qualidade no sistema;
 - CRUDProcessosDesenvolvimento: Incluir, alterar, consultar e excluir processos de desenvolvimento no sistema;
 - CRUDRecursos: Incluir, alterar, consultar e excluir recursos no sistema.
- **CCUGerenciarAplicacoes:** Controlador das aplicações.
 - CRUDAplicacoes: Incluir, alterar, consultar e excluir aplicações no sistema.
- **CCUGerenciarProjetos:** Controlador dos projetos. Esse controlador tem o intuito de controlar a inserção, alteração, consulta e remoção de projetos e suas características.
 - CRUDProjetos: Incluir, alterar, consultar e excluir projetos no sistema;
 - CRUDRecursosProjeto: Incluir, alterar, consultar e excluir recursos utilizados no projeto;
 - CRUDTecnologiasProjeto: Incluir, alterar, consultar e excluir tecnologias utilizadas no projeto;
 - CRUDFerramentasProjeto: Incluir, alterar, consultar e excluir ferramentas utilizadas no projeto;
 - CRUDNormasQualidade: Incluir, alterar, consultar e excluir normas de qualidade utilizadas no projeto;
 - CRUDProcessosDesenvolvimento: Incluir, alterar, consultar e excluir processos de desenvolvimento utilizados no projeto;
 - selecionarClasse: Selecionar a classe do projeto.
- **CCUGerenciarContagens:** Controlador das contagens. É responsável por controlar todas as operações relativas à criação, alteração, consulta e remoção de contagens, funções e fator de ajuste.
 - CRUDContagens: Incluir, alterar, consultar e excluir contagens no projeto;
 - CRUDFuncoes: Incluir, alterar, consultar e excluir funções na contagem;
 - CRUDFatorAjuste: Incluir, alterar, consultar e excluir o fator de ajuste da contagem;

- efetuarContagem: Efetuar o cálculo da contagem, de acordo com o tipo e abordagem .
- **CCUGerenciarEstimativas:** Controlador das estimativas. É responsável por controlar todas as operações relativas à criação, alteração, consulta e remoção de estimativas. Também é responsável por controlar os projetos similares utilizados nas estimativas.
 - CRUDEstimativas: Incluir, alterar, consultar e excluir estimativas no projeto;
 - CRUDProjetosSimilares: Incluir, alterar, consultar e excluir projetos similares na estimativa;
 - filtrarProjetosSimilares: Filtrar projetos similares através das características dos projetos;
 - gerarIndicadores: Gerar indicadores de produtividade, custo/PF e qualidade;
 - gerarEstimativas: Gerar estimativas de esforço, prazo, custo e quantidade de defeitos.

Dessa forma a camada de gerência de tarefas controla as requisições do usuário e executa os casos de uso, agrupando o controle de casos de uso similares. Assim, melhora-se a facilidade de reuso do código.

5.3.4 Camada de Interface

Nesta seção são apresentadas as classes e arquivos de interface com o usuário, considerando todos os elementos necessários à camada de visão (JFrame, JPanel, JInternalFrame, etc.).

Para a construção da camada de interface foi criada uma tela principal (FramePrincipal), onde todas as outras telas são internas (InternalFrame) à principal, conforme a figura 5.6. Dessa forma é possível abrir mais de uma tela ao mesmo tempo, podendo fazer comparações entre projetos, contagens, estimativas, etc.

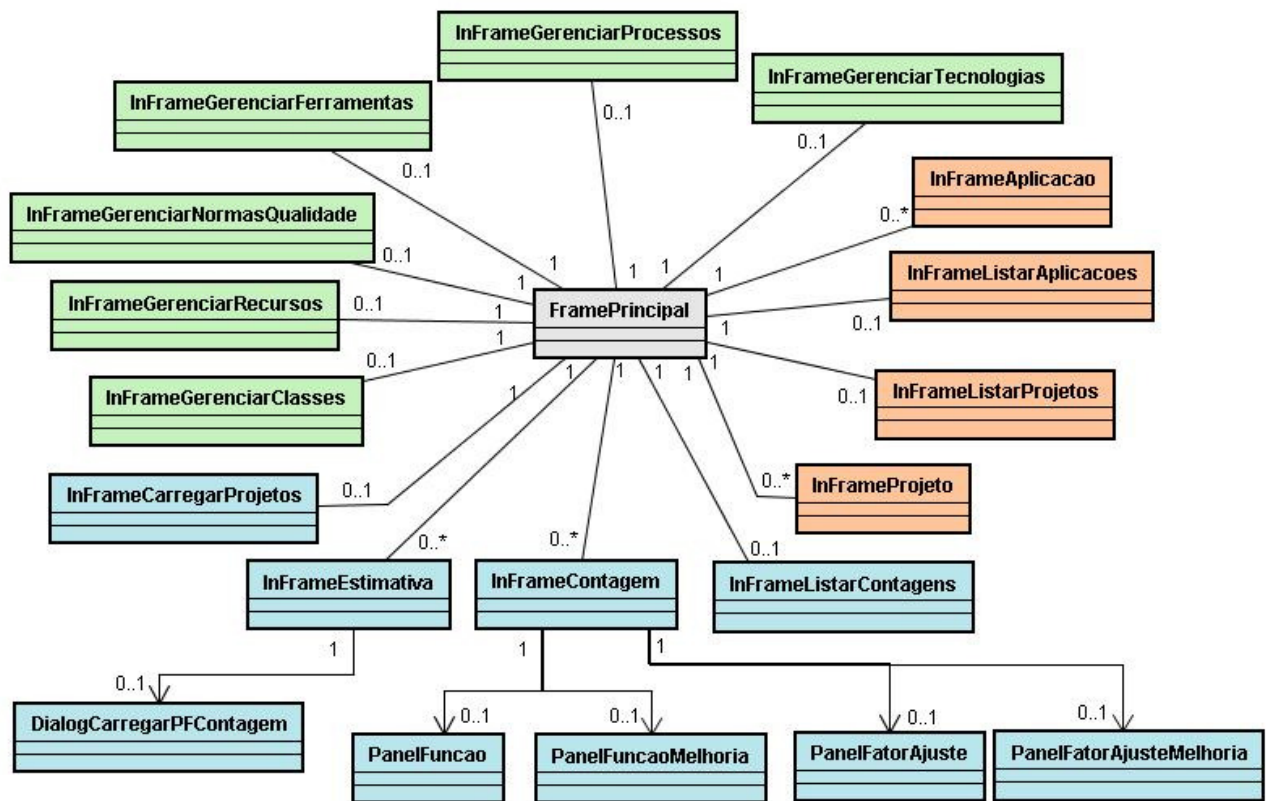


Figura 5.6 – Diagrama de Classes da Camada de Interface

A abordagem utilizada agiliza o processo de estimativas e comparações, fator de suma importância para a utilização do sistema por gerentes de projetos, tendo em vista que não se pode gastar muito tempo com a realização de estimativas. As telas do sistema são apresentadas no próximo capítulo.

5.4 Padrões de Projeto

Também na etapa de projeto, é feita uma verificação levando em conta todos os elementos já definidos e projetados, para a utilização de padrões de projeto. Segundo Freeman (2007) os padrões de projeto são lições aprendidas por aqueles que se depararam com os mesmos problemas de desenvolvimento de software. Os padrões de projeto permitem que sejam aproveitadas as melhores práticas e a experiência de outros.

Neste trabalho será utilizado um padrão de projeto muito utilizado e de grande aplicabilidade chamado Observer. Este padrão permite que objetos interessados

sejam avisados da mudança de estado ou outros eventos ocorridos num outro objeto. Com isso, ele será usado com o principal intuito de atualizar dados apresentados na interface sempre que determinados objetos do modelo sofram alguma alteração. Sua implementação pode ser feita de várias formas, mas para este trabalho será utilizando a própria implementação do java (interface Observer e classe Observable).

5.5 Conclusões do capítulo

Concluída essa importante etapa no ciclo de vida, onde foram definidas e explicitadas as tecnologias e a arquitetura a ser seguida para a implementação, o próximo passo é consolidar todas as informações levantadas e projetadas em um protótipo, como será visto no capítulo seguinte.

Capítulo 6

Implementação e Testes

De posse das informações necessárias para a implementação, descritas nos capítulos anteriores, um protótipo foi construído de forma a contemplar a maior parte dos requisitos funcionais.

Como forma de teste e divisão de tarefas para todo o processo de desenvolvimento do protótipo foi utilizada a técnica de revisão pelos pares. Este método consiste na verificação de um artefato que é examinado por qualquer integrante da equipe do projeto, exceto o autor do artefato, com o propósito de detectar defeitos (SOMMERVILLE, 2003).

Neste capítulo serão apresentadas as principais telas do sistema de forma seqüencial, seguindo como base o fluxo do diagrama de atividades descrito no capítulo 04. Além disso, será apresentada uma breve explicação das funcionalidades das telas e os testes realizados.

6.1 Implementação do Protótipo

Como já foi dito no capítulo anterior, o principal objetivo deste trabalho foi implementar o máximo de requisitos funcionais possíveis, não tendo grande preocupação com os requisitos não funcionais, tais como: navegabilidade, usabilidade, etc. Além disso, os requisitos técnicos foram escolhidos tendo em vista o pouco tempo para desenvolvimento do protótipo e o melhor conhecimento técnico em tecnologias Desktop dos desenvolvedores deste projeto no período estipulado para implementação.

Seguindo o que foi proposto na metodologia deste trabalho, o protótipo foi desenvolvido em dois ciclos iterativos. No primeiro ciclo foi implementado o necessário para se efetuar contagens de pontos de função de diferentes tipos e

abordagens. No segundo ciclo foram complementadas as demais funcionalidades, incluindo a parte de estimativa e fazendo os ajustes necessários. Tal abordagem facilitou a melhoria do protótipo, pois desde o fim do primeiro ciclo já era possível trabalhar em cima de um protótipo estruturado e funcional.

Dentre os requisitos listados no capítulo 03, apenas os três relatórios não foram implementados. Mas, isso não significa que pelo protótipo não seja possível fazer comparações de projetos ou contagens, já que em várias telas é apresentada uma listagem com os dados gerais, além de ser possível abrir várias telas ao mesmo tempo, facilitando assim as comparações.

Seguindo o fluxo de navegação, são apresentadas as telas do sistema na seguinte ordem: tela inicial, telas de registro, telas de aplicação, telas de projeto, telas de contagens, telas de estimativas.

6.1.1 Tela Inicial

Uma das características desejadas para o sistema era a criação de uma ferramenta livre e multiplataforma. Isso foi possível através da implementação na linguagem Java. Como resultado é possível ver o sistema executando tanto no Windows quanto no Linux. As figuras 6.1 e 6.2 a seguir mostram a tela inicial com a listagem das aplicações nas duas plataformas.

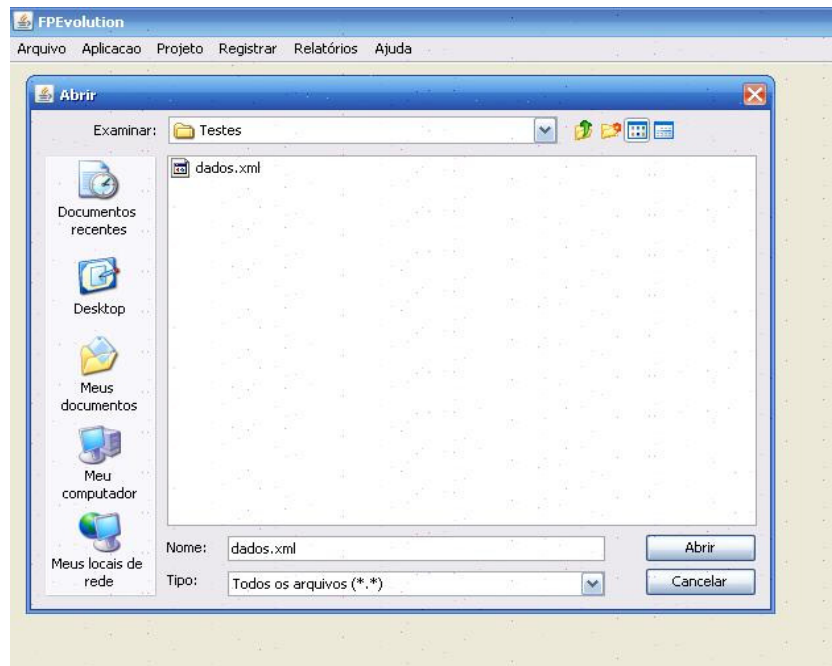


Figura 6.1 – Tela Principal (FramePrincipal) ao abrir arquivo no Windows

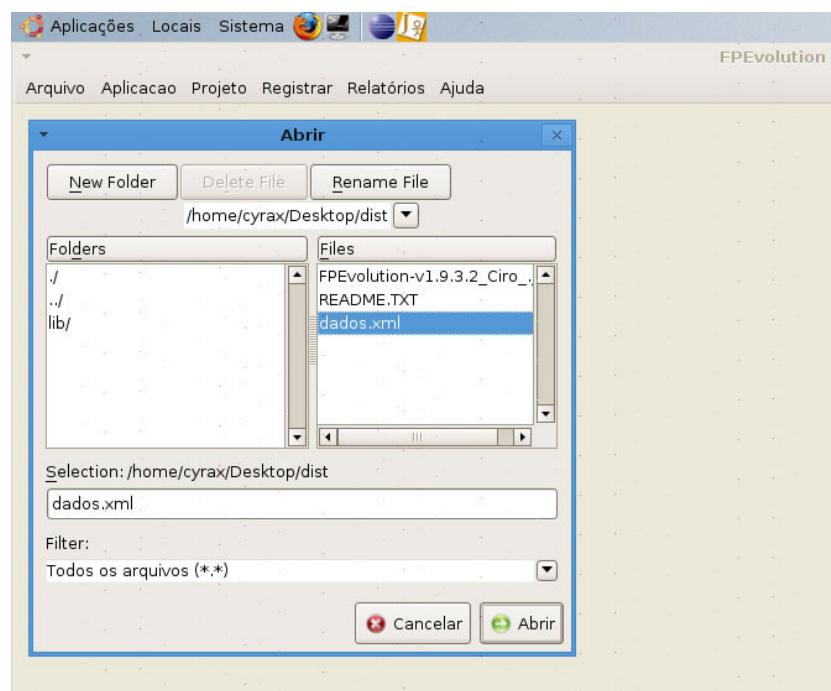


Figura 6.2 - Tela Principal (FramePrincipal) ao abrir arquivo no Linux

6.1.2 Telas de Registro

Com o intuito de abranger alguns itens do questionário elaborado pelo ISBSG para submeter projetos a sua base histórica, foram feitas algumas telas de registro. Tais

registros serão utilizados como características dos projetos criados. As telas de cadastros de apoio encontram-se no menu registrar, e são apresentadas a seguir:

- Classe, que tem o intuito de agrupar projetos com características semelhantes, como, por exemplo, Sistema Operacional, Sistema de Contabilidade, etc.;

Nome	Quantidade de Projetos	Produtividade
Administrativo	3	1
Sistema Operacional	1	1
Contabilidade	2	1
Automação	0	1
Controle de Estoque	0	1

Nome: (*)

Descrição:

Novo Salvar Excluir

Figura 6.3 - Registrar Classes (InFrameGerenciarClasses)

- Tecnologias, que consistem nas tecnologias a serem utilizadas nos projetos, como Banco de Dados, Linguagem de Programação, etc.;

Nome	Tipo
Java	Linguagem de Programação
FireBird	Banco de Dados
Postgre	Banco de Dados
MySQL	Banco de Dados
C#	Linguagem de Programação

Tipo: (*)

Nome: (*)

Observações:

Novo Salvar Excluir

Figura 6.4 - Registrar Tecnologias (InFrameGerenciarTecnologias)

- Ferramentas, que consistem nas ferramentas a serem utilizadas nos projetos, tais como SGBD, Ambiente de Desenvolvimento, etc.;

Processos de Desenvolvimento

Nome

XP

RUP

Espiral

Linear

Nome: (*)

Observações:

Novo Salvar Excluir

Figura 6.5 - Registrar Processos de Desenvolvimento (InFrameGerenciarProcessos)

- Processos de Desenvolvimento, que consistem nos processos de desenvolvimento a serem utilizados nos projetos, tais como RUP, XP, etc.;

Ferramentas

Nome	Versão	Tipo
NetBeans	6.1	Ambiente de Desenvolvimento
IBM Expert		SGBD
MS Project	2007	Gerenciamento de Projetos
Hibernate	3.3	Framework
Eclipse	3.3	Ambiente de Desenvolvimento
Visual Studio	2008	Ambiente de Desenvolvimento
PyArdoin		SGBD

Tipo: ▼ (*)

Nome: (*)

Versão:

Observações:

Novo Salvar Excluir

Figura 6.6 - Registrar Ferramentas (InFrameGerenciarFerramentas)

- Normas de Qualidade, que consistem nas normas de qualidade a serem utilizadas nos projetos, como CMMI, MPS-BR, etc.;

Figura 6.7 - Registrar Normas de Qualidade (InFrameGerenciarNormasQualidade)

- Recursos Humanos a serem utilizados nos projetos, explicitando entre outros dados a sua experiência em desenvolvimento de software. .

Nome	Quantidade de Projetos	Exp. em Desenvolvimento
Ciro Gomes		4 < 3 anos
William Bonner		3 < 3 anos
Vanessa Camargo		3 3 - 9 anos
Elton John		3 3 - 9 anos
Chico César		2 > 9 anos

Figura 6.8 - Registrar Recursos (InFrameGerenciarRecursos)

6.1.3 Telas de Aplicação

Para iniciar o cadastro de projetos é preciso antes criar a aplicação a ser desenvolvida. Inicia-se a navegação através da tela de listagem das aplicações, selecionando a aplicação desejada ou criando uma nova. Ao abrir a tela de aplicação são apresentados os dados que devem ser inseridos e a inclusão de projetos, conforme figuras 6.9 e 6.10.

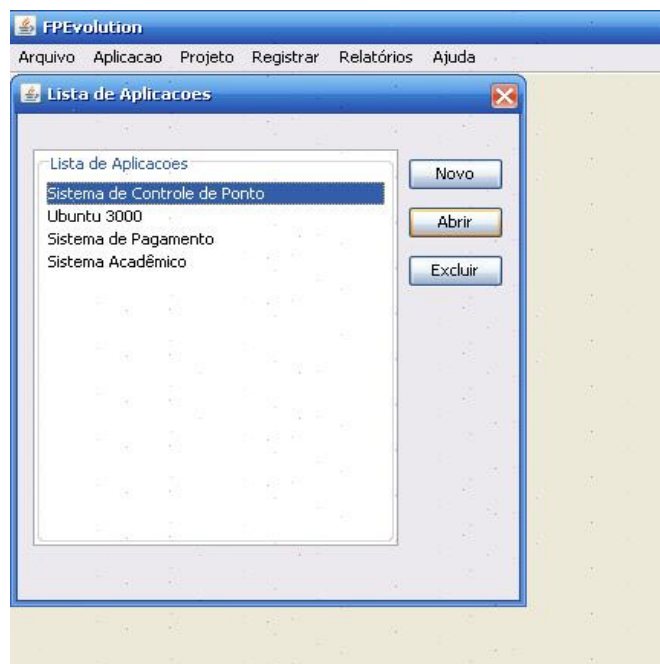


Figura 6.9 – Listar Aplicações (InFrameListarAplicacoes)

Na tela de aplicação também são apresentados os dados de esforço, produtividade, prazo, tamanho, qualidade e custo de todos os projetos desta aplicação, conforme mostrado na figura 6.10.

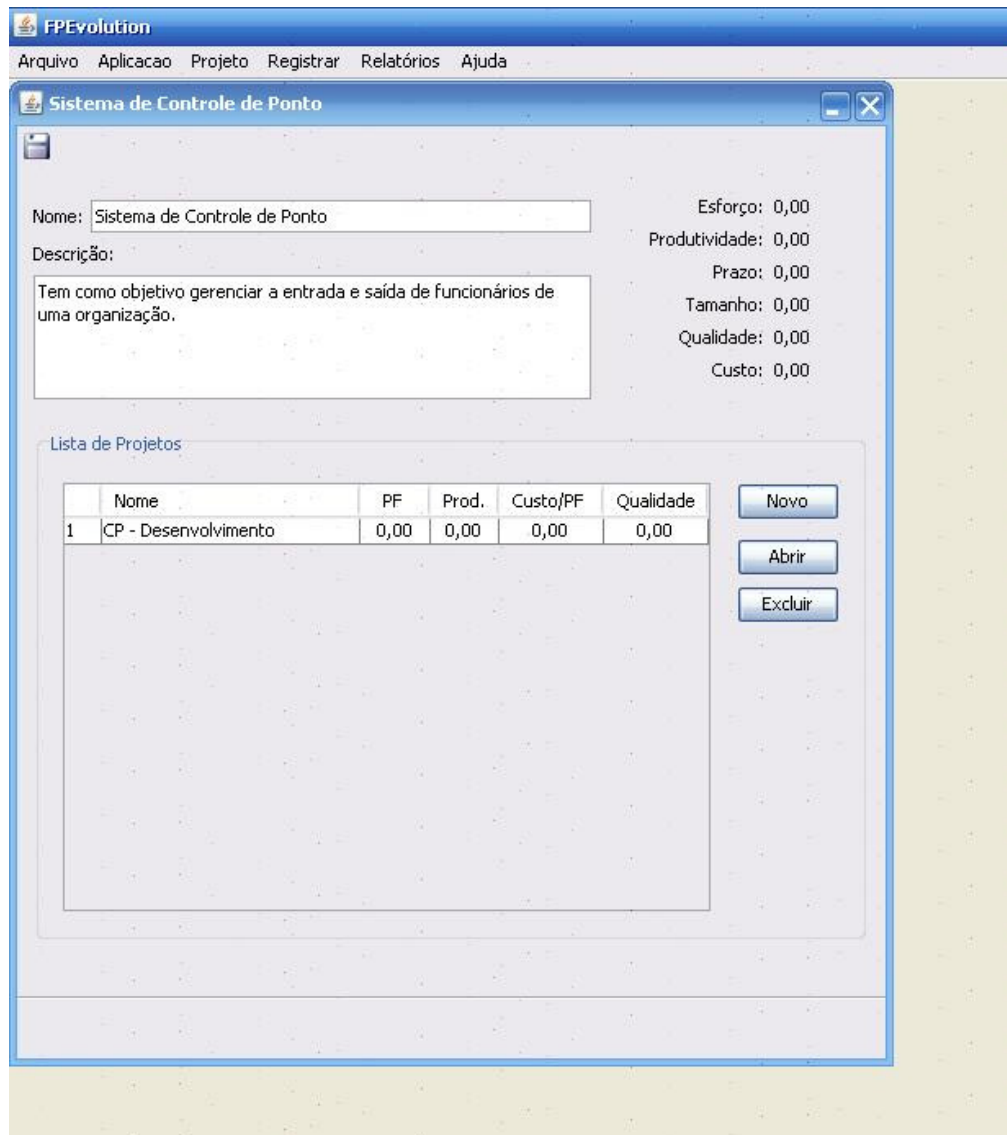


Figura 6.10 – Tela de Aplicação (InFrameAplicacao)

6.1.4 Telas de Projeto

Após a aplicação ser criada é possível criar vários projetos, sendo que eles podem ser criados de duas formas: a primeira através da tela de aplicação a outra utilizando a listagem de projetos, conforme figura 6.11.



