

## Roteiro SERPRO de Métricas para Contratos de Software

### Histórico de Versões

<b>Data</b>	<b>Versão</b>	<b>Descrição</b>	<b>Autor</b>	<b>Revisor</b>	<b>Aprovado por</b>
30/04/2010	1.0	Roteiro Corporativo de Métricas para Contratos de Sistemas	Claudia Hazan		

# Sumário

<b>1. INTRODUÇÃO.....</b>	<b>4</b>
<b>2. OBJETIVO.....</b>	<b>5</b>
<b>3. ESTIMATIVAS DE PROJETOS DE SOFTWARE.....</b>	<b>6</b>
<b>3.1 CONTAGEM ESTIMATIVA DE PONTOS DE FUNÇÃO (CEPF).....</b>	<b>9</b>
<b>3.2 ESTIMATIVA DE ESFORÇO DE PROJETOS DE SOFTWARE.....</b>	<b>14</b>
3.2.1 <i>Distribuição de Esforço por Fase do Projeto.....</i>	<i>16</i>
<b>3.3 ESTIMATIVA DE PRAZO DE PROJETOS DE SOFTWARE.....</b>	<b>16</b>
3.3.1 <i>Alocação de Equipe ao Projeto.....</i>	<i>19</i>
<b>3.4. MÉTODO PARA ESTIMATIVA DE CUSTO.....</b>	<b>19</b>
<b>3.5 ESTIMATIVA DE RECURSOS COMPUTACIONAIS .....</b>	<b>20</b>
<b>4. CONTAGEM DE PONTOS DE FUNÇÃO DE PROJETOS DE MANUTENÇÃO.....</b>	<b>21</b>
<b>4.1 PROJETO DE MELHORIA .....</b>	<b>21</b>
<b>4.2 MANUTENÇÃO CORRETIVA.....</b>	<b>23</b>
<b>4.3 MANUTENÇÃO COSMÉTICA.....</b>	<b>25</b>
<b>4.4 MANUTENÇÃO ADAPTATIVA EM REQUISITOS NÃO FUNCIONAIS.....</b>	<b>25</b>
4.4.1 <i>Redesenvolvimento de Projetos em outra Plataforma.....</i>	<i>26</i>
4.4.2 <i>Atualização de Plataforma.....</i>	<i>26</i>
4.4.3 <i>Adequação de Funcionalidades às Mudanças de Negócio.....</i>	<i>27</i>
<b>4.5 APURAÇÃO ESPECIAL.....</b>	<b>29</b>
4.5.1 <i>Apuração Especial – Base de Dados.....</i>	<i>29</i>
4.5.2 <i>Apuração Especial – Geração de Relatórios.....</i>	<i>30</i>
<b>4.6 MANUTENÇÃO EM PÁGINAS ESTÁTICAS DE INTRANET, INTERNET OU PORTAL.....</b>	<b>30</b>
<b>4.7 MANUTENÇÃO DE DOCUMENTAÇÃO DE SISTEMAS LEGADOS.....</b>	<b>32</b>
<b>4.9 FATOR DE CRITICIDADE DE SOLICITAÇÃO DE SERVIÇO.....</b>	<b>33</b>
<b>4.10 PONTOS DE FUNÇÃO DE TESTE.....</b>	<b>33</b>
<b>5. ATIVIDADES SEM CONTAGEM DE PONTOS DE FUNÇÃO.....</b>	<b>34</b>
<b>6. CONSIDERAÇÕES PARA CONTAGEM DE PONTOS DE FUNÇÃO .....</b>	<b>37</b>
<b>6.1 CONSIDERAÇÕES SOBRE MUDANÇA DE REQUISITOS.....</b>	<b>37</b>
<b>6.2 CONSIDERAÇÕES SOBRE PROJETOS CANCELADOS.....</b>	<b>38</b>
<b>6.3 CONSIDERAÇÕES SOBRE REDUÇÃO DE CRONOGRAMA.....</b>	<b>39</b>
<b>6.4 MÉTRICAS PARA ATIVIDADES DE NEGÓCIO.....</b>	<b>40</b>
<b>7. CONTAGEM DE PONTOS DE FUNÇÃO COM MÚLTIPLAS MÍDIAS.....</b>	<b>42</b>
<b>7.1 CENÁRIO 1: MESMOS DADOS APRESENTADOS EM TELA E IMPRESSOS.....</b>	<b>44</b>
<b>7.2 CENÁRIO 2: MESMOS DADOS DE SAÍDA COMO DADOS EM ARQUIVO E RELATÓRIO IMPRESSO.....</b>	<b>44</b>
<b>7.3 CENÁRIO 3: MESMOS DADOS DE ENTRADA BATCH E ON-LINE.....</b>	<b>45</b>
<b>7.4 CENÁRIO 4: MÚLTIPLOS CANAIS DE ENTREGA DA MESMA FUNCIONALIDADE.....</b>	<b>45</b>
<b>7.5 CENÁRIO 5: RELATÓRIOS EM MÚLTIPLOS FORMATOS.....</b>	<b>45</b>
<b>8. PROCESSO DE REVISÃO DO GUIA DE CONTAGEM.....</b>	<b>47</b>
<b>8.1 REVISÃO PARA CORREÇÃO DE INCONSISTÊNCIAS E SITUAÇÕES NÃO PREVISTAS.....</b>	<b>47</b>
<b>8.2 REVISÃO PARA ADOÇÃO DE NOVAS VERSÕES DO CPM.....</b>	<b>47</b>
<b>9. CONCLUSÃO.....</b>	<b>48</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>49</b>
<b>ANEXO I: DOCUMENTO DE REQUISITOS DE PROJETOS DE MANUTENÇÃO PEQUENOS (&lt; 100 PFS)</b>	<b>51</b>
<b>ANEXO II: MODELO DE DOCUMENTO DE CONTAGEM DE PONTOS DE FUNÇÃO PARA PROJETOS DE MANUTENÇÕES PEQUENOS (&lt; 100 PFS).....</b>	<b>57</b>