Figura 6.11 – Listagem de Projetos (InFrameListarProjetos)

Com o projeto criado o usuário pode fazer várias contagens e várias estimativas, selecionando a aba respectiva na tela de projetos. Essa tela também possui as características gerais do projeto que podem ser selecionadas, buscando os valores dos registros feitos no sistema quanto às tecnologias, ferramentas, classe, processos de desenvolvimento, normas de qualidade e recursos. Para representar que um projeto foi finalizado é preciso selecionar a opção “Projeto finalizado” e informar os valores reais gastos no projeto.

Na figura 6.12 é apresentada a tela de Projeto do sistema, com a aba de dados gerais selecionada. As demais abas serão apresentadas ao falar das telas de contagens e estimativas, respectivamente.

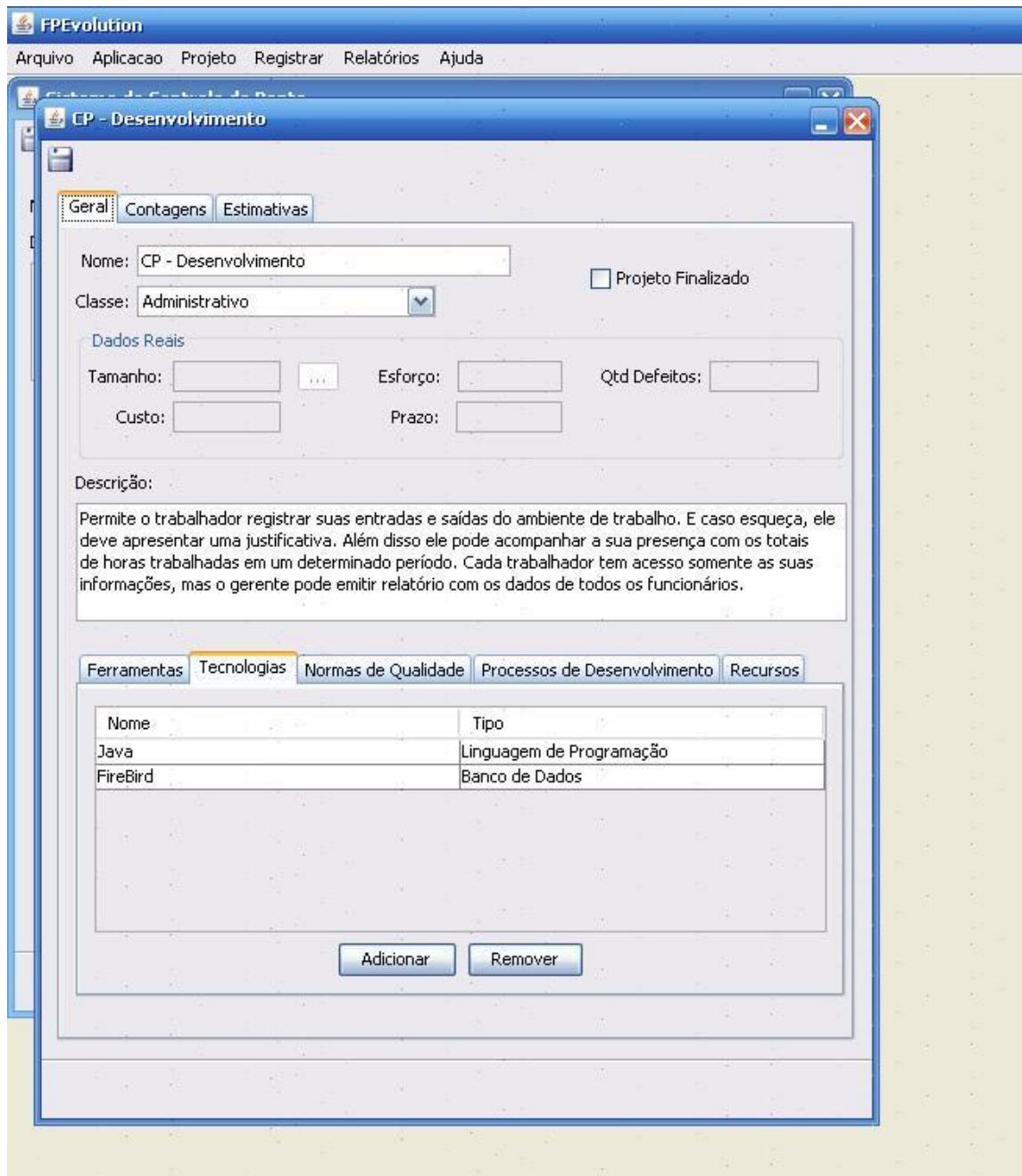


Figura 6.12 – Listagem de Projetos (InFrameListarProjetos)

6.1.5 Telas de Contagens

Com o objetivo de determinar o tamanho do projeto, podem ser realizadas várias contagens de pontos de função. Estas contagens são apresentadas na aba "Contagens" da tela de projeto, exibindo informações gerais, como podem ser vistas na figura 6.13.



Figura 6.13 - Aba de listagem de contagens

Em uma nova contagem o usuário deve determinar entre os campos de entrada qual será o tipo da contagem. Dependendo deste tipo, as abas de "Funções" e "Fator de Ajuste" serão diferentes, pois determinados tipos dependem de informações diferenciadas. Estas possibilidades serão mostradas a seguir.

Nome (*): Contagem 01 Data: 01/01/2008

Tipo (*): Desenvolvimento Aplicação Melhoria Responsável: Ciro

Abordagem (*): Indicativa Estimativa Detalhada Versão CPM: 4.2

Propósito:

Escopo:

Observações:

Deflatores

Utilizar deflatores

ADD: 1.0 CHG: 1.0 DEL: 1.0 -Calcular

Funções Adicionadas: 77,9 Funções Alteradas: 0 Funções Removidas: 0

Total de Pontos de Função: 77.9

Figura 6.14 - Dados gerais da contagem (InFrameContagem)

Pelo fato das telas nas contagens do tipo Desenvolvimento ou de Aplicação, possuírem sua apresentação bem próxima, elas são citadas juntas, já que a única diferença é que para as contagens do tipo aplicação a listagem de funções não possui a coluna de CFP.



Figura 6.15 - Listagem das funções (PanelFuncao)

A funcionalidade de carregar funções de outra contagem (botão "Carregar Funções") é um recurso para melhorar a usabilidade no sistema. Com isso é possível copiar todas as funções de uma outra contagem dentro de qualquer projeto da própria aplicação, como mostra a figura 6.16.



Figura 6.16 - Carregar funções de outra contagem (InFrameListarContagens)

Já para as contagens do tipo Melhoria são usadas duas listas, tanto para as funções quanto para o fator de ajuste, sendo uma lista para os dados de antes da melhoria e outra para os de depois da melhoria, conforme figura 6.17.

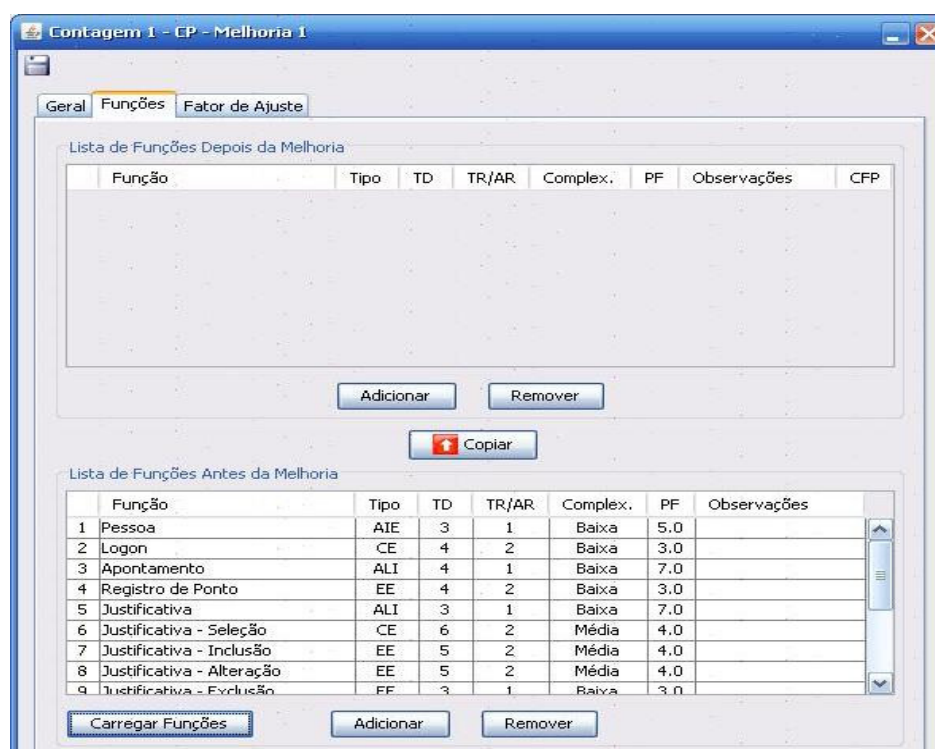


Figura 6.17 - Funções da Contagem de Melhoria (PanelFuncaoMelhoria)

Outra funcionalidade importante que foi implementada é a utilização de um deflator para contagens do tipo melhoria. Tal recurso pode ser usado como forma de calibrar o total de pontos de função das funções adicionadas, alteradas e deletadas. Essa funcionalidade encontra-se na tela de "Dados Gerais" da contagem, mas só pode ser utilizada caso seja selecionada uma contagem do tipo melhoria.

6.1.6 Telas de Estimativas

Para cada projeto o usuário pode fazer varias estimativas, que serão listadas na aba "Estimativas" na tela de projeto, apresentado os dados de tamanho (PF), produtividade (Prod.), esforço (E), prazo (P), custo (C) e qualidade (Q), como pode ser visto na figura 6.18.

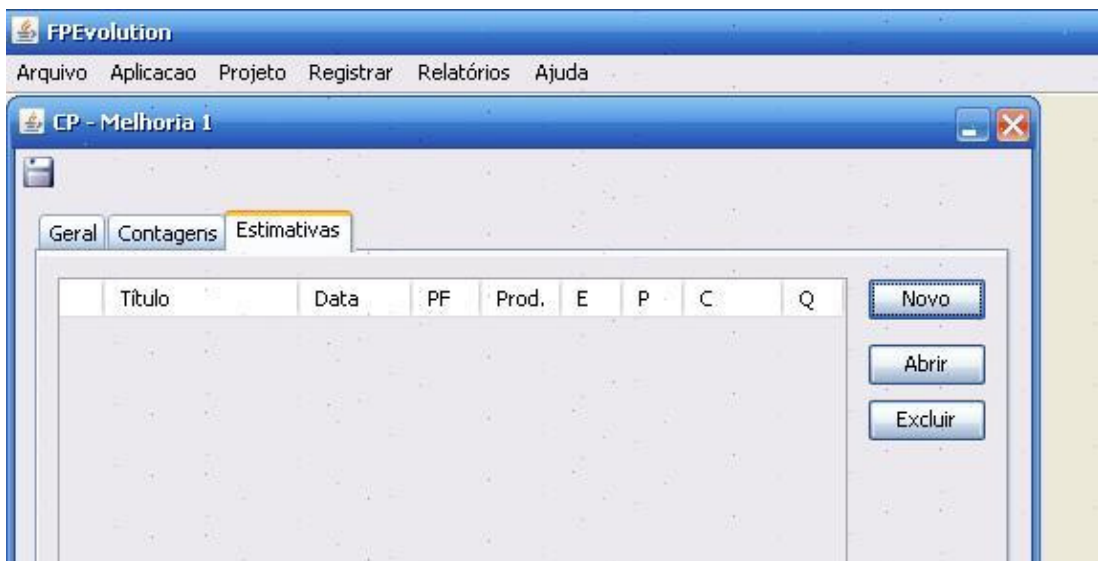


Figura 6.18 - Aba de listagem de estimativas

Ao criar uma nova estimativa o usuário pode utilizar o recurso de filtrar os projetos similares, por exemplo, listando todos os projetos que são da classe "Administrativo" e que foram feitos na linguagem "Java". Listado os projetos similares, o usuário ainda pode adicionar ou remover qualquer projeto, permitindo ampla flexibilidade para escolher os que serão usados para gerar os indicadores, conforme visto na figura 6.19.

Estimativa

Título: Data:

Filtrar Projetos Similares

Processos de Desenvolvimento | Normas de Qualidade
Tecnologias | Ferramentas

Nome	Tipo
Java	Linguagem de Programação

Equipe
Quantidade de Pessoas: a

Classe

Projetos Similares

Nome	Tamanho	Produtividade	Esforço	Prazo	Custo	Defeitos
SA - Melhoria	101,50	0,51	200,00	25,00	2200,00	15.0
SAC - Desenvolvimento	87,00	0,48	180,00	10,00	1900,00	93.0

Indicadores
Tamanho: Custo/PF:
Produtividade: Qualidade:

Estimativas
Esforço: Custo:
Qtd Defeitos: Prazo:

Figura 6.19 - Filtragem de projetos similares (InFrameEstimativa)

Os indicadores por sua vez podem ser gerados ao clicar no botão "Gerar Indicadores", fazendo com que o sistema calcule-os através da média dos indicadores de projetos similares e preencha os campos de produtividade, custo/PF e qualidade. O usuário também tem a opção de editar os indicadores, fazendo os ajustes caso desejado.

Gerar Indicadores

Indicadores

Tamanho:	77,9		Custo/PF:	21,76
Produtividade:	0,50		Qualidade:	0,61

Figura 6.20 - Gerar indicadores

Uma vez que os indicadores estão preenchidos é possível gerar as estimativas clicando no botão “Gerar Estimativas”, sendo assim finalizado o processo de estimativa.

Gerar Estimativas

Estimativas

Esforço:	155,80	Custo:	1695,10
Qtd Defeitos:	48	Prazo:	6,49

Figura 6.21 - Gerar estimativas

É importante ressaltar ainda que o sistema não faz mágica, o gerente deve ser cuidadoso ao classificar os projetos e realizar filtros coerentes com a realidade do projeto atual, somente assim serão obtidos resultados satisfatórios. Por exemplo, caso o gerente não insira algum filtro o sistema trará todos os projetos finalizados da organização, mas pouco provavelmente eles serão parecidos, pois cada um tem suas particularidades, além de alguns serem de desenvolvimento e outros de melhoria. Logo, esses fatores acarretariam em uma estimativa não precisa.

6.2 Conclusões do capítulo

Neste capítulo foi apresentado um protótipo da ferramenta de estimativa de projetos de software, com as funcionalidades do sistema. Lembrando que apenas os relatórios descritos na fase de requisitos não foram implementados.

Houve uma preocupação em implementar o máximo possível de requisitos funcionais, visando a utilização do sistema para realização do cálculo de estimativas em projetos de desenvolvimento de software.

Capítulo 7

Considerações Finais

A utilização de ferramentas que auxiliem no cálculo de estimativas de projetos de software é de grande utilidade para as equipes de desenvolvimento tendo em vista que agilizam o processo de estimativas, sendo esse um fator determinante para gerentes de projetos.

Além disso, a criação de uma base histórica de projetos é fundamental para obter maior precisão nas estimativas, já que a produtividade é diferente para cada organização e equipe. Dessa forma, cada organização deve possuir a sua base histórica para que possam assim obter indicadores de produtividade, custo/PF e qualidade com base nos seus projetos finalizados. Através destes indicadores, juntamente com a medição ou estimativa de tamanho, é possível gerar estimativas de esforço, prazo, custo e quantidade de defeitos que sejam mais perto da realidade.

É importante salientar que, dentre as técnicas para medição e estimativa de tamanho do software, destaca-se a APF devido ao fato de ser uma técnica consolidada, contendo inclusive um órgão regulador (IFPUG). Além disso, a técnica possui muitas vantagens para a sua utilização nas estimativas, destacando entre elas o fato dela não ser subjetiva e poder ser aplicada em todas as etapas de desenvolvimento. Por essa característica, as medições podem ser feitas por pessoas diferentes, em diferentes etapas, em que deverão ser obtidas as mesmas funções e conseqüentemente o mesmo valor, diferentemente das demais técnicas.

O foco do sistema foi apresentar uma solução eficaz, viável e produtiva para que as equipes de desenvolvimento de software possam realizar estimativas confiáveis e assim tomar decisões importantes, seja quanto aos contratos e comunicação com os clientes, seja quanto à melhoria de sua equipe.

Deve-se levar em consideração que muitos conceitos aqui apresentados foram frutos de um curso de Capacitação em Análise de Pontos de Função oferecido pela Fatto Consultoria e Sistemas aos desenvolvedores deste projeto. Tal curso foi de

fundamental importância para o desenvolvimento técnico e profissional dos desenvolvedores. De antemão é preciso agradecer a contribuição da Fatto, em especial ao Guilherme Siqueira Simões, instrutor do curso, que se colocou a dispor desde o início deste trabalho aprimorando os conceitos aqui apresentados com a sua grande experiência no assunto.

Com o intuito de aprimorar as estimativas a serem feitas poderão ser acrescentadas outras técnicas confiáveis, tais como o COCOMO II. Essa técnica, por exemplo, permitirá estimativas de prazo de forma mais precisa através de modelos paramétricos, tendo como um dos dados de entrada o tamanho em pontos de função.

Pode-se identificar como perspectiva futura a utilização de índices de similaridade para filtrar e comparar projetos similares nas estimativas. Existem técnicas específicas para isso e a sua utilização contribuirá na escolha de quais projetos serão utilizados como base para a estimativa.

Outra perspectiva futura é a transposição do sistema para a plataforma Web, utilizando Banco de Dados para a persistência. Através desta transposição diferentes organizações poderão utilizar o sistema ao mesmo tempo e compartilhar informações importantes sobre projetos finalizados e estimativas realizadas.

É possível identificar também como perspectiva futura a geração do questionário do ISBSG (Anexo B) pelo sistema. Dessa forma as empresas poderão submeter os seus projetos ao órgão e ter acesso à sua base histórica, sem perder muito tempo com a sua elaboração, fator esse de grande importância tendo em vista a quantidade de perguntas do questionário.

Por fim, ainda como perspectiva futura, é possível implementar os relatórios descritos na fase de requisitos deste trabalho e aprimorar os requisitos não funcionais do sistema, tais como navegabilidade e usabilidade.

REFERÊNCIAS BIBLIOGRÁFICAS

BFPUG. c2007. Disponível em: <<http://www.bfpug.com.br>> Acesso em 27 abr. 2008.

DEBONI, José E. Z. **Breve Introdução aos Diagramas UML** Disponível em: <<http://www.voxxel.com.br/pages/introdiauml.html>> Acesso em 06 de nov. 2008.

DIAS, Raquel. 2003. **Análise por pontos de função: uma técnica para dimensionamento de sistemas de informação.** Revista eletrônica de sistemas de informação – RESI, 3 ed., artigo 2.

FARIA JUNIOR, Jason Paulo Tavares. Comparação entre Ferramentas CASE para gerenciamento de Projeto e Métricas de Software no Curso de Sistema da Informação do UniFOA. **Cadernos UniFOA**, Volta Redonda, ano 1, n. 2, nov. 2006. Disponível em: <http://www.unifoa.edu.br/pesquisa/caderno/materias_ed2/11.html>. Acesso em: 18 maio 2008.

FALBO, Ricardo de Almeida. Projeto de Sistemas: Notas de Aula. Departamento de Informática, UFES, 2003.

FATTO Consultoria e Sistemas. c2007. Disponível em: <<http://www.fattocs.com.br>> Acesso em 26 abr. 2008.

FREEMAN, Eric; FREEMAN Elizabeth. **Use a Cabeça! Padrões de Projetos**, 2. ed. Rio de Janeiro: AltaBooks, 2007.

GALVAO, Ana Maria. c1999. **Pontos de função como ferramenta no gerenciamento de projetos de sistemas.** Disponível em: <www.bfpug.com.br/Artigos/Palestra_introdutoria_FPA.ppt> Acesso em 12 abr. 2008.

GLASS, R.L. **Facts and Fallacies of Software Engineering.** New York: Addison-Wesley, 2003.

HENRY, D. **Software Estimation: Perfect Practices Makes Perfect**. Crosstalk – The Journal of Defense Software Engineering, June 2002.

LabES. c2002. Disponível em: <<http://www.inf.ufes.br/~labes/ode>> Acesso em 18 mai. 2008.

LARMAN, Craig – **Utilizando UML e padrões: uma introdução à análise e ao projeto orientados a objetos e ao processo unificado**. 2 Ed. Porto Alegre: Bookman, 2003.

MACORATTI, José Carlos. c2007. Disponível em: <http://www.macoratti.net/eng_qvs.htm> Acesso em 22 mar. 2008.

MACORATTI, José Carlos. c2007. Disponível em: <http://www.macoratti.net/net_oocb.htm> Acesso em 02 mai. 2008.

MCT - Ministério da Ciência e Tecnologia. c2006. Disponível em: <<http://www.mct.gov.br/index.php/content/view/11084.html>> Acesso em 22 mar. 2008.

MOURA, David F. C. **XML - Extensible Markup Language** Disponível em: <<http://www.gta.ufrj.br/~mdavid/xml.htm>>. c2000. Acesso em 15 de nov. 2008.

PRESSMAN, Roger S. **Engenharia de Software**. 5. ed. Rio de Janeiro: Mc Graw Hill, 2002.

SOMMERVILLE, Ian. **Engenharia de Software**; tradução André Maurício de Andrade Ribeiro; revisão técnica Kechi Hiramã. 6ª edição. São Paulo: Pearson Addison Wesley, 2003.

SOUZA, Vitor. **Introdução à Orientação a Objetos**, Disponível em: <<http://www.inf.ufes.br/~vsouza/files/CursoOOSlides03.pdf>> Acesso em 03 de mai. 2008

THAYER, R.; DORFMAN, M. **System and Software Requirements Engineering**, IEEE Computer Society Press, 1990.

VAZQUEZ, Carlos at all. **Análise de Pontos de Função – Medição, Estimativas e Gerenciamento de Projetos de Software**. 1. ed. São Paulo: Erica, 2003.

VAZQUEZ, Carlos at all. **Análise de Pontos de Função – Medição, Estimativas e Gerenciamento de Projetos de Software**. 8. ed. São Paulo: Erica, 2008.

XSTREAM. Disponível em: <<http://xstream.codehaus.org/>> Acesso em 10 de maio 2008.

WAZLAWICK, Raul Sidnei. **Análise e Projeto de Sistemas de Informação Orientados a Objetos**. 2 ed. Rio de Janeiro: Campos, 2004.

Anexo A

Descrição dos Casos de Uso

Anexo B

Questionário do ISBSG

Anexo A

Descrição dos Casos de Uso

Caso de Uso: Gerenciar Classes

Descrição: Esse caso de uso descreve as operações do sistema para incluir, alterar, excluir classes de projetos no sistema. Uma classe é o conjunto de projetos com características semelhantes, por exemplo: Sistema Operacional, Editor de Texto, Jogos e outros.

Atores: Usuário

Pré-condições:

Curso Normal

Incluir

1. Usuário seleciona a opção "Adicionar Classe";
2. Sistema exibe o formulário que possui os campos relacionados a classes:
 - a. Nome*;
 - b. Descrição.
3. Usuário insere pelo menos os dados obrigatórios (*);
4. Usuário clica em salvar;
5. Sistema valida os dados (A1) e exibe a mensagem de sucesso.

Alterar

1. Usuário seleciona uma determinada classe;
2. Sistema exibe seus dados em um formulário. (com os mesmos campos do cenário "Incluir");
3. Usuário edita qualquer informação da classe;
4. Usuário clica em salvar;
5. Sistema valida os dados (A1) e exibe a mensagem de sucesso.

Excluir

1. Usuário seleciona uma determinada classe;
2. Usuário seleciona a opção "Excluir" (A3);
3. Sistema pede a confirmação do Usuário;
4. Usuário confirma (A2);
5. Sistema exibe a mensagem de sucesso.

Cursos Alternativos

- A1. Caso já exista alguma classe com o mesmo nome, o sistema exibe uma mensagem de erro, e não salva a operação.
- A2. Usuário não confirma operação e o sistema não efetua a exclusão.
- A3. Se existir algum projeto da classe selecionada, o sistema exibe uma mensagem de erro informando que existem projetos dessa classe, e não registra a operação.

Caso de Uso: Gerenciar Tecnologia

Descrição: Esse caso de uso descreve as operações do sistema para incluir, alterar, excluir registros de tecnologias no sistema. (Exemplos: Linguagem de Programação: JAVA; Banco de Dados: Postgre).

Atores: Usuário

Pré-condições:

Curso NormalIncluir

1. Usuário seleciona a opção "Gerenciar Tecnologia";
2. Sistema exibe as tecnologias já cadastradas no sistema, e um formulário para a inclusão de uma nova tecnologia contendo os seguintes campos:
 - a. Nome*;
 - b. Tipo*
Os tipos de tecnologia podem ser:
 1. Linguagem de Programação;
 2. Banco de Dados;
 3. Sistema Operacional;
 4. Outras;
 - c. Observações.
3. Usuário informa pelo menos os dados obrigatórios (*);
4. Usuário clica em salvar;
5. Sistema valida os dados (A1) e exibe a mensagem de sucesso.

Alterar

1. Usuário seleciona a opção "Gerenciar Tecnologia";
2. Sistema Exibe uma lista com todas as tecnologias já cadastradas no sistema e um formulário com os mesmo itens que o cenário "Incluir";
3. Usuário seleciona uma tecnologia;
4. Sistema exibe seus dados no formulário. (com os mesmos campos do cenário "Incluir");
5. Usuário edita qualquer informação da tecnologia;
6. Usuário clica em salvar;
7. Sistema valida os dados (A1) e exibe a mensagem de sucesso.

Excluir

1. Usuário seleciona a opção "Gerenciar Tecnologia";
2. Sistema Exibe uma lista com todas as tecnologias já cadastradas no sistema e um formulário com os mesmo itens que o cenário "Incluir";
3. Usuário seleciona uma tecnologia;
4. Usuário seleciona a opção "Excluir" (A3);
5. Sistema pede a confirmação do Usuário;
6. Usuário confirma (A2);
7. Sistema exibe a mensagem de sucesso.

Cursos Alternativos

- A1. Caso já exista alguma tecnologia com o mesmo nome, o sistema exibe uma mensagem de erro, e não salva a operação.
- A2. Usuário não confirma operação e o sistema não efetua a exclusão.

A3. Se existir algum projeto que possua a tecnologia solicitada, o sistema exibe uma mensagem de erro informando que existem projetos com essa tecnologia, e não registra a operação.

Caso de Uso: Gerenciar Processos de Desenvolvimento

Descrição: Esse caso de uso descreve as operações do sistema para incluir, alterar, excluir registros de processos de desenvolvimento no sistema (Exemplos de processos de desenvolvimento: RUP, XP, RAD entre outros).

Atores: Usuário

Pré-condições:

Curso NormalIncluir

1. Usuário seleciona a opção "Gerenciar Processos de Desenvolvimento";
2. Sistema exibe uma lista com todos os processos de desenvolvimento já cadastradas no sistema e um formulário para a inclusão de processos de desenvolvimento contendo os seguintes campos:
 - a. Nome*;
 - b. Observações.
3. Usuário informa pelo menos os dados obrigatórios (*);
4. Usuário clica em salvar;
5. Sistema valida os dados (A1) e exibe a mensagem de sucesso.

Alterar

1. Usuário seleciona a opção "Gerenciar Processos de Desenvolvimento";
2. Sistema exibe uma lista com todos os processos de desenvolvimento já cadastradas no sistema e um formulário com os mesmo itens que o cenário "Incluir";
3. Usuário seleciona um processo de desenvolvimento;
4. Sistema exibe seus dados em um formulário. (com os mesmos campos do cenário "Incluir");
5. Usuário edita qualquer informação do processo de desenvolvimento;
6. Usuário clica em salvar;
7. Sistema valida os dados (A1) e exibe a mensagem de sucesso.

Excluir

1. Usuário seleciona a opção "Gerenciar Processos de Desenvolvimento";
2. Sistema exibe uma lista com todos os processos de desenvolvimento já cadastradas no sistema e um formulário com os mesmo itens que o cenário "Incluir";
3. Usuário seleciona um processo de desenvolvimento;
4. Usuário seleciona a opção "Excluir" (A3);
5. O sistema pede a confirmação do Usuário;
6. Usuário confirma (A2);
7. Sistema exibe a mensagem de sucesso.

Cursos Alternativos

- A1. Caso já exista algum processo de desenvolvimento com o mesmo nome, o sistema exibe uma mensagem de erro, e não salva a operação.
- A2. Usuário não confirma operação e o sistema não efetua a exclusão.

A3. Se existir algum projeto que possua o processo de desenvolvimento solicitado, o sistema exibe uma mensagem de erro informando que existem projetos com esse processo de desenvolvimento, e não registra a operação.

Caso de Uso: Gerenciar Ferramentas

Descrição: Esse caso de uso descreve as operações do sistema para incluir, alterar e excluir as ferramentas a serem utilizadas nos projetos do sistema (Exemplos: Framework: Hibernate; IDE: Visual Studio 2008; SGBD: IBExpert).

Atores: Usuário

Pré-condições:

Curso NormalIncluir

1. Usuário Seleciona a opção "Gerenciar Ferramentas";
2. Sistema Exibe uma lista com todas as ferramentas já cadastradas no sistema e um formulário para a inclusão de uma nova ferramenta contendo os seguintes campos:
 - a. Tipo*;
Existem os seguintes tipos: Ambiente de Desenvolvimento; Framework; Ferramenta Case; SGBD; Gerenciamento de Projetos; Documentação; Outra.
 - b. Nome*;
 - c. Versão;
 - d. Observações.
3. Usuário insere pelo menos os dados obrigatórios (*);
4. Usuário clica em salvar;
5. Sistema valida os dados (A1) e exibe a mensagem de sucesso.

Alterar

1. Usuário Seleciona a opção "Gerenciar Ferramentas";
2. Sistema Exibe uma lista com todas as ferramentas já cadastradas no sistema e um formulário com os mesmo itens que o cenário "Incluir";
3. Usuário seleciona uma ferramenta;
4. Sistema exibe seus dados em um formulário. (com os mesmos campos do cenário "Incluir");
5. Usuário edita qualquer informação da ferramenta;
6. Usuário clica em salvar;
7. Sistema valida os dados (A1) e exibe a mensagem de sucesso.

Excluir

1. Usuário Seleciona a opção "Gerenciar Ferramentas";
2. Sistema Exibe uma lista com todas as ferramentas já cadastradas no sistema e um formulário com os mesmo itens que o cenário "Incluir";
3. Usuário seleciona uma determinada ferramenta;
4. Usuário seleciona a opção "Excluir" (A3);
5. Sistema pede a confirmação do Usuário;
6. O usuário confirma (A2);
7. O sistema exclui a ferramenta e exibe a mensagem de sucesso.

Cursos Alternativos

- A1. Caso já exista algum processo de desenvolvimento com o mesmo nome, o sistema exibe uma mensagem de erro, e não salva a operação.
- A2. Usuário não confirma operação e o sistema não efetua a exclusão.

A3. Caso exista algum projeto que utilizou a ferramenta ou que esteja utilizando, o sistema exibe uma mensagem de erro dizendo que não é possível excluir ferramentas utilizadas em projetos.

Caso de Uso: Gerenciar Normas de Qualidade

Descrição: Esse caso de uso descreve as operações do sistema para incluir, alterar e excluir Normas de Qualidade de projetos no sistema (Exemplos: CMMI nível 1; MPS.BR nível F).

Atores: Usuário

Pré-condições:

Curso NormalIncluir

1. Usuário seleciona a opção “Gerenciar Normas de Qualidade”;
2. Sistema exibe uma lista com todas as normas de qualidade já cadastrados no sistema e um formulário para a inclusão de uma nova norma contendo os seguintes campos:
 - a. Nome*;
 - b. Observações.
3. Usuário insere pelo menos os dados obrigatórios (*);
4. Usuário clica em salvar;
5. Sistema valida os dados (A1) e exibe a mensagem de sucesso.

Alterar

1. Usuário seleciona a opção “Gerenciar Normas de Qualidade”;
2. Sistema exibe uma lista com todas as normas de qualidade já cadastrados no sistema e um formulário com os mesmos itens que o cenário "Incluir";
3. Usuário seleciona uma norma de qualidade;
4. Sistema exibe seus dados em um formulário. (com os mesmos campos do cenário "Incluir");
5. Usuário edita qualquer informação da norma de qualidade;
6. Usuário clica em salvar;
7. Sistema valida os dados (A1) e exibe a mensagem de sucesso.

Excluir

1. Usuário seleciona a opção “Gerenciar Normas de Qualidade”;
2. Sistema exibe uma lista com todas as normas de qualidade já cadastrados no sistema e um formulário com os mesmos itens que o cenário "Incluir";
3. Usuário seleciona uma norma de qualidade;
4. Usuário seleciona a opção "Excluir" (A3);
5. O sistema pede a confirmação do Usuário;
6. Usuário confirma (A2);
7. O sistema exclui a norma de qualidade e exibe a mensagem de sucesso.

Cursos Alternativos

- A1. Caso já exista alguma norma de qualidade com o mesmo nome, o sistema exibe uma mensagem de erro, e não salva a operação.
- A2. Usuário não confirma operação e o sistema não efetua a exclusão.

A3. Caso exista algum projeto que utilizou a norma de qualidade ou que esteja utilizando, o sistema exibe uma mensagem de erro dizendo que não é possível excluir normas de qualidade já utilizadas em projetos.

Caso de Uso: Gerenciar Recursos

Descrição: Esse caso de uso descreve as operações do sistema para incluir, alterar e excluir recursos para serem utilizadas em projetos.

Atores: Usuário

Pré-condições:

Curso NormalIncluir

1. Usuário Seleciona a opção "Gerenciar Recursos";
2. Sistema Exibe uma lista com todos os recursos já cadastrados no sistema e um formulário para a inclusão de um novo recurso contendo os seguintes campos:
 - a. Nome*;
 - b. Data de Nascimento;
 - c. Experiência em Desenvolvimento;
Uma opção entre as três abaixo:
 - i. < 3 anos;
 - ii. 3-9 anos;
 - iii. > 9 anos;
 - d. Telefone;
 - e. E-mail;
 - f. Cursos;
 - g. Observações.
3. Usuário insere pelo menos os dados obrigatórios (*);
4. Usuário clica em salvar;
5. Sistema valida os dados (A1) e exibe a mensagem de sucesso.

Alterar

1. Usuário Seleciona a opção "Gerenciar Recursos";
2. Sistema Exibe uma lista com todos os recursos já cadastradas no sistema e um formulário com os mesmo itens que o cenário "Incluir";
3. Usuário seleciona um recurso;
4. Sistema exibe seus dados no formulário;
5. Usuário edita qualquer informação;
6. Usuário clica em salvar;
7. Sistema valida os dados (A1) e exibe a mensagem de sucesso.

Excluir

1. Usuário Seleciona a opção "Gerenciar Recursos";
2. Sistema Exibe uma lista com todos os recursos já cadastradas no sistema e um formulário com os mesmo itens que o cenário "Incluir";
3. Usuário seleciona um recurso;
4. Usuário seleciona a opção "Excluir" (A3);
5. Sistema pede a confirmação do Usuário;
6. O usuário confirma (A2);
7. O sistema exclui o recurso e exibe a mensagem de sucesso.

Cursos Alternativos

- A1. Caso já exista algum recurso com o mesmo nome, o sistema exibe uma mensagem de erro, e não salva a operação.
- A2. Usuário não confirma operação e o sistema não efetua a exclusão.
- A3. Caso exista algum projeto que utilizou o recurso ou que esteja utilizando, o sistema exibe uma mensagem de erro dizendo que não é possível excluir recursos utilizados em projetos.

Caso de Uso: Gerenciar Projetos

Descrição: Esse caso de uso descreve as operações do sistema para incluir, alterar, excluir projetos no sistema.

Atores: Usuário

Pré-condições:

Curso Normal

Abrir/Selecionar

1. Usuário seleciona a opção "Abrir" do menu "Projeto";
2. Sistema exibe uma janela contendo uma lista com todos os projetos da organização;
3. Usuário seleciona um projeto;
4. Usuário clica em "Abrir";
5. Sistema exibe em um formulário os dados do projeto selecionado (o mesmo do curso Incluir);

Incluir

1. Usuário seleciona a opção "Novo" do menu "Projeto";
2. Sistema exibe o formulário com os campos relacionados a projeto:
 - Características Gerais:
 - a. Nome *;
 - b. Classe;
 - c. Descrição;
 - d. Projeto Finalizado (campo usado para indicar se o projeto está finalizado);
 - Tecnologias (caso de uso "Selecionar Tecnologia");
 - Processos de Desenvolvimento (caso de uso "Selecionar Processos de Desenvolvimento");
 - Normas de Qualidade (caso de uso "Selecionar Normas de Qualidade");
 - Ferramentas (caso de uso "Selecionar Ferramentas");
 - Equipe (caso de uso "Selecionar Recurso");
 - Dados Reais (A3);
3. Usuário insere pelo menos os dados obrigatórios (*);
4. Usuário clica em salvar;
5. Sistema valida os dados (A1) e exibe a mensagem de sucesso.

Alterar

1. Usuário seleciona um projeto (cenário: Abrir/Selecionar);
2. Usuário edita qualquer informação do projeto exibida no formulário;
3. Usuário clica em salvar;
4. Sistema valida os dados (A1) e exibe a mensagem de sucesso.

Excluir

1. Usuário seleciona a opção "Abrir" do menu "Projeto";
2. Sistema exibe uma janela contendo uma lista com todos os projetos da organização;
3. Usuário seleciona um projeto;
4. Usuário seleciona a opção "Excluir";

5. Sistema pede a confirmação do Usuário;
6. Usuário confirma (A2);
7. Sistema exibe a mensagem de sucesso.

Cursos Alternativos

- A1. Caso já exista algum projeto com o mesmo nome, o sistema exibe uma mensagem de erro, e não salva a operação.
- A2. Usuário não confirma operação e o sistema não efetua a exclusão.
- A3. Caso o projeto esteja finalizado (caso de uso "Finalizar Projeto") o sistema exibe os dados reais preenchidos:
 - a. Tamanho;
 - b. Custo;
 - c. Esforço;
 - d. Prazo;
 - e. Quantidade de Defeitos;

Caso de Uso: Selecionar Tecnologia

Descrição: Esse caso de uso descreve as operações do sistema para incluir e excluir tecnologias no projeto.

Atores: Usuário

Pré-condições: Projeto selecionado

Curso Normal

Incluir Tecnologia ao Projeto

1. Usuário seleciona a opção "Tecnologias" do Projeto corrente;
2. Sistema exibe uma tela com todas as tecnologias já vinculadas ao projeto.
3. Usuário clica em "Adicionar";
4. Usuário seleciona na lista "Nome" a nova tecnologia;
5. Sistema atualiza os dados.

Excluir Tecnologia do Projeto

1. Usuário seleciona a opção "Tecnologias" do Projeto corrente;
2. Sistema exibe uma tela com todas as tecnologias já vinculadas ao projeto.
3. Usuário seleciona uma tecnologia;
4. Usuário clica em "Remover";
5. Sistema exibe uma mensagem de sucesso.

Caso de Uso: Selecionar Ferramentas

Descrição: Esse caso de uso descreve as operações do sistema para incluir e excluir ferramentas no projeto.

Atores: Usuário

Pré-condições: Projeto selecionado

Curso Normal

Incluir Ferramenta ao Projeto

1. Usuário seleciona a opção "Ferramentas" do Projeto corrente;
2. Sistema exibe uma tela com todas as ferramentas já vinculadas ao projeto.
3. Usuário clica em "Adicionar";
4. Usuário seleciona na lista "Nome" a nova ferramenta;
5. Sistema atualiza os dados.

Excluir Ferramenta do Projeto

1. Usuário seleciona a opção "Ferramentas" do Projeto corrente;
2. Sistema exibe uma tela com todas as ferramentas já vinculadas ao projeto.
3. Usuário seleciona uma ferramenta;
4. Usuário clica em "Remover";
5. Sistema exibe uma mensagem de sucesso.

Caso de Uso: Selecionar Processos de Desenvolvimento

Descrição: Esse caso de uso descreve as operações do sistema para incluir e excluir processos de desenvolvimento no projeto.

Atores: Usuário

Pré-condições: Projeto selecionado

Curso Normal

Incluir Processo de Desenvolvimento ao Projeto

1. Usuário seleciona a opção "Processos de Desenvolvimento" do Projeto corrente;
2. Sistema exibe uma tela com todos os processos de desenvolvimento já vinculados ao projeto.
3. Usuário clica em "Adicionar";
4. Usuário seleciona na lista "Nome" o novo processo de desenvolvimento;
5. Sistema atualiza os dados.

Excluir Processo de Desenvolvimento do Projeto

1. Usuário seleciona a opção "Processos de Desenvolvimento" do Projeto corrente;
2. Sistema exibe uma tela com todos os processos de desenvolvimento já vinculados ao projeto.
3. Usuário seleciona um processo de desenvolvimento;
4. Usuário clica em "Remover";
5. Sistema exibe uma mensagem de sucesso.

Caso de Uso: Selecionar Normas de Qualidade

Descrição: Esse caso de uso descreve as operações do sistema para incluir e excluir normas de qualidade no projeto.

Atores: Usuário

Pré-condições: Projeto selecionado

Curso Normal

Incluir Normas de Qualidade ao Projeto

1. Usuário seleciona a opção "Normas de Qualidade" do Projeto corrente;
2. Sistema exibe uma tela com todas as normas de qualidade já vinculadas ao projeto.
3. Usuário clica em "Adicionar";
4. Usuário seleciona na lista "Nome" a norma de qualidade;
5. Sistema atualiza os dados.

Excluir Normas de Qualidade do Projeto

1. Usuário seleciona a opção "Normas de Qualidade" do Projeto corrente;
2. Sistema exibe uma tela com todas as normas de qualidade já vinculadas ao projeto.
3. Usuário seleciona uma norma de qualidade;
4. Usuário clica em "Remover";
5. Sistema exibe uma mensagem de sucesso.

Caso de Uso: Selecionar Recurso

Descrição: Esse caso de uso descreve as operações do sistema para incluir e excluir recursos no projeto.

Atores: Usuário

Pré-condições: Projeto selecionado

Curso Normal

Incluir Recurso

1. Usuário seleciona a opção "Equipe" do Projeto corrente;
2. Sistema exibe uma tela com todos os recursos já vinculados ao projeto.
3. Usuário clica em "Adicionar";
4. Usuário seleciona na lista "Nome" do novo recurso e preenche os campos:
 - a. Experiência (Experiência em anos no contexto do projeto);
Uma opção entre as três abaixo:
 - i. < 1 ano;
 - ii. 1-3 anos;
 - iii. > 3 anos;
 - b. Horas/Dia*;
5. Sistema atualiza os dados.

Excluir Recurso

1. Usuário seleciona a opção "Equipe" do Projeto corrente;
2. Sistema exibe uma tela com todos os recursos já vinculados ao projeto.
3. Usuário seleciona um recurso;
4. Usuário clica em "Remover";
5. Sistema exibe uma mensagem de sucesso.

Caso de Uso: Finalizar Projeto

Descrição: Esse caso de uso descreve as operações para indicar que um projeto da aplicação está finalizado, ou seja, já possui valores reais que podem ser usados em estimativas.

Atores: Usuário

Pré-condições: Projeto selecionado;

Curso NormalFinalizar Projeto

1. Usuário seleciona um projeto (caso de uso "Gerenciar Projetos" - cenário *Abrir/Selecionar*);
2. Sistema exibe o formulário com os campos dos dados reais do projeto:
 - a. Tamanho*;
 - b. Custo*;
 - c. Esforço*;
 - d. Prazo*;
 - e. Quantidade de Defeitos*;
3. Usuário insere os dados e clica em salvar;
4. Sistema valida os dados (A1) e exibe a mensagem de sucesso.

Cursos Alternativos

- A1. Caso algum item não tenha sido preenchido, o sistema exibe uma mensagem de erro, e não salva a operação.

Caso de Uso: Gerenciar Aplicações

Descrição: Esse caso de uso descreve as operações do sistema para incluir, alterar, excluir aplicações no sistema.

Atores: Usuário

Pré-condições:

Curso Normal

Abrir/Selecionar

1. Usuário seleciona a opção "Abrir" do menu "Aplicação";
2. Sistema exibe uma janela contendo uma lista com todas as aplicações cadastradas;
3. Usuário seleciona uma aplicação;
4. Usuário clica em "Abrir";
5. Sistema exibe em um formulário os dados da aplicação selecionada (os mesmos do curso *Incluir*)

Incluir

1. Usuário seleciona a opção "Novo" do menu "Aplicação";
2. Sistema exibe o formulário com os campos relacionados à aplicação:
 - Características Gerais:
 - a. Nome *;
 - b. Descrição;
 - Lista de Projetos
Lista contendo todos os projetos da aplicação selecionada, exibindo as seguintes informações sobre o projeto:
 - a. Nome*;
 - b. PF (Tamanho do projeto ao ser finalizado);
3. O sistema exibe informações gerais sobre os indicadores da aplicação:
 - a. Esforço (soma do esforço dos projetos);
 - b. Produtividade (soma do tamanho dos projetos dividido pela soma do esforço dos projetos);
 - c. Prazo (soma do prazo dos projetos);
 - d. Tamanho (soma do tamanho dos projetos);
 - e. Qualidade (soma da quantidade de defeitos dos projetos dividido pela soma do tamanho dos projetos);
 - f. Custo (soma do custo dos projetos);
4. Usuário insere pelo menos os dados obrigatórios (*);
5. Usuário clica em salvar;
6. Sistema valida os dados (A1) e exibe a mensagem de sucesso.

Alterar

1. Usuário seleciona uma aplicação (curso normal: Abrir/Selecionar);
2. Usuário edita qualquer informação da aplicação exibida no formulário;
3. Usuário clica em salvar;
4. Sistema valida os dados (A1) e exibe a mensagem de sucesso.

Excluir

1. Usuário seleciona a opção "Abrir" do menu "Aplicação";

2. Sistema exibe uma janela contendo uma lista com todas as aplicações da organização;
3. Usuário seleciona uma aplicação;
4. Usuário seleciona a opção "Excluir";
5. Sistema pede a confirmação do Usuário;
6. Usuário confirma (A2);
7. Sistema exibe a mensagem de sucesso.

Cursos Alternativos

- A1. Caso já exista alguma aplicação com o mesmo nome, o sistema exibe uma mensagem de erro, e não salva a operação.
- A2. Usuário não confirma operação e o sistema não efetua a exclusão.

Caso de Uso: Gerar Relatório de Projeto

Descrição: Descreve as operações do sistema para gerar o Relatório de Projeto, que consiste em mostrar os dados gerais de todos os projetos, como também a relação do valor estimado e real, indicando ainda o fator de crescimento.

Atores: Usuário

Pré-condições:

Curso Normal

Gerar Relatório

1. Usuário seleciona a opção “Gerar Relatório de Projeto” no menu principal;
2. Usuário informa o valor para o filtro:
 - a. Projeto;
3. O sistema exibe os seguintes dados sobre o projeto:
 - a. Nome;
 - b. Classe;
 - c. Descrição;
 - d. Lista de Tecnologias;
 - e. Lista de Ferramentas;
 - f. Lista de Processos de Desenvolvimento;
 - g. Lista de Normas de Qualidade;
 - h. Lista dos Recursos Humanos;
 - i. Projeto finalizado; (Sim/Não)
 - j. Dados Estimados: (última estimativa do projeto)
 - i. Esforço;
 - ii. Custo;
 - iii. Prazo;
 - iv. Quantidade de defeitos.
 - v. Tamanho;
 - vi. Produtividade;
 - k. Dados Reais;
 - i. Esforço;
 - ii. Custo;
 - iii. Prazo;
 - iv. Quantidade de defeitos.
 - v. Tamanho;
 - vi. Produtividade;
 - l. Fator de Crescimento; ((Tamanho Real - Tamanho Estimado)/ Tamanho Estimado);

Caso de Uso: Gerar Relatório das Classes

Descrição: Descreve as operações do sistema para gerar o Relatório das Classes, que consiste em exibir os indicadores de todos os projetos agrupados pela classe.

Atores: Usuário

Pré-condições:

Curso Normal

Gerar Relatório

1. Usuário seleciona a opção “Gerar Relatório das Classes” no menu principal;
2. O sistema exibe todos os projetos agrupados pela classe, exibindo as seguintes informações:
 - a. Classe;
 - b. Produtividade média da classe;Dados dos Projetos:
 - c. Nome;
 - d. Tamanho;
 - e. Produtividade;
 - f. Custo/PF.
 - g. Qualidade;

Caso de Uso: Gerar Relatório de Produtividade e Custo

Descrição: Descreve as operações do sistema para gerar o Relatório de Produtividade e Custo, que consiste em exibir uma relação entre as características do projeto (Ferramentas, tecnologias, Normas de Qualidade e Processos de Desenvolvimento) e a sua produtividade média e custo.

Atores: Usuário

Pré-condições:

Curso Normal

Gerar Relatório

1. Usuário seleciona a opção “Gerar Relatório de Produtividade e Custo” no menu principal;
2. O sistema exibe os seguintes dados para cada item das ferramentas, tecnologias, normas de qualidade e processos de desenvolvimento: (valor médio entre os projetos que utilizam esses itens)
 - a. Nome do Item;
 - b. Produtividade;
 - c. Custo/PF;
 - d. Qualidade;

Caso de Uso: Gerenciar Contagem

Descrição: Esse caso de uso descreve as operações do sistema para incluir, alterar e excluir contagens em projetos no sistema.

Atores: Usuário

Pré-condições: Projeto selecionado

Curso NormalListar Contagens:

1. Usuário seleciona a seção "Contagens" do projeto selecionado;
2. Sistema exibe uma lista com todas as contagens do projeto, contendo os seguintes dados:
 - a. N^o;
 - b. Nome;
 - c. Tipo;
 - d. PF;

Incluir

1. Usuário seleciona a seção "Contagens" do projeto selecionado;
2. Usuário seleciona a opção "Nova Contagem";
3. Sistema exibe o formulário para inserir os dados da contagem:
 - a. Nome *;
 - b. Tipo*;
Pode ser de 03 tipos diferentes (cada tipo possui uma fórmula diferente para o cálculo de pontos de função ajustados descrita no caso de uso "Efetuar Contagem"):
 1. Desenvolvimento;
 2. Aplicação;
 3. Melhoria.
 - c. Abordagem*;
Pode ser de 03 abordagens diferentes (cada tipo possui uma fórmula diferente para o cálculo de pontos de função não ajustados):
 1. Indicativa (caso de uso "Efetuar Cálculo da Contagem Indicativa");
 2. Estimativa (caso de uso "Efetuar Cálculo da Contagem Estimativa");
 3. Detalhada (caso de uso "Efetuar Cálculo da Contagem Detalhada").
 - d. Versão do CPM;
 - e. Propósito;
 - f. Escopo;
 - g. Data;
 - h. Responsável;
 - i. Observações.
 - j. Deflator (caso de uso "Utilizar Deflator");
 - k. Total de pontos de Função (campo calculado seguindo os passos do caso de uso "Efetuar Contagem");
4. Usuário insere pelo menos os dados obrigatórios (*);

5. Usuário clica em salvar;
6. Sistema valida os dados (A1);
7. Sistema exibe a mensagem de sucesso.

Alterar

1. Usuário seleciona uma determinada contagem;
2. Sistema exibe seus dados em um formulário. (com os mesmos campos do cenário "Inserir");
3. Usuário edita qualquer informação da contagem;
4. Usuário clica em salvar;
5. Sistema valida os dados e exibe a mensagem de sucesso.

Excluir

1. Usuário seleciona uma determinada contagem;
2. Usuário seleciona a opção "Excluir";
3. Sistema exibe a mensagem de sucesso.

Cursos Alternativos

- A1. Caso já exista alguma contagem com o mesmo nome, o sistema exibe uma mensagem de erro, e não salva a operação.

Caso de Uso: Gerenciar Funções

Descrição: Esse caso de uso descreve as operações do sistema para incluir, alterar e excluir funções nas contagens de projetos do sistema.

Atores: Usuário

Pré-condições: Contagem selecionada

Curso NormalIncluir Funções

1. Usuário seleciona a opção "Funções" da contagem corrente;
2. Sistema exibe todas as funções já cadastradas na contagem (A1);
3. Usuário seleciona a opção "Adicionar" e uma nova linha é criada na tabela, contendo os seguintes campos de entrada:
 - a. Função*;
 - b. Tipo*:
Podem ser de 05(cinco) tipos:
 - i. ALI (Arquivo Lógico Interno);
 - ii. AIE (Arquivo de Interface Externa);
 - iii. EE (Entrada Externa);
 - iv. SE (Saída Externa);
 - v. CE (Consulta Externa);
 - c. TD (Quantidade de Tipos de Dados)* (A2);
 - d. TR/AR (Quantidade de Tipos de Registros / Arquivos Referenciados)* (A2);
 - e. Observações;
 - f. CFP (A3);
4. Usuário insere pelo menos os dados obrigatórios (*);
5. Usuário clica em "Salvar";
6. Sistema valida os dados (A4);
7. Sistema exibe a complexidade e a quantidade de PF da função de acordo com os dados em anexo.
8. O sistema atualiza a quantidade total de PF da contagem (Executa o UC Efetuar Contagem).

Carregar funções de outra contagem

1. Usuário seleciona a opção "Funções" da contagem corrente;
2. Sistema exibe todas as funções já cadastradas na contagem (A1);
3. Usuário clica em "Carregar Funções";
4. Sistema exibe uma janela com as contagens já cadastradas e um filtro por projeto.
5. Usuário seleciona uma contagem e clica em "Carregar";
6. Sistema adiciona uma copia das funções da contagem selecionada na contagem corrente;

Alterar Funções

1. Usuário Seleciona a opção "Funções" da contagem corrente;
2. Sistema Exibe todas as funções já cadastradas na contagem (A1);
3. Usuário seleciona uma função;
4. Usuário edita qualquer informação da função;

5. Usuário clica em "Salvar";
6. Sistema valida os dados (A1) e exibe a mensagem de sucesso.
7. O sistema atualiza a quantidade total de PF da contagem (Executa o UC Efetuar Contagem).

Excluir Funções

1. Usuário Seleciona a opção "Funções" da contagem corrente;
2. Sistema Exibe todas as funções já cadastradas na contagem (A1);
3. Usuário seleciona uma função;
4. Usuário seleciona a opção "Excluir";
5. O sistema exclui a função e atualiza a quantidade total de PF da contagem (Executa o UC Efetuar Contagem).

Cursos Alternativos

- A1. Caso a contagem seja do tipo melhoria o sistema exibe duas tabelas com os dados já cadastrados, uma com as funções de antes da melhoria e outra com as de depois da melhoria;
- A2. Caso a contagem possua a abordagem do tipo Estimativa ou Indicativa, os campos de TD e AR/TR não são obrigatórios;
- A3. O campo CFP indica se a função é de conversão de dados, e só será apresentado para funções de contagens do tipo desenvolvimento, ou para as funções de depois da melhoria;
- A4. Caso já exista alguma função com o mesmo grupo de dados / processo elementar (nome), o sistema exibe uma mensagem de erro, e não salva a operação.

Anexos:

1. Para contagem com a abordagem detalhada

Tabelas definidas pelo IFPUG para determinação da complexidade de uma função:

- a. Tipo Dado (ALI ou AIE):

Complexidade Funcional para ALI e AIE

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

Tipo de função	Baixa	Média	Alta
Arquivo Lógico Interno (ALI)	7 PF	10 PF	15 PF
Arquivo de Interface Externa (AIE)	5 PF	7 PF	10 PF

b. Tipo Transação (EE, SE ou CE):

Complexidade Funcional para EE

Arquivos Referenciado	Tipos de Dados		
	< 5	5 - 15	> 15
< 2	Baixa	Baixa	Média
2	Baixa	Média	Alta
> 2	Média	Alta	Alta

Complexidade Funcional para SE e CE

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

Tipo de função	Baixa	Média	Alta
Entrada Externa (EE)	3 PF	4 PF	6 PF
Saída Externa (SE)	4 PF	5 PF	7 PF
Consulta Externa (CE)	3 PF	4 PF	6 PF

2. Para contagens do tipo Indicativa

Nessa abordagem as funções de transação não são consideradas. A Tabela abaixo mostra os valores definidos pela NESMA para determinação da quantidade de PF de uma função na abordagem indicativa.

Tipo de função	PF
Arquivo Lógico Interno (ALI)	35
Arquivo de Interface Externa (AIE)	15
Entrada Externa (EE)	0
Saída Externa (SE)	0
Consulta Externa (CE)	0

3. Para contagens do tipo Estimativa

Tabela com os valores definidos pela NESMA para determinação da quantidade de PF de uma função na abordagem estimativa:

Tipo de função	PF
Arquivo Lógico Interno (ALI)	4,3
Arquivo de Interface Externa (AIE)	5,4
Entrada Externa (EE)	3,8
Saída Externa (SE)	7,4
Consulta Externa (CE)	5,5

Caso de Uso: Efetuar Contagem

Descrição: Esse caso de uso descreve as operações do sistema para calcular a quantidade de pontos de função de uma contagem.

Atores: Usuário

Pré-condições: Contagem selecionada.

Curso Normal

Calcular quantidade de pontos de função

1. O sistema executa um cálculo, com base na fórmula definida pelo tipo e abordagem da Contagem selecionada;

A abordagem determina a quantidade de PF não ajustados da contagem e pode ser um dos 03 tipos a seguir:

- i. *Indicativa:* nessa abordagem a quantidade de pontos de função não ajustados da contagem é definida pelo caso de uso “Efetuar Cálculo da Contagem Indicativa”;
- ii. *Estimativa:* nessa abordagem a quantidade de pontos de função não ajustados da contagem é definida pelo caso de uso “Efetuar Cálculo da Contagem Estimativa”;
- iii. *Detalhada:* nessa abordagem a quantidade de pontos de função não ajustados da contagem é definida pelo caso de uso “Efetuar Cálculo da Contagem Detalhada”;

O tipo determina a quantidade de PF ajustados e são 03 tipos de contagem diferentes (A1):

- i. *Desenvolvimento:*

$$DFP = UFP \times VAF$$

DFP: Número de pontos de função do projeto de desenvolvimento

UFP: Número de pontos de função não-ajustados das funções disponíveis após instalação.

VAF: Valor do fator de ajuste.

- ii. *Aplicação:*

1. *Aplicação – Contagem Inicial:*

$$AFP = ADD \times VAF$$

AFP: Número de pontos de função ajustados da aplicação.

ADD: Pontos de função não-ajustados das funções instaladas.

VAF: Valor do fator de ajuste da aplicação.

2. *Aplicação – Após Melhoria:*

$$AFP = [(UFPB + ADD + CHGA) - (CHGB + DEL)] \times VAF$$

AFP: Número de pontos de função ajustados da aplicação.

UFPB: Pontos de função não-ajustados da aplicação antes do projeto de melhoria.

ADD: Pontos de função não-ajustados das funções incluídas pelo projeto de melhoria.

CHGA: Pontos de função não-ajustados das funções alteradas pelo projeto de melhoria depois de seu término.

CHGA: Pontos de função não-ajustados das funções alteradas pelo projeto de melhoria antes de seu término.

DEL: Pontos de função não-ajustados das funções excluídas pelo projeto de melhoria.

VAFA: Valor do fator de ajuste depois do projeto de melhoria.

iii. Melhoria:

$$EFP = [(ADD + CHGA) \times VAFA] + (DEL \times VAFB)$$

EFP: Número de pontos de função do projeto de melhoria.

UFPB: Pontos de função não-ajustados da aplicação antes do projeto de melhoria.

ADD: Pontos de função não-ajustados das funções incluídas pelo projeto de melhoria.

CHGA: Pontos de função não-ajustados das funções modificadas. Reflete as funções depois das modificações.

VAFA: Valor do fator de ajuste depois do projeto de melhoria.

DEL: Pontos de função não-ajustados das funções excluídas pelo projeto de melhoria.

VAFA: Valor do fator de ajuste antes do projeto de melhoria.

2. O sistema atualiza a informação da quantidade de PF da contagem na tela.

Cursos Alternativos

- A1. Caso a contagem não utilize o Fator de Ajuste, o valor do Fator de Ajuste não influenciará no cálculo.

Caso de Uso: Efetuar Cálculo da Contagem Detalhada

Descrição: Esse caso de uso descreve as operações do sistema para calcular a quantidade de pontos de função de uma contagem, a partir da abordagem detalhada.

Atores: Usuário

Pré-condições: Contagem selecionada.

Curso Normal

Contagem Detalhada

1. Sistema soma a quantidade total de PF de cada tipo de função (EE, SE, CE, ALI, AIE) de acordo com sua complexidade, conforme tabela abaixo:

Contribuição das funções para Pontos de Função Não Ajustados

Tipo de função	Baixa	Média	Alta
Arquivo Lógico Interno (ALI)	7 PF	10 PF	15 PF
Arquivo de Interface Externa (AIE)	5 PF	7 PF	10 PF
Entrada Externa (EE)	3 PF	4 PF	6 PF
Saída Externa (SE)	4 PF	5 PF	7 PF
Consulta Externa (CE)	3 PF	4 PF	6 PF

- Como resultado o Sistema obtém o valor de Pontos de Função não Ajustados a partir da seguinte expressão:

$$\begin{aligned} \text{Contagem} = & \\ & (\text{quant_ALI_baixa} \times 7) + (\text{quant_ALI_media} \times 10) + \\ & (\text{quant_ALI_Alta} \times 15) + \\ & (\text{quant_AIE_baixa} \times 5) + (\text{quant_AIE_media} \times 7) + \\ & (\text{quant_AIE_Alta} \times 10) + \\ & (\text{quant_EE_baixa} \times 3) + (\text{quant_EE_media} \times 4) + (\text{quant_EE_Alta} \\ & \times 6) + \\ & (\text{quant_SE_baixa} \times 4) + (\text{quant_SE_media} \times 5) + (\text{quant_SE_Alta} \\ & \times 7) + \\ & (\text{quant_CE_baixa} \times 3) + (\text{quant_CE_media} \times 4) + (\text{quant_CE_Alta} \\ & \times 6) \end{aligned}$$

2. Sistema multiplica o valor obtido no passo anterior com o valor do Fator de Ajuste (Caso de Uso Registrar Fator de Ajuste), e assim obtém a quantidade de Pontos de Função Ajustados (A1).

Caso de Uso: Efetuar Cálculo da Contagem Indicativa

Descrição: Esse caso de uso descreve as operações do sistema para calcular a quantidade de pontos de função de uma contagem, a partir da abordagem indicativa.

Atores: Usuário

Pré-condições: Contagem selecionada.

Curso Normal

Dados do NESMA

1. Sistema soma a quantidade apenas dos tipos de função ALI e AIE da contagem corrente;
2. Sistema multiplica a quantidade de cada função pelos valores definidos pelo NESMA, conforme tabela abaixo:

Tipo de Função	Multiplicador em PF
ALI	35
AIE	15

3. Como resultado o Sistema retorna o valor da seguinte expressão:
 $Contagem = (quant_ALI \times 35) + (quant_AIE \times 15)$

Caso de Uso: Efetuar Cálculo da Contagem Estimativa

Descrição: Esse caso de uso descreve as operações do sistema para calcular a quantidade de pontos de função de uma contagem, a partir da abordagem estimativa.

Atores: Usuário

Pré-condições: Contagem selecionada.

Curso Normal

Dados do NESMA

1. Sistema soma a quantidade de cada tipo de função (EE, SE, CE, ALI, AIE) da contagem corrente;
2. Sistema multiplica a quantidade de cada função pelos valores definidos pelo NESMA, conforme tabela abaixo:

Tipo de Função	Multiplicador em PF
EE	4,3
SE	5,4
CE	3,8
ALI	7,4
AIE	5,5

3. Como resultado o Sistema retorna o valor da seguinte expressão:
$$\text{Contagem} = (\text{quant_EE} \times 4,3) + (\text{quant_SE} \times 5,4) + (\text{quant_CE} \times 3,8) + (\text{quant_ALI} \times 7,4) + (\text{quant_AIE} \times 5,5)$$

Caso de Uso: Registrar Fator de Ajuste

Descrição: Esse caso de uso descreve as operações do sistema para registrar o fator de ajuste em um contagem de pontos de função de um projeto.

Atores: Usuário

Pré-condições: Contagem selecionada

Curso Normal

Registrar 14 CGS

1. Usuário marca a opção "Utilizar Fator de Ajuste" na contagem selecionada;
2. Sistema exibe o formulário que possui as 14 Características Gerais do Sistema (CGS), contendo (A1):
 - a. Comunicação de Dados;
 - b. Processamento Distribuído;
 - c. Performance;
 - d. Configuração Altamente Utilizada;
 - e. Volume de Transações;
 - f. Entrada de Dados On-Line;
 - g. Eficiência do Usuário Final;
 - h. Atualização On-Line;
 - i. Complexidade de Processamento;
 - j. Reusabilidade;
 - k. Facilidade de Instalação;
 - l. Facilidade de Operação;
 - m. Múltiplos Locais;
 - n. Facilidade de mudanças.
3. Usuário insere o nível de influência de cada CGS, sendo 0 nenhuma influência e 5 muito influente (R1);
4. Sistema processa os dados inseridos e calcula fator de ajuste através da seguinte fórmula: $VFA = (\text{Somatório dos níveis de influência} * 0,01) + 0,65$;
5. Sistema exibe o valor do fator de ajuste;
6. Usuário clica em salvar;
7. Sistema valida os dados;
8. O sistema atualiza a quantidade total de PF da contagem (Executa o UC Efetuar Contagem).

Alterar 14 CGS

1. Usuário seleciona uma determinada contagem que tenha um fator de ajuste registrado;
2. Sistema exibe as 14 CGS e os níveis de influência registrados em um formulário. (com os mesmos campos do cenário "Incluir");
3. Usuário edita qualquer nível de influência de uma CGS;
4. Usuário clica em salvar;
5. Sistema valida os dados;
6. O sistema atualiza a quantidade total de PF da contagem (Executa o UC Efetuar Contagem).

Retirar Fator de Ajuste

1. Usuário desmarca a opção “Utilizar Fator de Ajuste” na contagem selecionada;
2. O sistema atualiza a quantidade total de PF da contagem (Executa o UC Efetuar Contagem).

Cursos Alternativos

- A1. Caso a contagem seja do tipo melhoria o sistema exibe duas tabelas com as 14 perguntas, uma com os dados de antes da melhoria e outra com os de depois da melhoria;

Restrições de Integridade

- R1. O nível de influência só pode receber valores de 0 a 5.

Caso de Uso: Utilizar Deflator

Descrição: Esse caso de uso descreve as operações para utilizar o deflator influenciando no tamanho de uma contagem de melhoria.

Atores: Usuário

Pré-condições: Contagem do tipo Melhoria selecionada

Curso Normal

Utilizar Deflator

1. Usuário marca a opção "Utilizar Deflator" da contagem selecionada;
2. Sistema exibe o formulário com os seguintes campos:
 - a. ADD;
 - b. CHG;
 - c. DEL;
3. O Usuário clica em "Calcular" (A1);
4. O Sistema calcula e exibe os valores corrigidos da quantidade de PF das funções adicionadas, alteradas e removidas:
 - a. Funções Adicionadas (quantidade de PF das funções adicionadas multiplicado pelo fator do campo ADD);
 - b. Funções Alteradas (quantidade de PF das funções alteradas multiplicado pelo fator do campo CHG);
 - c. Funções Removidas (quantidade de PF das funções removidas multiplicado pelo fator do campo DEL);

Não Utilizar Deflator

1. Usuário desmarca a opção "Utilizar Deflator" da contagem selecionada;
2. O Sistema calcula e exibe os valores sem nenhuma correção da quantidade de PF das funções adicionadas, alteradas e removidas:
 - a. Funções Adicionadas (quantidade de PF das funções adicionadas);
 - b. Funções Alteradas (quantidade de PF das funções alteradas);
 - c. Funções Removidas (quantidade de PF das funções removidas);

Cursos Alternativos

- A1. Caso o usuário tenha inserido valores inválidos o sistema exibe uma mensagem de erro e não efetua o calculo;

Caso de Uso: Gerenciar Estimativas

Descrição: Esse caso de uso descreve as operações do sistema para incluir, alterar, excluir estimativas de projetos no sistema.

Atores: Usuário

Pré-condições: Projeto Selecionado

Curso Normal

Listar Estimativas

1. Usuário seleciona a opção "Estimativas" na tela de Projeto;
2. Sistema exibe uma a lista com todas as estimativas do projeto selecionado, contendo os seguintes dados:
 - a. Nº;
 - b. Título;
 - c. Data;
 - d. Total de PF;
 - e. Produtividade (Média dos Projetos Similares);
 - f. Custo/PF (Média dos Projetos Similares);
 - g. Qualidade (Média dos Projetos Similares);
 - h. Esforço (Total de PF / Produtividade);
 - i. Prazo (Esforço / Recursos);
 - j. Custo (Total de PF x Custo/PF)
 - k. Quantidade de Defeitos (Qualidade x PF);

Incluir

1. Usuário seleciona a opção "Nova" na seção de "Estimativas" da tela de Projeto;
2. Sistema exibe o formulário que possui os campos de entrada relacionados à estimativa:
 - a. Título*;
 - b. Data*;
 - c. Total de PF*:
 - i. Para informar a quantidade total de PF o usuário clica no botão "...";
 - ii. O Sistema exibe numa janela a lista de contagens;
 - iii. Usuário seleciona uma contagem e clica em "OK";
 - iv. O Sistema atualiza o valor do campo editável "Total de PF" para o valor de PF da contagem selecionada;
 - d. Projetos Similares (Caso de uso "Filtrar Projetos Similares")*;
 - e. Indicadores (Caso de uso "Gerar Indicadores")*;
3. Usuário insere pelo menos os dados obrigatórios (*);
4. Usuário seleciona a opção Gerar Estimativas (Caso de uso "Gerar Estimativas");
5. Usuário clica em salvar;
6. Sistema valida os dados (A1) e exibe a mensagem de sucesso.

Alterar

1. Usuário seleciona uma determinada estimativa na seção de "Estimativas" da tela de Projeto;

2. Usuário seleciona a opção "Abrir";
3. Sistema Exibe em uma nova janela um formulário com os mesmo itens do cenário "Incluir";
4. Usuário edita qualquer informação da estimativa;
5. Usuário clica em salvar;
6. Sistema valida os dados (A1) e exibe a mensagem de sucesso.

Excluir

1. Usuário seleciona uma determinada estimativa na seção de "Estimativas" da tela de Projeto;
2. Usuário seleciona a opção "Excluir";
3. Sistema exibe a mensagem de sucesso.

Cursos Alternativos

- A1. Caso já exista alguma estimativa com o mesmo título, o sistema exibe uma mensagem de erro, e não salva a operação.

Caso de Uso: Filtrar Projetos Similares

Descrição: Esse caso de uso descreve as operações do sistema para filtrar projetos similares que serão utilizados para a geração de indicadores de uma estimativa.

Atores: Usuário

Pré-condições: Estimativa selecionada.

Curso Normal

Filtrar Projetos Similares

1. O sistema exibe um formulário com os seguintes campos de entrada:
 - a. Ferramentas (Caso de uso "Selecionar Ferramentas");
 - b. Tecnologias (Caso de uso "Selecionar Tecnologia");
 - c. Processos de Desenvolvimento (Caso de uso "Selecionar Processos de Desenvolvimento");
 - d. Normas de Qualidade (Caso de uso "Selecionar Normas de Qualidade");
 - e. Quantidade de Pessoas;
 - f. Classe;
2. Usuário seleciona a opção "Filtrar Projetos Similares";
3. O sistema busca todos os projetos que possuem as características inseridas pelo usuário (A1);
4. O sistema exibe os projetos encontrados na lista de Projetos Similares, contendo os seguintes dados:
 - a. Nome;
 - b. Tamanho;
 - c. Produtividade;
 - d. Esforço;
 - e. Prazo;
 - f. Custo;
 - g. Defeitos;

Adicionar Projeto Similar

1. O usuário clica em "Adicionar";
2. O sistema exibe uma janela contendo uma lista com todos os projetos;
3. O usuário seleciona um projeto e clica em carregar;
4. O sistema adiciona o projeto selecionado na lista de projetos similares;

Remover Projeto Similar

1. O usuário seleciona um projeto da lista de projetos similares;
2. O usuário clica em "Remover";
3. O sistema remove o projeto selecionado da lista;

Cursos Alternativos

- A1. Caso o usuário não tenha inserido nenhum parâmetro para o filtro o sistema retorna todos os projetos cadastrados;

Caso de Uso: Gerar Indicadores

Descrição: Esse caso de uso descreve as operações do sistema para calcular os indicadores produtividade, custo/PF e qualidade de uma estimativa.

Atores: Usuário

Pré-condições: Estimativa selecionada.

Curso NormalGerar Indicadores

1. Usuário seleciona a opção "Gerar Indicadores" no formulário de Estimativa (A1,A2);
2. O sistema executa um cálculo, com base na média dos indicadores de projetos similares (carregados a partir do UC "Filtrar Projetos Similares");
3. O sistema preenche os seguintes campos editáveis (A2):
 - a. Produtividade;
 - b. Custo/PF;
 - c. Qualidade.

Cursos Alternativos

- A1. Caso não existam projetos similares o sistema exibe uma mensagem de erro avisando que não é possível gerar indicadores, pois não há projetos similares.
- A2. Usuário insere os indicadores manualmente, sem executar o cálculo pelo sistema.

Caso de Uso: Gerar Estimativas

Descrição: Esse caso de uso descreve as operações do sistema para calcular as estimativas de esforço, custo, prazo e quantidade de defeitos de um projeto.

Atores: Usuário

Pré-condições: Estimativa selecionada.

Curso NormalGerar Estimativas

1. Usuário seleciona a opção “Gerar Estimativas” no formulário de Estimativa (A1);
2. O sistema executa um cálculo, com base nos indicadores (Caso de Uso “Gerar Indicadores”) e no tamanho utilizado para a estimativa;
3. O sistema preenche os seguintes campos:
 - a. Esforço (total de PF / produtividade);
 - b. Custo (custo por PF * total de PF);
 - c. Prazo (esforço / soma do total de horas diárias dos recursos);
 - d. Quantidade de defeitos (qualidade * total de PF).

Cursos Alternativos

- A1. Caso os indicadores não estejam preenchidos o sistema exibe uma mensagem de erro avisando que não é possível gerar estimativas sem o preenchimento dos indicadores.

**International Software Benchmarking
Standards Group**



**DATA COLLECTION
QUESTIONNAIRE
NEW DEVELOPMENT,
REDEVELOPMENT OR
ENHANCEMENT
Sized Using
IFPUG or NESMA
FUNCTION POINTS**

Version 5.10

Table of Contents

Introduction.....	3
The Project Submission Process	5
A. Submitter Information.....	6
B. Project Process.....	7
Project Principle Characteristics	7
Planning	9
Specification	11
Design	12
Build or Programming	13
Test.....	14
Implementation or Installation.....	15
Project Management and Monitoring	16
C. Technology	17
D. People and Work Effort.....	20
Development Team.....	20
Customers and End Users	22
IT Operations	24
Work Effort Validation	25
E. Product	27
F. IFPUG or NESMA Project Function Points	29
New Development or Redevelopment Software Size	29
Enhancement Software Size	30
IFPUG or NESMA Counting Style.....	32
Context of the Function Point Count	33
Experience of the Function Point Counter	35
G. Project Completion	36
User Satisfaction Survey	37
Project Costs	38
Cost Validation	39

Introduction

The International Software Benchmarking Standards Group

The International Software Benchmarking Standards Group, (ISBSG), is a not-for-profit organisation.

The Group's members are the Software Metrics Associations of many countries.

The ISBSG Software Project Repository exists:

- ❖ to provide software development practitioners with industry output standards against which they may compare their aggregated or individual projects, and
- ❖ to provide real data of international software development that can be analysed to help improve the management of IT resources by both business and government

Benefits of Your Involvement

Contributing to the ISBSG Project Repository provides you with an opportunity to see how your projects compare with the industry. You will be able to answer the question:

"Where do we fit in and why?"

In addition, you are contributing to a body of software engineering knowledge that is open, available and used for the betterment of the I.T. profession.

Anonymity

The anonymity of contributors to the ISBSG repository is assured. The ISBSG has a process in place that ensures that the identity of contributors remains confidential. Refer to the Project Submission Process chart.

Structure of the Collection Package

This data collection questionnaire is written as a Word Form, prompting entry in fields with some validation and drop-down lists where appropriate. It can be printed and used as a paper form if required. A combination of these methods could be used for separate sections. The data collection questionnaire contains seven sections.

- A. Submitter Information: Collects the submitter's details, which are kept confidential to ISBSG.
- B. Project Process: Collects information about how the project was performed.
- C. Technology: Collects information about the technology used on the project.
- D. People and Work Effort: Collects descriptive information about the people who worked on the project and the effort they expended. (No information is gathered about individuals).
- E. Product: Collects descriptions of the software product or application created or enhanced.
- F. Project Functional Size: Collects metrics on the amount of functionality the project delivered.
- G. Project Completion: Collects overview information on the project completion.

The format of this questionnaire has the numbered questions on the left and descriptive text on the right. **Particularly important questions that provide core data appear in bold typeface.**

Lodging Your Submission

Please send your submission to: ISBSG Repository Registrar

P.O. Box 550

Hawthorn, Victoria, 3122

AUSTRALIA

E-mail: admin@isbsg.org

Functional Size Approach

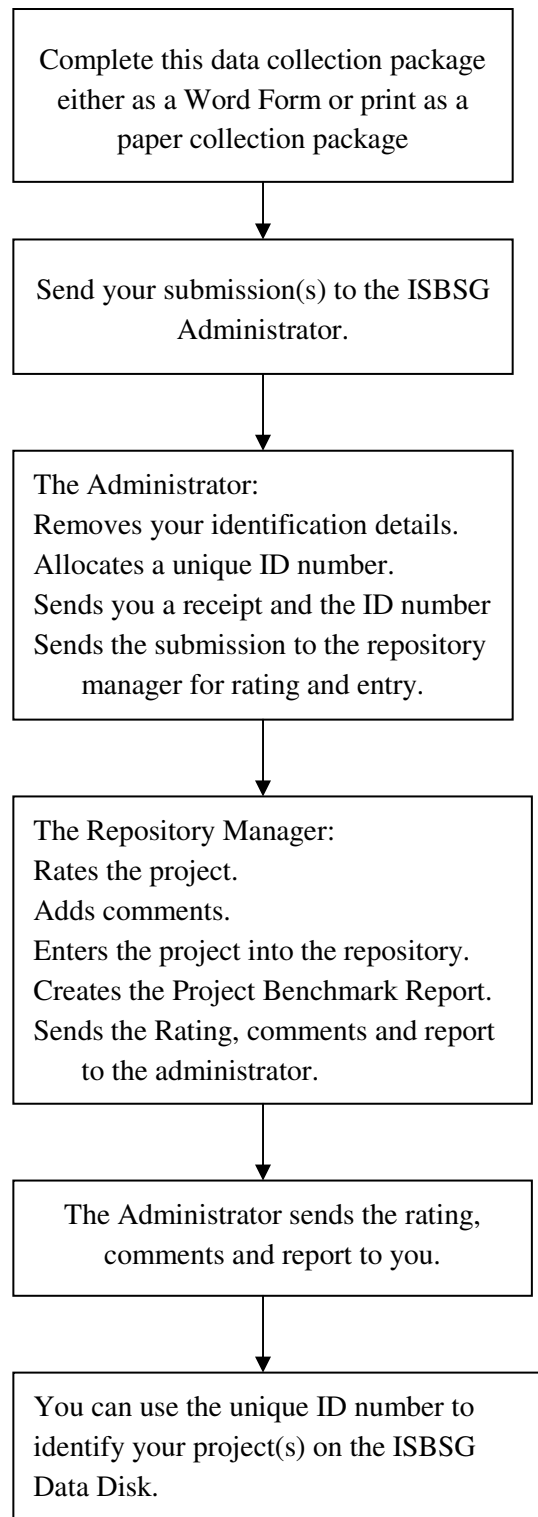
This data collection questionnaire is one of a set of questionnaires. This one is for recording an IFPUG or NESMA or FiSMA Function Point Analysis measurement of software size.

Other questionnaires provide forms for recording measurements with other methods. If your project was sized by a method other than IFPUG, NESMA or FiSMA, or was sized by another method in addition to IFPUG, NESMA or FiSMA, please use the appropriate form(s).

For the full list of ISBSG data collection questionnaires refer to our web site <http://www.isbsg.org> or contact ISBSG (address & email given above) or contact your local ISBSG representative.

If your project was sized by more than one method; when submitting forms all sections should be completed on the first form, and only sections A and F (the Submitter Information and Size Details) need to be completed on second (and subsequent) forms.

The Project Submission Process



A. Submitter Information

<p>1. Contact information for the questionnaire submitter</p> <p>Contact person: _____</p> <p>Organisation: _____</p> <p>Address: _____</p> <p>State/Province: _____ Postcode: _____</p> <p>Country: _____ Fax: _____</p> <p>Telephone: _____</p> <p>E-mail: _____</p>	<p>This information is necessary for our quality assurance processes. This data, like all information on this page, is kept confidential and is only seen by the ISBSG Administrator. This prevents users of ISBSG data from identifying project submitters. (Refer to our web site for details of our confidentiality policy.)</p>
<p>2. Are you willing for ISBSG to have further contact with you?</p> <p><input type="checkbox"/> Not at all <input type="checkbox"/> Only to clarify data items</p> <p><input type="checkbox"/> Only to provide feedback on the analyses</p> <p><input type="checkbox"/> Interested in ISBSG acting as an intermediary between companies for sharing information on an individual inquiry basis</p>	<p>If you do not respond to this question, we will assume that you are only willing to clarify data provided.</p>
<p>3. Your identifying name or ID for this submitted project.</p> <p>Project ID: _____</p> <p>Date Submitted: _____ (d-mmm-yy e.g. 5-Jul-01)</p>	<p>This allows identification of the Project Benchmark Report provided for your submitted project.</p> <p>Date you completed this questionnaire.</p>
<p>4. What was the role in this submitted project of the person who completed this questionnaire?</p> <p><input type="checkbox"/> Analyst or Programmer <input type="checkbox"/> Customer or End User</p> <p><input type="checkbox"/> Development Manager <input type="checkbox"/> Independent Reviewer</p> <p><input type="checkbox"/> IT/MIS Manager <input type="checkbox"/> Metrics Manager/Consultant</p> <p><input type="checkbox"/> Project Manager/Leader <input type="checkbox"/> Project Office/Tech. Support</p> <p><input type="checkbox"/> None</p> <p>Other (specify): _____</p>	<p>This information is used in our quality assurance processes.</p>
<p>5. Have you already submitted a questionnaire for this project, and this submission is to amend previous data?</p> <p><input type="checkbox"/> Yes please quote the ISBSG ID: _____</p> <p><input type="checkbox"/> No <input type="checkbox"/> Don't Know</p>	<p>We want the most accurate data practical, so we are keen to receive any amendments to data.</p> <p>Where projects have been sized by more than 1 method, submission of the additional counts is encouraged, (see note on Functional Count Approach on Page 4).</p>
<p>6. ISBSG Office Use Only</p> <p>ID: _____ Date Rec.: _____ Initials: _____</p>	<p>This question is for ISBSG's initial processing of your submission.</p>

B. Project Process

There is strong evidence that the process that a software project follows has a major impact on both its performance and its outcomes. In the software industry, there are no widely used, consistent terms describing software processes and a great variety of software processes are in use. Consequently, this section of the questionnaire gathers data within the structure of a simple, highly generic process. ISBSG is not assuming that all software projects should conform to this process. Instead we hope that the process your project followed can be approximately translated into this generic structure to allow accurate comparison with other projects.

PROJECT PRINCIPLE CHARACTERISTICS

<p>7. What type of software project was your project?</p> <p><input type="checkbox"/> New Development (For package acquisition, or support & maintenance, please use the ISBSG forms designed to collect data for these types of projects.)</p> <p><input type="checkbox"/> Re-development</p> <p><input type="checkbox"/> Enhancement</p> <p>Other (specify): _____</p>	<p>New Development: building a new software product in a context where the customer has no existing product meeting their requirements.</p> <p>Re-development: creating a software product with new technology that replaces or enhances a product that customers currently use.</p> <p>Enhancement: changing or extending the functionality of an existing product.</p>
<p>8. Which, if any, of the following techniques were used during the project?</p> <p><input type="checkbox"/> Agile Development <input type="checkbox"/> Business Area Modelling</p> <p><input type="checkbox"/> Data Modelling <input type="checkbox"/> Event Modelling</p> <p><input type="checkbox"/> Multifunctional Teams <input type="checkbox"/> Object Oriented Analysis</p> <p><input type="checkbox"/> Process Modelling <input type="checkbox"/> Object Oriented Design</p> <p><input type="checkbox"/> Personal Software Process (PSP) <input type="checkbox"/> Prototyping</p> <p><input type="checkbox"/> JAD (Joint Application Development)</p> <p><input type="checkbox"/> RAD (Rapid Application Development)</p> <p><input type="checkbox"/> Rational Unified Process (RUP)</p> <p><input type="checkbox"/> Team Software Process (TSP)</p> <p>Other (specify): _____</p>	<p>Software development teams have varied approaches used in software development.</p> <p>The ISBSG are interested in the primary categorisations of software development.</p>
<p>9. Did the project follow a defined and documented process?</p> <p><input type="checkbox"/> Yes <input type="checkbox"/> No (skip the next question)</p>	<p>The development team had documents describing the process, and standards or templates for most work products.</p>

<p>10. If yes to the question above, how was the process documentation acquired?</p> <p><input type="checkbox"/> Purchased <input type="checkbox"/> Developed In House <input type="checkbox"/> Combination</p>	<p>Choose only one option.</p>
<p>11. Is the development team involved in a process improvement program?</p> <p><input type="checkbox"/> Yes <input type="checkbox"/> No</p> <p>Which if any, software process or quality standards was the project performed under?</p> <p><input type="checkbox"/> Software-CMM Details: _____</p> <p><input type="checkbox"/> CMMI Details: _____</p> <p><input type="checkbox"/> SPICE Details: _____</p> <p><input type="checkbox"/> ISO 9002 Details: _____</p> <p><input type="checkbox"/> TICKIT Details: _____</p> <p><input type="checkbox"/> Other Details: _____</p>	<p>A process improvement program comprises a planned series of actions to improve how the software development team does its work.</p> <p>Where applicable, please provide additional details about the standards – such as their version, level and year of certification.</p> <p>Software standards typically define series of actions and documentation structures and contents required to deliver quality software and software processes.</p> <p>Many software standards are maintained by international organisations. Software developers are typically required to be formally and externally assessed in order to achieve certification to these standards.</p>
<p>12. Is the development team part of a software metrics program?</p> <p><input type="checkbox"/> Yes <input type="checkbox"/> No</p>	<p>A metrics program collects data such as effort, output & defects to analyse for estimating or process improvement.</p>

PLANNING

This covers both high-level project planning and preliminary requirements analysis. It focuses on objectives, stakeholders, risks, budgets and schedules.

<p>13. Please rank the following typical project objectives in order of priority where '1' is the most important and so on.</p> <p>___ Deliver all the functionality that users/customer needed.</p> <p>___ Deliver functionality with minimum possible defects.</p> <p>___ Deliver acceptable functionality at minimum cost.</p> <p>___ Deliver acceptable functionality in shortest time.</p>	<p>Some understanding of the project's objectives helps us interpret the responses to the rest of the questionnaire.</p>
<p>14. Which if any, documents or other work products were produced during the planning activity?</p> <p><input type="checkbox"/> Budget <input type="checkbox"/> Business Case <input type="checkbox"/> Feasibility Study</p> <p><input type="checkbox"/> Project Schedule <input type="checkbox"/> Proposal/Tender <input type="checkbox"/> Quality Plan</p> <p><input type="checkbox"/> Resource Plan <input type="checkbox"/> Risk Analysis</p> <p><input type="checkbox"/> Software Development Plan <input type="checkbox"/> Vision/Objectives</p> <p><input type="checkbox"/> None Other (specify): _____</p>	<p>When performed, project planning may produce documents, lists, charts or other work products.</p> <p>This question is trying to identify what content the project produced during any planning activity that took place. Producing the content requires effort, and thus knowledge of the content allows better comparison of projects. Hopefully this content also helps projects perform better.</p>
<p>15. What, if any, was the initial measure of the project's <i>functional size</i> made in project planning?</p> <p>Size: _____ Unit/Method: _____</p> <p>Other Method:</p> <p>What inputs(s) were used for the initial functional size?</p> <p><input type="checkbox"/> Data Model <input type="checkbox"/> CASE Tool <input type="checkbox"/> Use Case Model</p> <p><input type="checkbox"/> Other Project Example <input type="checkbox"/> Fixed Size/Budget (\$ per FP)</p> <p>Other (specify): _____</p>	<p>If no preliminary or initial functional size measurement was performed, such as an approximate Function Point (FP) count, skip this question.</p> <p>Unit/Method: functional counting using IFPUG FPs, COSMIC fsu etc.</p> <p>Other Method: there are several methods of making an initial FP measurement, such as using data models or use case models.</p> <p>Fixed Size/Budget: One method to set a project budget is to limit the functional size to a set figure.</p>
<p>16. Estimate of total <i>project effort</i> made in project planning?</p> <p>Effort: _____(hours)</p> <p>What method(s) was used to estimate the project effort?</p> <p><input type="checkbox"/> FSM Based <input type="checkbox"/> Task Breakdown <input type="checkbox"/> Tool Calculation</p> <p>Other (specify): _____</p>	<p>If no project effort estimate was made, then skip this question.</p> <p>FSM Based: Effort estimate was calculated directly or indirectly from a measurement of functional size such as function points.</p>

<p>17. Estimated project <i>completion date</i> set in project planning?</p> <p>Date: _____(d-mmm-yy)</p> <p>Which method(s) was used to set the project completion?</p> <p><input type="checkbox"/> FSM Based <input type="checkbox"/> Task Schedule <input type="checkbox"/> Tool Calculation</p> <p><input type="checkbox"/> Management Directive <input type="checkbox"/> Legal Requirement</p> <p>Other (specify): _____</p>	<p>If no project completion date was set, then skip this question.</p> <p>FSM Based: Completion date was calculated directly or indirectly from a measurement of functional size such as function points.</p>
<p>18. Estimate of total <i>project cost</i> made in project planning?</p> <p>Cost: _____ Currency: _____ Other: _____</p> <p>Are costs expressed in multiples of base currency unit? <input type="checkbox"/> No</p> <p><input type="checkbox"/> Yes (in which case what is the multiple? _____)</p> <p>What method(s) was used to estimate the project cost?</p> <p><input type="checkbox"/> FSM Based <input type="checkbox"/> Task Breakdown <input type="checkbox"/> Tool Calculation</p> <p><input type="checkbox"/> Fixed Budget Other (specify): _____</p>	<p>If no project cost estimate was made, then skip this question.</p> <p>If currency is not in list enter in 'other'.</p> <p>Some currencies are usually expressed in multiples (e.g. Lira or Yen). If so, please indicate the multiple.</p> <p>FSM Based: Effort estimate was calculated directly or indirectly from a measurement of functional size such as function points.</p>
<p>19. Size of any, if any, preliminary functional model created during project planning?</p> <p>_____ Model Objects: _____</p>	<p>Functional models comprise objects of specific types, such as data entities or use cases. A measure of the model size is the number of primary objects in the model, such as number of data entities.</p>
<p>20. What was the elapsed time of project planning?</p> <p>Start Date: _____ End Date: _____</p>	<p>Please enter dates as d-mmm-yy e.g. 5-Jul-01. This information allows better scheduling of future projects.</p>
<p>21. Comments on the project planning or estimates?</p> <p>_____</p> <p>_____</p> <p>_____</p> <p>_____</p>	<p>Any other comments on the project planning or estimates can be provided here.</p>

SPECIFICATION

Structured activity during which the development team works with the customer to identify and document what functionality, what interfaces, and what quality is required.

<p>22. Which, if any, documents or other work products were produced during the specification activity?</p> <p><input type="checkbox"/> System Concept Document <input type="checkbox"/> Project Treatment</p> <p><input type="checkbox"/> Requirements Spec. <input type="checkbox"/> Functional Specification</p> <p><input type="checkbox"/> System Analysis Report <input type="checkbox"/> User Manual</p> <p><input type="checkbox"/> User Interface Prototype <input type="checkbox"/> Graphical Look & Feel</p> <p><input type="checkbox"/> Logical Data/ER Model <input type="checkbox"/> Data Flow Model</p> <p><input type="checkbox"/> Event Model <input type="checkbox"/> Use Case Model</p> <p><input type="checkbox"/> State Transition Model <input type="checkbox"/> Storyboards</p> <p><input type="checkbox"/> External/System Interface Specifications</p> <p><input type="checkbox"/> None</p> <p>Other (specify): _____</p>	<p>When performed, the specification activity may produce documents, diagrams or prototypes. This question is trying to identify what content the project produced during any specification activity that took place.</p> <p>We understand that the work products produced, if any, will vary according to the preferred process of your development team, and the project scope and risk.</p> <p>ER Model: Entity/Relationship Model</p>
<p>23. Size of any, if any, functional model(s) created during the specification activity?</p> <p>_____ Model Objects: _____</p>	<p>Functional models comprise objects of specific types, such as data entities or use cases. A measure of the model size is the number of primary objects in the model, e.g. number of data entities.</p>
<p>24. Which, if any, of the following techniques were used during the project?</p> <p><input type="checkbox"/> JAD (Joint Application Design) <input type="checkbox"/> Timeboxing</p> <p><input type="checkbox"/> Specification Review Other (specify): _____</p>	<p>Software development teams have varied approaches to the specification of software. We are trying to identify the activities that involve significant effort or cost, and thus influence how a project is compared to others.</p>
<p>25. What was the number of defects recorded in the documents and other work products of the specification activity (if any)?</p> <p>Defects: _____ Resolution/rework effort: _____ (hours)</p>	<p>This information helps identify the benefits, if any, of specification reviews.</p>
<p>26. What was the functional size measured after the specification activity (if it was measured)?</p> <p>Size: _____ Unit/Method: _____</p> <p>Other Method: _____</p>	<p>If there was no functional size measure performed after specification, such as an FP count, skip this question. This information helps identify 'scope creep', or requirements growth.</p>
<p>27. What was the elapsed time of the specification activity?</p> <p>Start Date: _____ End Date: _____</p>	<p>Please enter dates as d-mmm-yy e.g. 5-Jul-01. This information allows better scheduling of future projects.</p>

DESIGN

Activity during which the development team creates a general, high-level design of the software structure, then possibly a detailed design. This activity often overlaps with specification and build.

<p>28. Which, if any, documents or other work products were produced during the design activity?</p> <p><input type="checkbox"/> System Architecture Spec. <input type="checkbox"/> Technical Specification</p> <p><input type="checkbox"/> User Interface Prototype <input type="checkbox"/> Graphical Look & Feel</p> <p><input type="checkbox"/> Technical Prototypes <input type="checkbox"/> Component Designs</p> <p><input type="checkbox"/> Database/Data File Design <input type="checkbox"/> Software Interface Designs</p> <p><input type="checkbox"/> Detailed Design Documents <input type="checkbox"/> Module/Unit/Class Designs</p> <p><input type="checkbox"/> None</p> <p>Other (specify): _____</p>	<p>When performed, the design activity may produce documents, diagrams or prototypes. (Some of these different work products may have been compiled together into a single document.)</p> <p>We understand that the work products produced, if any, will vary according to the preferred process of your development team, and the project scope and risk.</p>
<p>29. Which, if any, of the following techniques were used during the project?</p> <p><input type="checkbox"/> JAD (Joint Application Design)</p> <p><input type="checkbox"/> UML (Unified Modelling Language)</p> <p><input type="checkbox"/> Structured Design <input type="checkbox"/> Design Review/Inspection</p> <p><input type="checkbox"/> Usability Review/Testing</p> <p>Other (specify): _____</p>	<p>Software development teams have varied approaches to software design.</p>
<p>30. What was the number of defects recorded during design (if any)?</p> <p>Defects: _____ Resolution/rework effort: _____ (hours)</p>	<p>This information helps identify the benefits, of any, of design review/inspection.</p>
<p>31. What was the number of changes raised during design (if any)?</p> <p>Specification Changes Raised: _____</p>	<p>This information helps measure ‘scope creep’, which is a major issue in software development.</p>
<p>32. What was functional size measured after completion of design (if it was measured)?</p> <p>Size: _____ Unit/Method: _____</p> <p>Other Method: _____</p>	<p>If there was no functional size measurement, such as an FP count, performed after design, skip this question.</p> <p>This information helps identify ‘scope creep’, or requirements growth.</p>
<p>33. What was the elapsed time of the design activity?</p> <p>Start Date: _____ End Date: _____</p>	<p>Please enter dates as d-mmm-yy e.g. 5-Jul-01. This information allows better scheduling of future projects.</p>

BUILD OR PROGRAMMING

Development team performs the programming and unit testing that produces new software or changes existing software. This activity often consumes a major part of the schedule or effort.

<p>34. What was produced or modified during the build activity?</p> <p><input type="checkbox"/> Source Code/Objects <input type="checkbox"/> Database Objects</p> <p><input type="checkbox"/> Graphics Items/Images <input type="checkbox"/> Information/Text Content</p> <p><input type="checkbox"/> Unit Test Plans/Designs <input type="checkbox"/> Integration Scripts/Checklist</p> <p><input type="checkbox"/> None</p> <p>Other (specify): _____</p>	<p>Not all development projects perform the build activity, some may just involve specification activity, with build performed as a separate project.</p> <p>Source Code/Objects: The executable instructions created with the programming language.</p>
<p>35. Which, if any, of the following detailed activities occurred during the project?</p> <p><input type="checkbox"/> Unit Testing <input type="checkbox"/> Code Review/Inspection</p> <p><input type="checkbox"/> Software Integration <input type="checkbox"/> System Integration</p> <p>Other (specify): _____</p>	<p>Unit testing: Test by developers of their own source code/objects, typically against the structure the code/objects.</p> <p>Software Integration: Combine the software modules/units/classes to produce software components.</p> <p>System Integration: Integrate the software components with other software components, and with hardware items etc as necessary, to produce a complete system.</p>
<p>36. What was the number of defects recorded and resolved during the build activity (if any)?</p> <p>Minor: _____ Major: _____ Extreme: _____</p> <p>Or Total Defects: _____</p> <p>Resolution/rework effort: _____ (hours)</p>	<p>This information helps identify the benefits, if any, of code review/inspection, and of unit testing.</p> <p>Minor defect: does not make the software unusable in any way (e.g. a modification required to a report).</p> <p>Major defect: causes part of the software to become unusable.</p> <p>Extreme defect: failure causing the software to become totally unusable.</p>
<p>37. What was the number of changes raised during build (if any)?</p> <p>Specification Changes Raised: _____</p>	<p>This information helps measure 'scope creep', which is a major issue in software development.</p>
<p>38. What was the elapsed time of the build activity?</p> <p>Start Date: _____ End Date: _____</p>	<p>Please enter dates as d-mmm-yy e.g. 5-Jul-01. This information allows better scheduling of future projects.</p>

TEST

Planning and performing the various levels of testing on the software by people who may be independent of the developers. This activity often overlaps with build.

<p>39. Which, if any, documents or other work products were produced during the planning or performance of testing?</p> <p><input type="checkbox"/> Test Plans/Designs <input type="checkbox"/> Test Scripts/Harnesses</p> <p><input type="checkbox"/> Test Data <input type="checkbox"/> Incident/Defect Lists</p> <p><input type="checkbox"/> None</p> <p>Other (specify): _____</p>	<p>The ‘Test’ activity covers testing against the specifications. When performed, the test activity may produce documents, lists, or other work products.</p> <p>The different work products produced may explain some of the variation in effort expended on testing.</p>
<p>40. Which, if any, of the following detailed activities occurred during the testing of the software?</p> <p><input type="checkbox"/> Create & Run Automated Testing</p> <p><input type="checkbox"/> Performance, Stress or Load Testing</p> <p><input type="checkbox"/> Regression Testing</p> <p>Other (specify): _____</p>	<p>Software development teams have varied approaches to testing software. We are trying to identify the activities that involve significant effort or cost, and thus influence how a project is compared to others.</p>
<p>41. What was the number of defects recorded during the testing activity (if any)?</p> <p>Minor: _____ Major: _____ Extreme: _____</p> <p>Or Total Defects: _____</p> <p>Resolution/rework effort: _____ (hours)</p>	<p>Minor defect: does not make the software unusable in any way (e.g. a modification required to a report).</p> <p>Major defect: causes part of the software to become unusable.</p> <p>Extreme defect: failure causing the software to become totally unusable.</p>
<p>42. What was the number of changes raised during testing (if any)?</p> <p>Specification Changes Raised: _____</p>	<p>This information helps measure ‘scope creep’, which is a major issue in software development.</p>
<p>43. What was the elapsed time of the testing activity?</p> <p>Start Date: _____ End Date: _____</p>	<p>Please enter dates as d-mmm-yy e.g. 5-Jul-01. This information allows better scheduling of future projects.</p>

IMPLEMENTATION OR INSTALLATION

Preparing for the installation of the software by/for customer or end user personnel, then working with them for installation, user documentation and training.

<p>44. Which, if any, documents or other work products were produced during preparation for, or performance of, the implementation activity?</p> <p><input type="checkbox"/> Installation Procedures <input type="checkbox"/> Release Description/Notes</p> <p><input type="checkbox"/> Installation Software Utility <input type="checkbox"/> Incident/Defect Lists</p> <p><input type="checkbox"/> End user Manual or Help <input type="checkbox"/> Operation/Admin. Manual</p> <p><input type="checkbox"/> End user Training Material <input type="checkbox"/> Data Establishment Process</p> <p><input type="checkbox"/> None Other (specify): _____</p>	<p>When being prepared for or performed, the implementation activity may produce documents or lists, or other work products.</p> <p>This question is trying to identify what content the project produced during any preparation for, or any implementation activity that took place. Producing the content requires effort, and thus knowledge of the content allows better comparison of projects. Hopefully this content also helps projects perform better.</p>
<p>45. How many distinct release/versions of the software were delivered to the customer or end users during the project?</p> <p>Releases primarily providing new functionality: _____</p> <p>Releases primarily providing defect repairs: _____</p>	<p>A distinct release/version has different functionality or performance from its predecessors. This release may be part of an incremental/staged software delivery plan, or it may be a defect repair release (or both).</p>
<p>46. Which, if any, of the following detailed activities occurred during the implementation of the software?</p> <p><input type="checkbox"/> Prepare Releases for Delivery/Installation</p> <p><input type="checkbox"/> Install Software for Users <input type="checkbox"/> Acceptance/Beta Testing</p> <p><input type="checkbox"/> Provide User Training <input type="checkbox"/> Provide User Support</p> <p><input type="checkbox"/> Data Establishment</p> <p>Other (specify): _____</p>	<p>Software development teams have varied approaches to implementing software. We are trying to identify the activities that involve significant effort or cost, and thus influence how a project is compared to others.</p>
<p>47. What was the number of defects recorded during the implementation activity (if any)?</p> <p>Minor: _____ Major: _____ Extreme: _____</p> <p>Or Total Defects: _____</p> <p>Resolution/rework effort: _____ (hours)</p>	<p>Minor defect: does not make the software unusable in any way (e.g. a modification required to a report).</p> <p>Major defect: causes part of the software to become unusable.</p> <p>Extreme defect: failure causing the software to become totally unusable.</p>
<p>48. What was the number of changes raised during implementation (if any)?</p> <p>Specification Changes Raised: _____</p>	<p>This information helps measure ‘scope creep’, which is a major issue in software development.</p>
<p>49. What was the functional size measured after completion of implementation (if it was measured)?</p> <p>Size: _____ Unit/Method: _____</p> <p>Other Method: _____</p>	<p>If there was no functional size measurement, such as an FP count, performed after implementation, skip this question.</p>
<p>50. What was the elapsed time of the implementation activity?</p> <p>Start Date: _____ End Date: _____</p>	<p>Please enter dates as d-mmm-yy e.g. 5-Jul-01. This information allows better scheduling of future projects.</p>

PROJECT MANAGEMENT AND MONITORING

<p>51. Which, if any, documents or other work products were produced during project management?</p> <p><input type="checkbox"/> Budget Variations <input type="checkbox"/> Change Lists <input type="checkbox"/> Issue Lists</p> <p><input type="checkbox"/> Project Reviews <input type="checkbox"/> Project Log <input type="checkbox"/> Status Reports</p> <p><input type="checkbox"/> Meeting Minutes <input type="checkbox"/> Change Control Procedures</p> <p><input type="checkbox"/> Project Plan Revisions <input type="checkbox"/> Risk Assessments</p> <p><input type="checkbox"/> None</p> <p>Other (specify): _____</p>	<p>When performed, the project management activity often produces documents or lists. (Some of these different work products may have been compiled together into a single document.)</p>
<p>52. Which, if any, of the following detailed activities occurred during project management and monitoring?</p> <p><input type="checkbox"/> Manage Customer and End user Relations</p> <p><input type="checkbox"/> Manage 3rd-Party Suppliers and/or Sub-Contractors</p> <p><input type="checkbox"/> Manage Development Team <input type="checkbox"/> Manage Quality Assurance</p> <p>Other (specify): _____</p>	<p>Project managers have varied approaches to project management. We are trying to identify the activities that involve significant effort or cost, and thus influence how a project is compared to others.</p>
<p>53. What was the decision making structure in place for the project?</p> <p><input type="checkbox"/> Customer Project Sponsor <input type="checkbox"/> Change Control Board</p> <p><input type="checkbox"/> Issues Resolution System <input type="checkbox"/> Steering Committee</p> <p><input type="checkbox"/> None</p> <p>Other (specify): _____</p>	<p>Decision-making process between customer and the development organisation appears to have a significant impact on the project.</p>

C. Technology

The technology used in software development has a significant impact on the performance of the project. Typically, each major activity within the process requires its own technology.

<p>54. Which tools, if any, were used during project <i>planning</i>?</p> <p style="text-align: center;">Yes No Tool:</p> <p>Task & Resource Planning <input type="checkbox"/> <input type="checkbox"/> _____</p> <p>Project Estimating <input type="checkbox"/> <input type="checkbox"/> _____</p> <p>Other (specify): _____</p>	<p>Task & Resource Planning: tools that assist with organising the task breakdown of the project, and allocating resources to tasks.</p> <p>Project Estimating: tools that support or perform the calculation of estimates for delivery date, effort and cost.</p>
<p>55. Which tools, if any, were used during the <i>specification</i> activity?</p> <p style="text-align: center;">Yes No Tool:</p> <p>Requirements Management <input type="checkbox"/> <input type="checkbox"/> _____</p> <p>Spec. Writing & Delivery <input type="checkbox"/> <input type="checkbox"/> _____</p> <p>CASE Tool <input type="checkbox"/> <input type="checkbox"/> _____</p> <p>User Interface Prototyping <input type="checkbox"/> <input type="checkbox"/> _____</p> <p>Other (specify): _____</p>	<p>Requirements Management: tools used to record and manage the large numbers of distinct requirements that can occur on a software project.</p> <p>Spec. Writing & Delivery: tools used to compile and deliver the specification documents.</p> <p>CASE Tool: Computer Aided Software Engineering tools, typically used to develop functional models.</p>
<p>56. Which tools, if any, were used during the <i>design</i> activity?</p> <p style="text-align: center;">Yes No Tool:</p> <p>Spec. Writing & Delivery <input type="checkbox"/> <input type="checkbox"/> _____</p> <p>CASE Tool <input type="checkbox"/> <input type="checkbox"/> _____</p> <p>Other (specify): _____</p>	<p>Spec. Writing & Delivery: tools used to compile and deliver the specification documents.</p> <p>CASE Tool: Computer Aided Software Engineering tools, typically used to develop design models and diagrams.</p>

<p>57. What was the primary technology used to build or enhance the software? i.e. that used for most of the build effort.</p> <p style="text-align: center;">Yes No Primary Tool:</p> <p>Programming Language _____</p> <p>Operating System <input type="checkbox"/> <input type="checkbox"/> _____</p> <p>Integrated Development Environment <input type="checkbox"/> <input type="checkbox"/> _____</p> <p>Debugging <input type="checkbox"/> <input type="checkbox"/> _____</p> <p>Database <input type="checkbox"/> <input type="checkbox"/> _____</p> <p>Object/Component Server <input type="checkbox"/> <input type="checkbox"/> _____</p> <p>HTML/Web Server <input type="checkbox"/> <input type="checkbox"/> _____</p> <p>E-Mail or Message Server <input type="checkbox"/> <input type="checkbox"/> _____</p> <p>Other (specify): _____</p>	<p>This is a question for which a response is essential if the project data is to be useful.</p> <p>Programming Language: primary language/tool used to create the source code/objects.</p> <p>Integrated Development Environment: development environment integrating a range of tools to aid the processes of designing, constructing and testing the software; typically, incorporating graphical and component based development techniques.</p> <p>Debugging: tools specifically to identify location of software defects.</p> <p>Database: specific tool used for persistent data storage that is distinct from the programming language.</p> <p>Object/Component Server: tool under which software objects execute for multiple users e.g. CORBA broker or Microsoft™ MTS.</p> <p>E-Mail or Message Server: tool used specifically to support messages between different software systems.</p>
<p>58. What other development technology was involved?</p> <p style="text-align: center;">Yes No Other Platforms:</p> <p>Programming Language _____</p> <p>Operating System <input type="checkbox"/> <input type="checkbox"/> _____</p> <p>Database <input type="checkbox"/> <input type="checkbox"/> _____</p> <p>Object/Component Server <input type="checkbox"/> <input type="checkbox"/> _____</p> <p>HTML/Web Server <input type="checkbox"/> <input type="checkbox"/> _____</p> <p>E-Mail or Message Server <input type="checkbox"/> <input type="checkbox"/> _____</p> <p>Other (specify): _____</p>	<p>Many software projects now develop software systems that involve multiple computer systems operating on distinct platforms, using multiple development technologies. This question is to identify such projects, so that we can analyse the impact of distributed architectures on project performance.</p>
<p>59. What would you consider to be the environment in which software was developed?</p> <p><input type="checkbox"/> PC or microcomputer</p> <p><input type="checkbox"/> Mid Range</p> <p><input type="checkbox"/> Main Frame</p> <p><input type="checkbox"/> Multi platform</p> <p>Other (specify): _____</p>	<p>The ISBSG classify the development platform of the project to be the environment in which the software was developed, based on primarily the development operating system.</p> <p>This question provides primary input to that classification.</p> <p>A Multi platform environment would include aspects of more than one of the categories Mainframe, Midrange, or PC.</p>

<p>60. Which tools, if any, were used during the <i>testing</i> activity?</p> <p style="text-align: center;">Yes No Tool:</p> <p>Code Checking <input type="checkbox"/> <input type="checkbox"/> _____</p> <p>Test Planning & Management <input type="checkbox"/> <input type="checkbox"/> _____</p> <p>Automated Test Performance <input type="checkbox"/> <input type="checkbox"/> _____</p> <p>Performance Monitoring <input type="checkbox"/> <input type="checkbox"/> _____</p> <p>Other (specify): _____</p>	<p>Code Checking: tools used to identify potential defects in the code through automatic checking.</p> <p>Automated Test Performance: tools used to execute the software automatically to locate defects.</p> <p>Performance Monitoring: tools used to monitor the execution performance of specific software, or a distributed software system.</p>
<p>61. Which tools, if any, were used to support <i>project management</i>?</p> <p style="text-align: center;">Yes No Tool:</p> <p>Version Control/Archiving <input type="checkbox"/> <input type="checkbox"/> _____</p> <p>Time/Effort Recording <input type="checkbox"/> <input type="checkbox"/> _____</p> <p>Defect & Change Tracking <input type="checkbox"/> <input type="checkbox"/> _____</p> <p>Status Tracking & Reporting <input type="checkbox"/> <input type="checkbox"/> _____</p> <p>Other (specify): _____</p>	<p>Version Control/Archiving: tools used to maintain a master version of source code/objects and control who has the ability to update them.</p> <p>Status Tracking & Reporting: Tool to track the performance or status of the project relative to the project plan, and to report this status to appropriate stakeholders.</p>
<p>62. What was the implementation platform of the software product? i.e. that which the software was implemented into.</p> <p>Is the implementation platform the same as development?</p> <p><input type="checkbox"/> Yes (skip rest of question) <input type="checkbox"/> No (please provide details)</p> <p style="text-align: center;">Primary Platform:</p> <p>Implementation Platform <input type="checkbox"/> Device Embedded</p> <p style="padding-left: 100px;"><input type="checkbox"/> PC</p> <p style="padding-left: 100px;"><input type="checkbox"/> MR (Mid range)</p> <p style="padding-left: 100px;"><input type="checkbox"/> MF (Mainframe)</p> <p style="padding-left: 100px;"><input type="checkbox"/> Multi Platform</p> <p>If device embedded, please specify the target</p> <p><input type="checkbox"/> Automotive <input type="checkbox"/> Aviation</p> <p><input type="checkbox"/> Domestic appliance <input type="checkbox"/> Machine tool</p> <p><input type="checkbox"/> Mobile phone <input type="checkbox"/> PDA</p> <p><input type="checkbox"/> Games device <input type="checkbox"/> Music device</p> <p>Other (specify): _____</p>	<p>The implementation platform may be different from that on which the software was developed, or may be the only platform known for the project.</p> <p>A Multi platform environment would include aspects of more than one of the categories Mainframe, Midrange, or PC.</p> <p>For device embedded software, please specify the generic device into which the software is implemented.</p>

D. People and Work Effort

For software projects, there are typically three groups of people involved.

- Development team or organisation, which specifies, designs and/or builds the software. It typically also performs testing and implementation activities.
- Customers and end users, who provide the requirements for the software, pay for the software development and use the software when it goes into operation. The relationship between the project customer and the software’s end users can vary, as can their involvement in a software project.
- IT Operations, who operate the IT systems that support the end users. The development team often has an IT operations group supporting it, which will be regarded as part of the development team.

This section gathers information on the experience, roles, effort expended and number of the different people involved in the project.

DEVELOPMENT TEAM

<p>63. In which country did the development team perform most of the project work? _____ Other: _____</p> <p>In which country was the project implemented? _____ Other: _____</p>	<p>If country is not in drop-down list enter in ‘other’. If implemented in multiple countries state this in ‘other’. This data allows demographic analysis by country. The country of an individual project will not be published however.</p>
<p>64. What was the experience of team members in the end users’ area of business at the time they joined the project?</p> <p>Number of members with < 1 years experience: _____</p> <p>Number of members with 1 - 3 years experience: _____</p> <p>Number of members with > 3 years experience: _____</p>	<p>Experience in the end users’ business helps communication with them and with understanding requirements. This may impact project performance.</p>
<p>65. What was team member experience in software development?</p> <p>Number of members with < 3 years experience: _____</p> <p>Number of members with 3 – 9 years experience: _____</p> <p>Number of members with > 9 years experience: _____</p>	<p>Conventional wisdom indicates that the experience of the development team has a major impact on project performance.</p>
<p>66. What was the experience of the project manager responsible for most of the project?</p> <p>Number of past projects (IT & non-IT) managed: _____</p>	<p>The experience of the project manager may be a major factor in the project’s performance.</p>
<p>67. How stable was the development team during the project?</p> <p>Number of team members unexpectedly replaced during course of the project: _____</p> <p>Number of changes of project manager: _____</p>	<p>Team members can change unexpectedly because of resignations, illness, or dismissal.</p>

68. **The following table is used to gather information about the composition of the development team and the effort that it expended. Most important is data for the number of distinct people involved in the project and the total effort, (in hours), that they expended.**

For each role there is a pair of entry boxes; one for the number of distinct people involved in that role; one for the effort expended on that role. When one person performs multiple roles on a project, we only want to count that person once as a team member, so enter the person's count against the primary role, but as we want to get the effort break down by role, record his or her effort distributed across each role that the person performed. (This second approach could result in effort recorded against a role that was no one person's primary role.)

Role		Num. of distinct people & their total effort (hrs)	Role		Num. of distinct people & their total effort (hrs)
Project Manager or Leader	People		Requirements or Business Analyst	People	
	Effort	hours		Effort	hours
Software Architect or Designer	People		User-Interface Designer	People	
	Effort	hours		Effort	hours
Graphic Artist or Designer	People		Developers/ Programmers	People	
	Effort	hours		Effort	hours
QA/Testers	People		End user Training & Documentation	People	
	Effort	hours		Effort	hours
Database Administrator	People		IT System Administrator	People	
	Effort	hours		Effort	hours
Other (specify): _____	People		Other (specify): _____	People	
	Effort	hours		Effort	hours

69. This table attempts to gather the development team effort (in hours) expended in each major activity of the 'generic' project process, and the number of team members involved in each activity. (For guidance on activity/phase contents please refer to the Glossary of Terms).

		Enter numbers of people & their effort for each activity						Or summary values for the whole project
		Plan	Specify	Design	Build	Test	Impl.	
Development Team	People							
Totals	Effort							hours

The most important data is the number of distinct people involved in the development team, and the total effort expended by the development team. Without this data, we cannot perform any significant analysis on the project. The next most important data is the total effort and number of people performing each role.

CUSTOMERS AND END USERS

The project customer is the organisation that funds the software development. End users are the people who physically interact with the software.

<p>70. In which industry(s) do the software's end users work?</p> <p><input type="checkbox"/> Aerospace/Automotive <input type="checkbox"/> Agriculture, Forestry etc.</p> <p><input type="checkbox"/> Banking <input type="checkbox"/> Chemicals</p> <p><input type="checkbox"/> Communications <input type="checkbox"/> Community Services</p> <p><input type="checkbox"/> Computers & Software <input type="checkbox"/> Construction</p> <p><input type="checkbox"/> Consumer Goods <input type="checkbox"/> Defence</p> <p><input type="checkbox"/> Education Institution <input type="checkbox"/> Electricity, Gas & Water</p> <p><input type="checkbox"/> Electronics <input type="checkbox"/> Food Processing</p> <p><input type="checkbox"/> Finance & Business Service <input type="checkbox"/> Government</p> <p><input type="checkbox"/> Insurance <input type="checkbox"/> Manufacturing</p> <p><input type="checkbox"/> Media <input type="checkbox"/> Medical/Health Care</p> <p><input type="checkbox"/> Mining <input type="checkbox"/> Oil & Petroleum</p> <p><input type="checkbox"/> Professional Services <input type="checkbox"/> Real Estate & Property</p> <p><input type="checkbox"/> Recreation & Personnel Services <input type="checkbox"/> Transport & Storage</p> <p><input type="checkbox"/> Wholesale & Retail</p> <p>Other (specify): _____</p>	<p>There is strong evidence that the type of industry for which the application or software product is intended, or in which it is currently being used, has an impact on the project performance</p>
<p>71. What was the relationship between the project's customer, end users and development team?</p> <p><input type="checkbox"/> Customer, end users & development team all in the same organisation.</p> <p><input type="checkbox"/> Customer & end-users in one organisation; development team in another organisation(s).</p> <p><input type="checkbox"/> Customer & development team in one organisation; end users in other organisations.</p> <p><input type="checkbox"/> Customer, end users and development team each in different organisation(s). Other (specify): _____</p>	<p>There is anecdotal evidence that the nature of this relationship has an impact on project performance, and on the project process.</p>
<p>72. What is the expected or known size of end user base?</p> <p>Number of distinct installations: _____</p> <p>Number of distinct end users: _____</p> <p>Maximum num. of concurrent end users: _____</p>	<p>A 'distinct installation' is an individual installation of the complete software system.</p> <p>Concurrent end users applies to single distinct installation.</p>

73. The following table is used to gather information about the different roles of the customer and end users involved in the project. Most important is data for the number of distinct people involved. If a person performed more than one role on the project, record his or her involvement only against that person's primary role.

Role		Num. of distinct people & their total effort	Role		Num. of distinct people & their total effort
Project/Executive	People		Business Manager	People	
Sponsor	Effort	hours		Effort	hours
Project Manager	People		Marketing/Sales	People	
	Effort	hours	Rep. or Manager	Effort	hours
End user Rep.	People		Training or Human	People	
	Effort	hours	Resources Rep.	Effort	hours
Other (specify):	People		Other (specify):	People	
_____	Effort	hours	_____	Effort	hours

74. This table attempts to gather the number of customer and end user personnel involved in each major activity, and if possible, the effort that they expended. (For guidance on activity/phase contents please refer to the Glossary of Terms).

		Enter numbers of people & their effort for each activity						Or summary values for the whole project
		Plan	Specify	Design	Build	Test	Impl.	
Customer / End	People							
User Totals	Effort							hours

IT OPERATIONS

This is the IT personnel who support the customer and/or end users.

75. This table attempts to gather information about the different roles of IT operations personnel involved in the project. **Most important is data for the number of distinct people involved.**

If a person performed more than one role on the project, record his or her involvement in the project only against that person's primary role.

Role		Num. of distinct people & their total effort (hrs)	Role		Num. of distinct people & their total effort (hrs)
IT Manager	People		IT Project Manager or Leader	People	
	Effort	hours		Effort	hours
IT Database Administrator	People		IT System Administrator	People	
	Effort	hours		Effort	hours
Other (specify): _____	People		Other (specify): _____	People	
	Effort	hours		Effort	hours

76. This table attempts to gather the number of IT operations personnel involved in each major activity and if possible, the effort that they expended. (For guidance on activity/phase contents please refer to the Glossary of Terms).

		Enter numbers of people & their effort for each activity					Or summary values for the whole project
		Plan	Specify	Design	Build	Test	
IT Operations	People						
Totals	Effort						hours

WORK EFFORT VALIDATION

The following questions support our data quality procedures.

<p>77. What procedure, if any, was used to record effort spent on the project by development team/organisation?</p> <p><input type="checkbox"/> <i>No timesheets</i> were recorded by the development team.</p> <p><input type="checkbox"/> Recorded only the total <i>hours worked each day</i> or week.</p> <p><input type="checkbox"/> Recorded <i>hours worked on each project</i> for each day/week.</p> <p><input type="checkbox"/> Recorded the <i>work done on each project task</i> for each day.</p> <p>Other (specify): _____</p>	<p>Effort data for the development team is core to ISBSG analysis of projects, and to the comparison of projects.</p>
<p>78. If the development team recorded hours worked on each project for each day, what was the percentage of their total work hours was devoted to project work (if known)?</p> <p>% of total work hours devoted to project work: _____</p>	<p>People need to work at a range of non-project activities such as training, administration, leave and marketing. Very rarely are 100% of work hours devoted to project work over the whole duration of a project.</p>
<p>79. If the development team did not record timesheets, or timesheet data was unavailable, how did you calculate their effort expended on the project?</p> <p>_____</p> <p>_____</p> <p>_____</p>	<p>Because of the significance of effort data to project benchmarking, this question is important for our data quality processes.</p>
<p>80. Has all the work done been included in the effort figure?</p> <p><input type="checkbox"/> Yes (skip the next question) <input type="checkbox"/> No</p>	<p>For example, do the figures include unpaid overtime, work done from home, initial planning effort?</p>
<p>81. If no to the question above, what do you estimate the uncollected effort to be?</p> <p><input type="checkbox"/> Less than 5% of recorded <input type="checkbox"/> 5 – 10% of recorded effort</p> <p>Other (specify): _____ <input type="checkbox"/> Unable to estimate</p>	<p>Uncollected effort data makes a project appear more effective than it really was, which typically results in unrealistic future expectations.</p>
<p>82. What procedure, if any, was used to record time spent on the project by customer and end user personnel?</p> <p><input type="checkbox"/> No timesheets were recorded by customers or end users.</p> <p><input type="checkbox"/> Recorded only the total hours worked each day or week.</p> <p><input type="checkbox"/> Recorded the hours worked on each project or job each day.</p> <p><input type="checkbox"/> Recorded the work done on each project/job task each day.</p> <p>Other (specify): _____</p>	<p>Effort data for customers and end users helps informing future customers and end users what commitment they need to make for project success.</p>

<p>83. What procedure, if any, was used to record time spent on the project by IT operations personnel who support the end users?</p> <p><input type="checkbox"/> No timesheets were recorded by IT operations.</p> <p><input type="checkbox"/> Recorded only the total hours worked each day or week.</p> <p><input type="checkbox"/> Recorded the hours worked on each project or job each day.</p> <p><input type="checkbox"/> Recorded the work done on each project/job task each day.</p> <p>Other (specify): _____</p>	<p>Effort data for IT operations helps planning what commitment they need to make for project success.</p>
<p>84. How would you rate the quality of the work effort data?</p> <p><input type="checkbox"/> Poor <input type="checkbox"/> Adequate <input type="checkbox"/> Good <input type="checkbox"/> Excellent</p>	
<p>85. Why did you assign the above quality rating?</p> <p>_____</p> <p>_____</p> <p>_____</p>	<p>This assists our data quality processes on work effort data, which is core data for project analysis.</p>

E. Product

This section of questions gathers information on the software product itself.

<p>86. What type of software application has the project produced, or is enhancing?</p> <p><input type="checkbox"/> 3D modelling or animation <input type="checkbox"/> Catalogue or register of things or events</p> <p><input type="checkbox"/> Customer billing or relationship management <input type="checkbox"/> Document management</p> <p><input type="checkbox"/> Device or interface driver <input type="checkbox"/> Electronic data interchange</p> <p><input type="checkbox"/> Financial transaction processing & accounting <input type="checkbox"/> Geographic or spatial information systems</p> <p><input type="checkbox"/> Graphics & publishing tools or system <input type="checkbox"/> Image, video or sound processing</p> <p><input type="checkbox"/> Software for machine control <input type="checkbox"/> Job, case, incident or project management</p> <p><input type="checkbox"/> Logistic or supply planning & control <input type="checkbox"/> Management or performance reporting</p> <p><input type="checkbox"/> Mathematical modelling (finance or engineering) <input type="checkbox"/> Network management</p> <p><input type="checkbox"/> Online analysis and reporting <input type="checkbox"/> Operating system or software utility</p> <p><input type="checkbox"/> Personal productivity (e.g. word processor, spreadsheet) <input type="checkbox"/> Process control</p> <p><input type="checkbox"/> Software development tool <input type="checkbox"/> Stock control & order processing</p> <p><input type="checkbox"/> Trading <input type="checkbox"/> Workflow support & management</p> <p>Other (specify): _____</p>	<p>Conventional wisdom indicates that the type of software application being developed, or being enhanced, has an impact on the project's productivity.</p>
<p>87. Does the software application or product require more than one computer to operate different components or parts of it?</p> <p><input type="checkbox"/> Yes <input type="checkbox"/> No (skip the next two questions)</p>	<p>Many software applications perform all functionality running on one computer. Other software applications are designed with components running on different computers e.g. Client/server.</p>

<p>88. Of the roles below, which roles do the computers providing the interface to the software's external users perform?</p> <p><input type="checkbox"/> Run a computer-human interface <input type="checkbox"/> Business logic or rule processing</p> <p><input type="checkbox"/> Data entry & validation <input type="checkbox"/> Data retrieval & presentation</p> <p><input type="checkbox"/> Device/equipment interface <input type="checkbox"/> Terminal emulation</p> <p><input type="checkbox"/> Web/HTML browser <input type="checkbox"/> Web public interface</p> <p>Other (specify): _____</p>	<p>Almost all client/server or distributed software has external users. These are either people, or other computer systems. Typically in a client/server or distributed software system, the primary role of some computers is to provide the interface to external users.</p> <p>For this question, the word 'data' is used in the broadest sense of all/any types digital information.</p>
<p>89. Which of the services below do the host/server computer(s) provide to the software application/product?</p> <p><input type="checkbox"/> Database server <input type="checkbox"/> File &/or print server</p> <p><input type="checkbox"/> FTP server <input type="checkbox"/> HTML/web server</p> <p><input type="checkbox"/> Mail server <input type="checkbox"/> Messaging server</p> <p><input type="checkbox"/> Multi-user legacy application <input type="checkbox"/> Object/component server</p> <p><input type="checkbox"/> Security/authentication Other (specify): _____</p>	<p>In client/server or distributed software applications, one or more computers provide services to software operating on other computers. There can be multiple services running on one, or multiple, computer(s).</p> <p>Multi-user legacy application: an application originally designed to support multiple human users directly from a server/host computer.</p> <p>Object/component server: e.g. CORBA-compliant server or Microsoft Transaction Server.</p>
<p>90. If this project made no reuse of previous software development work, tick: <input type="checkbox"/> none</p> <p>If this project made any reuse of software development work products created prior to the project, what form did this reuse take? (If an enhancement project, the concept of reuse excludes the existing software that the project is enhancing.)</p> <p><input type="checkbox"/> Requirements/Functional Specification content <input type="checkbox"/> Design/Technical Specification content</p> <p><input type="checkbox"/> Purchased software components/libraries <input type="checkbox"/> In-house software components/libraries</p> <p><input type="checkbox"/> Purchased software framework or set of foundation classes</p> <p><input type="checkbox"/> In-house software framework or foundation classes</p> <p>Other (specify): _____</p>	<p>Promoters of reuse claim that it improves development productivity.</p> <p>Purchased components: Collection of source code/objects (or compiled objects) purchased separately from the programming languages used.</p> <p>In-house components / libraries: Collection of source code/objects formed and maintained by the development organisation itself.</p> <p>Purchased framework/foundation: Extensive set of software classes designed to be the foundation of a product and purchased separately from the programming language.</p>
<p>91. If there was reuse of software development work products on this project, what was the amount of functionality provided by reused work products (if it was measured)?</p> <p>Size: _____ Unit/Method: _____</p> <p>Other Method: _____</p>	<p>Software development work products include software components, libraries or frameworks.</p> <p>The 'amount of functionality' is measured using Function Points etc.</p>

F. IFPUG or NESMA Project Function Points

These forms are for recording an IFPUG or NESMA or FiSMA Function Point Analysis measurement of project software size. Other questionnaires provide forms for recording measurements with other methods. Questions 91 – 95 below gather data for a project developing a new software application/product, or redeveloping existing software with different technology. Questions 96 – 103 on the following pages gather data for a project enhancing an existing software product.

NEW DEVELOPMENT OR REDEVELOPMENT SOFTWARE SIZE

<p>92. Which function point counting standards were applied?</p> <p>IFPUG or NESMA? _____</p> <p>Version: _____ Specific customisation of rules? No <input type="checkbox"/></p> <p>Yes <input type="checkbox"/> Description of customisation: _____</p>	<p>This is essential to validating the count. Please specify the version used.</p> <p>Some organisations extend the standard with specific rules or guidelines.</p>
<p>93. What was the approach used to determine the project's functional size?</p> <p><input type="checkbox"/> Followed the IFPUG/NESMA Counting Practices Manual</p> <p><input type="checkbox"/> Followed the IFPUG/NESMA Counting Practices Manual but didn't individually assess each Transactional and Data Function Type's complexity</p> <p><input type="checkbox"/> Estimated from software components</p> <p><input type="checkbox"/> Backfired from source lines of code</p> <p><input type="checkbox"/> Other (specify): _____</p>	<p>This information is also essential for validating the count and is utilised in determining information on industry average complexities for Transactional and Data Function Types.</p> <p>Examples of software components used for estimation are screens, reports, modules etc.</p>

94. Unadjusted Function Points

	Low	Average	High	
External Inputs	_____ * 3 = _____	_____ * 4 = _____	_____ * 6 = _____	_____
External Outputs	_____ * 4 = _____	_____ * 5 = _____	_____ * 7 = _____	_____
External Enquiries	_____ * 3 = _____	_____ * 4 = _____	_____ * 6 = _____	_____
Internal Logical Files	_____ * 7 = _____	_____ * 10 = _____	_____ * 15 = _____	_____
External Interface Files	_____ * 5 = _____	_____ * 7 = _____	_____ * 10 = _____	_____
	Total Unadjusted Function Points			_____

95. General Systems Characteristics

1. Data Communications	–	8. On-line Update	–
2. Distributed Data Processing	–	9. Complex Processing	–
3. Performance	–	10. Reusability	–
4. Heavily Used Configuration	–	11. Installation Ease	–
5. Transaction Rate	–	12. Operational Ease	–
6. On-line Data Entry	–	13. Multiple Sites	–
7. End user Efficiency	–	14. Facilitate Change	–
Value Adjustment Factor = $0.65 + (0.01 * TDI)$		Total Degree of Influence (TDI) = _____	
		Value Adjustment Factor (VAF) = _____	

96. Adjusted Function Points

Unadjusted Function Points * VAF = _____ Adjusted Function Points

ENHANCEMENT SOFTWARE SIZE

<p>97. Which function point counting standards were applied?</p> <p>IFPUG or NESMA? _____</p> <p>Version: _____ Specific customisation of rules? No <input type="checkbox"/></p> <p>Yes <input type="checkbox"/> Description of customisation: _____</p>	<p>This is essential to validating the count. Please specify the version used.</p> <p>Some organisations extend the standard with specific rules or guidelines.</p>
<p>98. What was the approach used to determine the project's functional size?</p> <p><input type="checkbox"/> Followed the IFPUG/NESMA Counting Practices Manual</p> <p><input type="checkbox"/> Followed the IFPUG/NESMA Counting Practices Manual but didn't individually assess each Transactional and Data Function Type's complexity</p> <p><input type="checkbox"/> Estimated from software components</p> <p><input type="checkbox"/> Backfired from source lines of code</p> <p><input type="checkbox"/> Other (specify): _____</p>	<p>This information is also essential for validating the count and is utilised in determining information on industry average complexities for Transactional and Data Function Types.</p> <p>Examples of software components used for estimation are screens, reports, modules etc.</p>
<p>99. What was the function point size of the software before the enhancement project?</p> <p>Unadjusted Function Points: _____</p> <p>Value Adjustment Factor: _____</p>	<p>The scope of the enhancement project relative to the size of the original application may have some influence on project performance.</p>

100. **Added Functionality - Unadjusted Function Points**

	Low	Average	High	
External Inputs	_____ * 3 = _____	_____ * 4 = _____	_____ * 6 = _____	_____
External Outputs	_____ * 4 = _____	_____ * 5 = _____	_____ * 7 = _____	_____
External Enquiries	_____ * 3 = _____	_____ * 4 = _____	_____ * 6 = _____	_____
Internal Logical Files	_____ * 7 = _____	_____ * 10 = _____	_____ * 15 = _____	_____
External Interface Files	_____ * 5 = _____	_____ * 7 = _____	_____ * 10 = _____	_____
	Total Unadjusted Function Points (UFP) for Added Functionality			_____

101. **Changed Functionality - Unadjusted Function Points** (measured after the changes made)

	Low	Average	High	
External Inputs	_____ * 3 = _____	_____ * 4 = _____	_____ * 6 = _____	_____
External Outputs	_____ * 4 = _____	_____ * 5 = _____	_____ * 7 = _____	_____
External Enquiries	_____ * 3 = _____	_____ * 4 = _____	_____ * 6 = _____	_____
Internal Logical Files	_____ * 7 = _____	_____ * 10 = _____	_____ * 15 = _____	_____
External Interface Files	_____ * 5 = _____	_____ * 7 = _____	_____ * 10 = _____	_____
	Total Unadjusted Function Points (UFP) for Changed Functionality			_____

102. **Deleted Functionality - Unadjusted Function Points** (measured before it was deleted)

	Low	Average	High	
External Inputs	_____ * 3 = _____	_____ * 4 = _____	_____ * 6 = _____	_____
External Outputs	_____ * 4 = _____	_____ * 5 = _____	_____ * 7 = _____	_____
External Enquiries	_____ * 3 = _____	_____ * 4 = _____	_____ * 6 = _____	_____
Internal Logical Files	_____ * 7 = _____	_____ * 10 = _____	_____ * 15 = _____	_____
External Interface Files	_____ * 5 = _____	_____ * 7 = _____	_____ * 10 = _____	_____
	Total Unadjusted Function Points (UFP) for Deleted Functionality			_____

Software Size in Unadjusted Function Points

$(UFP^{\text{added}} + UFP^{\text{changed}} + UFP^{\text{deleted}}) =$ _____ Unadjusted Function Points

103. General Systems Characteristics

	Before ^a	After ^b		Before ^a	After ^b
1. Data Communications	–	–	8. On-line Update	–	–
2. Distributed Data Processing	–	–	9. Complex Processing	–	–
3. Performance	–	–	10. Reusability	–	–
4. Heavily Used Configuration	–	–	11. Installation Ease	–	–
5. Transaction Rate	–	–	12. Operational Ease	–	–
6. On-line Data Entry	–	–	13. Multiple Sites	–	–
7. End user Efficiency	–	–	14. Facilitate Change	–	–
Value Adjustment Factor ^c = 0.65 + (0.01 * TDI)			Total Degree of Influence =	_____	_____
			Value Adjustment Factor ^c =	_____	_____

a. Before = before the enhancement project began; b. After = after the enhancements are completed

c. Value Adjustment Factor = apply the equation to each TDI to calculate before and after VAFs.

104. Software Size in Adjusted Function Points

$$([UFP^{added} + UFP^{changed}] * VAF^{after}) + (UFP^{deleted} * VAF^{before}) = \text{_____ Adjusted Function Points}$$

IFPUG OR NESMA COUNTING STYLE

There are some possible variations in approach with IFPUG or NESMA function points that can have a significant impact on the resultant count.

<p>105. How was the Value Adjustment Factor calculated?</p> <p><input type="checkbox"/> Not assessed and set the VAF = 1.00</p> <p><input type="checkbox"/> Not assessed and General Systems Characteristics defaulted to average (3), so VAF = 1.07</p> <p><input type="checkbox"/> General Systems Characteristics assessed and set, and the VAF calculated to the value above.</p> <p><input type="checkbox"/> Other (specify): _____</p>	<p>Many people no longer use the Value Adjustment Factor in IFPUG function point counting.</p>
<p>106. If IFPUG functional count approach has been used, tick:</p> <p><input type="checkbox"/> IFPUG approach used</p> <p>If NESMA functional count approach has been used, were NESMA enhancement adjustment factors were used?</p> <p><input type="checkbox"/> Yes</p> <p><input type="checkbox"/> No</p> <p><input type="checkbox"/> Don't Know</p> <p><input type="checkbox"/> Other (specify): _____</p>	<p>The major difference between NESMA and IFPUG approaches is the use of scaling factors for the functional count of enhancement projects.</p>

<p>107. How were drop-down lists of values that appear on screens or forms counted (eg. list of client types, payment types, states)?</p> <p><input type="checkbox"/> Ignored</p> <p><input type="checkbox"/> Count each occurrence as an enquiry</p> <p><input type="checkbox"/> Count all occurrences as one or more system-wide enquiries</p> <p><input type="checkbox"/> Count each occurrence as an additional File Type Referenced (FTR) for the parent logical transaction</p> <p><input type="checkbox"/> Other (specify): _____</p>	<p>Most software products with graphical user interfaces make use of drop-down lists and other pick lists.</p>
<p>108. How were reference code data and static list data counted?</p> <p><input type="checkbox"/> Count an Internal Logical File for each distinct set of code values or reference list</p> <p><input type="checkbox"/> Count one Internal Logical File for all the sets of code values and reference lists combined</p> <p><input type="checkbox"/> Other (specify): _____</p>	<p>Many software products make use of reference code data used to classify primary data, or to provide a static list of values for a pick list. Such data typically takes the form of code with description, or just a list of values.</p>
<p>109. Is there any other comment on functional count? _____</p>	<p>If there is any other comment on functional count you wish to make it can be provided here.</p>

CONTEXT OF THE FUNCTION POINT COUNT

<p>110. Date of the function point count: _____ (d-mmm-yy)</p>	
<p>111. After which of the following activities was this Function Point count performed?</p> <p><input type="checkbox"/> Planning <input type="checkbox"/> Specification <input type="checkbox"/> Design</p> <p><input type="checkbox"/> Build <input type="checkbox"/> Test <input type="checkbox"/> Implementation</p>	<p>This helps us with our data quality processes.</p>
<p>112. If this Function Point count was performed before the test and implement activities, does it accurately measure the implemented software?</p> <p><input type="checkbox"/> Yes (skip the next question) <input type="checkbox"/> No</p>	<p>If this function point count was performed before test and implement, there may have been significant functionality added to the software after this count.</p>
<p>113. If this function point count does not accurately measure the implemented software, what is the likely difference from the function point count to the implemented software?</p> <p>Increased by: <input type="checkbox"/> 0-10% <input type="checkbox"/> 11-20% <input type="checkbox"/> 21-50% <input type="checkbox"/> > 50%</p> <p>Decreased by: <input type="checkbox"/> 0-10% <input type="checkbox"/> 11-20% <input type="checkbox"/> 21-50% <input type="checkbox"/> > 50%</p>	<p>This provides essential information for accurate comparison of this project to others.</p>

<p>114. Does the functional size entered in section F Q92-104 match the functionality that was developed by the project effort entered in section D Q63-76?</p> <p><input type="checkbox"/> Yes <input type="checkbox"/> No</p> <p>If 'No' describe any additional functionality that was developed by the project: _____</p> <p>Describe any additional functionality that was delivered, but not developed, by the project: _____</p>	<p>Additional functionality may occur in software components or layers that were not addressed by the functional size measurement entered in 'Software size', for example the development of device drivers.</p> <p>Additional functionality may be delivered but not developed, for example purchased software.</p>
<p>115. Which of the following information sources were used for the function point count?</p> <p><input type="checkbox"/> Feasibility Study <input type="checkbox"/> Requirements Specification</p> <p><input type="checkbox"/> Functional Specification <input type="checkbox"/> User Interface Prototype</p> <p><input type="checkbox"/> Logical Data/ER Model <input type="checkbox"/> User Manual</p> <p><input type="checkbox"/> High-Level Design Spec. <input type="checkbox"/> Technical Design Spec.</p> <p><input type="checkbox"/> Report layouts <input type="checkbox"/> Physical use of the software</p> <p><input type="checkbox"/> Message Sequence Diag. <input type="checkbox"/> Use Cases</p> <p><input type="checkbox"/> None</p> <p>Other (specify): _____</p>	<p>A function point count requires information about the software product being measured.</p> <p>This question refers to what documents and other information sources were used for the count.</p>
<p>116. For counting purposes, what was the documentation quality?</p> <p><input type="checkbox"/> Low <input type="checkbox"/> Average <input type="checkbox"/> High</p>	<p>This assists our data quality process.</p>
<p>117. What technology was used to support the counting process?</p> <p><input type="checkbox"/> Manually counted and manually documented</p> <p><input type="checkbox"/> Manually counted & documented with a software tool e.g. a spreadsheet or specialist documentation tool</p> <p><input type="checkbox"/> Count automatically generated by a software tool e.g. a CASE tool</p> <p><input type="checkbox"/> Derived from a count of lines of code (i.e. Backfired)</p> <p><input type="checkbox"/> Other (specify): _____</p>	<p>Certain technologies used in function point counting can impact on the count's potential accuracy.</p>
<p>118. What was the involvement of business area experts, (end users, business analyst, or requirements analyst), in the count?</p> <p><input type="checkbox"/> Not at all <input type="checkbox"/> Provided documentation & initial interview</p> <p><input type="checkbox"/> Walked through documentation & answered questions</p> <p><input type="checkbox"/> Present throughout count <input type="checkbox"/> Reviewed count</p>	<p>Business area experts can help resolve issues or questions that the function point counter may have about the software application's functionality.</p>

<p>119. Did an independent reviewer check the count for completeness and accuracy?</p> <p><input type="checkbox"/> Not reviewed at all <input type="checkbox"/> Informal review only</p> <p><input type="checkbox"/> Formally reviewed & approved</p> <p><input type="checkbox"/> Reviewed & conditionally approved (i.e. errors to correct)</p> <p>Other (specify): _____</p>	<p>A review of a function point count is an effective method of improving its accuracy.</p> <p>Informal review: A second function point counter examined the count.</p> <p>Formally reviewed & approved: A second function point counter examined the count using a review procedure and/or checklist.</p>
---	--

EXPERIENCE OF THE FUNCTION POINT COUNTER

<p>120. What training had the function point counter under gone?</p> <p><input type="checkbox"/> Reading &/or mentoring <input type="checkbox"/> Course from in-house trainer</p> <p><input type="checkbox"/> Course from specialised function point trainer</p> <p><input type="checkbox"/> Course certified by FSM method's certification body</p> <p>Other (specify): _____</p>	<p>What training had the function point counter under gone in order to have sufficient expertise to perform the count?</p> <p>Course certified... For example, a training course on IFPUG function points that IFPUG has certified.</p>																								
<p>121. How many years experience in function point counting had the counter prior to this functional size measurement?</p> <p><input type="checkbox"/> Less than 6 months <input type="checkbox"/> 6 months to 2 years</p> <p><input type="checkbox"/> 2 to 5 years <input type="checkbox"/> More than 5 years</p>	<p>Accurate function point counting requires experience.</p>																								
<p>122. How often does the function point counter perform a count?</p> <p><input type="checkbox"/> Once a week <input type="checkbox"/> Once a month</p> <p><input type="checkbox"/> Several times in 6 months <input type="checkbox"/> Several times a year</p>	<p>The accuracy of a function point count is influenced by how often the function point counter performs a count.</p>																								
<p>123. Does the counter have any of the following certifications?</p> <table border="0" style="width: 100%;"> <thead> <tr> <th></th> <th style="text-align: center;">Yes</th> <th style="text-align: center;">Version</th> <th style="text-align: center;">Years certification held:</th> </tr> </thead> <tbody> <tr> <td>IFPUG</td> <td style="text-align: center;"><input type="checkbox"/></td> <td style="text-align: center;">_____</td> <td style="text-align: center;">_____</td> </tr> <tr> <td>NESMA</td> <td style="text-align: center;"><input type="checkbox"/></td> <td style="text-align: center;">_____</td> <td style="text-align: center;">_____</td> </tr> <tr> <td>MARK II</td> <td style="text-align: center;"><input type="checkbox"/></td> <td style="text-align: center;">_____</td> <td style="text-align: center;">_____</td> </tr> <tr> <td>COSMIC FFP</td> <td style="text-align: center;"><input type="checkbox"/></td> <td style="text-align: center;">_____</td> <td style="text-align: center;">_____</td> </tr> <tr> <td>Other</td> <td style="text-align: center;"><input type="checkbox"/></td> <td style="text-align: center;">_____</td> <td style="text-align: center;">_____</td> </tr> </tbody> </table> <p>Name for other: _____</p>		Yes	Version	Years certification held:	IFPUG	<input type="checkbox"/>	_____	_____	NESMA	<input type="checkbox"/>	_____	_____	MARK II	<input type="checkbox"/>	_____	_____	COSMIC FFP	<input type="checkbox"/>	_____	_____	Other	<input type="checkbox"/>	_____	_____	<p>A function point counter with certification can be expected to be more accurate than one without.</p>
	Yes	Version	Years certification held:																						
IFPUG	<input type="checkbox"/>	_____	_____																						
NESMA	<input type="checkbox"/>	_____	_____																						
MARK II	<input type="checkbox"/>	_____	_____																						
COSMIC FFP	<input type="checkbox"/>	_____	_____																						
Other	<input type="checkbox"/>	_____	_____																						
<p>124. Is the function point counter a member of a local metrics group (e.g. NESMA, ASMA, IFPUG etc), or working within a group of counters?</p> <p><input type="checkbox"/> Yes <input type="checkbox"/> No</p>	<p>This indicates how much support the function point counter may have to resolve issues and keep skills up to date.</p>																								

G. Project Completion

Upon project completion, data about the overall project can be collected.

<p>125. On what date did the software go into operation? _____ (d-mmm-yy)</p>	<p>Please enter dates as d-mmm-yy e.g. 5-Jul-01.</p>
<p>126. What was the total elapsed duration of the project? _____ (months)</p>	<p>The calendar period in months between the project start and end including any period of inactivity (i.e. end date minus start date).</p>
<p>127. If there was any time of total inactivity on the project, what was its duration? _____ (months)</p>	<p>Total of periods during the project's elapsed duration, in which no project activity took place by any person involved in the project.</p>
<p>128. Are there any factors that you think had a positive impact on the project performance or outcomes? _____ _____ _____ _____</p>	<p>This question assists us in identifying factors that may have a significant effect on project performance or outcomes so they may be considered in future research.</p>
<p>129. Are there any factors that you think had a negative impact on the project performance or outcomes? _____ _____ _____ _____</p>	<p>This question assists us in identifying factors that may have a significant effect on project performance or outcomes so they may be considered in future research.</p>
<p>130. What was the number of defects recorded during the first month of the software's operation (first 30-days after the date on which the software began operation)? Minor: _____ Major: _____ Extreme: _____ Or Total Defects: _____</p>	<p>Minor defect: does not make the software unusable in any way (e.g. a modification required to a report). Major defect: causes part of the software to become unusable. Extreme defect: failure causing the software to become totally unusable.</p>
<p>131. If available, please state the lines of code generated by this project. _____ What percentage of these lines of code are not program statements? _____%</p>	<p>Number of lines of code/source statements (non-comment) is a good indicator of software complexity, particularly for algorithmic or compile intensive software where function points are low but software is substantial.</p>

USER SATISFACTION SURVEY

132. This table is to gather the results of any user satisfaction survey that was performed at the completion of the project.

	Poorly Met	Mostly Met	Fully Met	Exceeded Expectations
Did the project meet stated objectives?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Did the software meet business requirements?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Quality expectations for the software?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Quality expectations for user documentation?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ease-of-use requirements for the software?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Was sufficient training or explanation given?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Schedule for planning & specification?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Schedule for design, build, test & implement?	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Other (specify): _____	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
133. What was the project role of the person(s) who completed the user satisfaction survey? <input type="checkbox"/> Project/Executive Sponsor <input type="checkbox"/> Business Manager <input type="checkbox"/> End Users <input type="checkbox"/> Customer's Project Manager Other (specify): _____	This helps identify whether the person's role in the project has an impact on the user satisfaction survey.			
134. When was the user satisfaction survey performed? _____ (d-mmm-yy)	Please enter dates as d-mmm-yy e.g. 5-Jul-01.			

PROJECT COSTS

135. This table is to gather the costs of various components of the project. **If you cannot provide costs broken down for each activity, please provide total costs.** If you cannot provide costs, please skip the remainder of this questionnaire.

		Enter costs for each activity						Or total for the project
		Plan	Specify	Design	Build	Test	Impl.	
Development Team Costs	In-house labour							
	Sub-contractors ¹							
	Tools							
	Hardware ²							
	Or Development Team Total Cost							
Customer/ End user Costs	In-house labour							
	Sub-contractors ¹							
	Package Licences ³							
	Hardware ²							
	Or Customer / End User Total Cost							
IT Operation Costs⁴	In-house labour							
	Sub-contractors ¹							
	Package Licences							
	Hardware ²							
	Or IT Operations Total Cost							
Or Project Total Cost								

1. If sub-contractor costs are not known as a separate total, include their costs as in-house labour.
2. Either hardware purchased specifically for the project, or hardware costs specifically assigned to the project.
3. If the Development Team purchased the packages and passed the licence costs on to the customer, please still record them as Customer/ End user Costs.
4. This is the cost for the IT Operations group that supports the end users.

COST VALIDATION

<p>136. In which currency are the above costs given?</p> <p>Currency: _____ Other: _____</p> <p>Are costs expressed in multiples of base currency unit? <input type="checkbox"/> No <input type="checkbox"/> Yes (in which case what is the multiple? _____)</p>	<p>Currency is essential for cost comparison. If currency is not in list enter in 'other'.</p> <p>Some currencies are usually expressed in multiples (e.g. Lira or Yen). If so, please indicate the multiple.</p>
<p>137. Are the above costs the cost to develop or price paid by customer/client?</p> <p><input type="checkbox"/> Cost to developer <input type="checkbox"/> Cost to customer/client <input type="checkbox"/> Combination (in which case please describe: _____)</p>	<p>It is recognised that some organisations collect costs in terms of the cost of development while others collect costs in terms of the cost to the client or customer.</p>
<p>138. What procedure was used to calculate the project costs?</p> <p><input type="checkbox"/> All costs were specifically recorded as they occurred. <input type="checkbox"/> All costs were derived from other data, such as effort data. <input type="checkbox"/> Combination of recorded and derived costs. Other (specify): _____</p>	<p>This supports our data quality processes.</p>
<p>139. If the costs were calculated as a combination of recorded and derived, or by another method, please describe how costs were calculated?</p> <p>_____</p> <p>_____</p> <p>_____</p>	<p>This supports our data quality processes</p>
<p>140. Have all the costs been included in the data supplied?</p> <p><input type="checkbox"/> Yes (skip the next question) <input type="checkbox"/> No</p>	<p>For example, do the figures include project-specific training, consumables, and initial planning effort?</p>
<p>141. If no to the question above, what do you estimate the uncollected cost data to be?</p> <p><input type="checkbox"/> Less than 5% of recorded <input type="checkbox"/> 5 – 10% of recorded effort Other (specify): _____ <input type="checkbox"/> Unable to estimate</p>	<p>Uncollected cost data makes a project appear better value than it really was, which typically results in unrealistic future expectations.</p>

Thank You