## 1. Introdução

O SERPRO, assim como muitas empresas governamentais brasileiras têm utilizado a métrica de Pontos de Função (PF) nas estimativas e dimensionamento de tamanho funcional de projetos de software, devido aos diversos benefícios de utilização da métrica (independência da solução tecnológica utilizada) e às recomendações dos órgãos de controle do governo brasileiro.

O manual de práticas de contagem de Pontos de Função (CPM 4.3) [IFPUG, 2010] publicado pelo *International Function Point Users Group* (IFPUG) define as regras de contagem de Pontos de Função. No entanto, a contagem de Pontos de Função é baseada no projeto lógico da aplicação (logical design) e nas fases iniciais do ciclo de vida do software, o documento de requisitos para a estimativa e elaboração do plano do projeto é um documento inicial de requisitos, por exemplo: Documento de Visão, Formalização Simples de Requisitos ou algum outro tipo de especificação elaborada pelo analista de negócios. Assim, torna-se importante o estabelecimento de métodos para estimar o tamanho funcional dos projetos de software nas fases iniciais do ciclo de vida.

Outro ponto a ser destacado é a importância da definição de métodos para geração de estimativas de prazo, esforço, custo e recursos computacionais dos projetos de software da empresa, visando melhorar o gerenciamento dos projetos.

Além disso, é importante ressaltar que a métrica de Pontos de Função foi concebida como uma medida de tamanho funcional para projetos de desenvolvimento e de manutenção evolutiva de software. No entanto, os projetos de software não estão limitados a projetos de desenvolvimento e projetos de manutenção evolutiva ou Melhoria (*enhancement*). Assim, torna-se essencial a definição de métodos para dimensionar o tamanho de projetos de manutenção (*maintenance*) baseados em Pontos de Função para que estes projetos possam ser avaliados e gerenciados com base em uma métrica. Observe que a Instrução Normativa 04, publicada pela SLTI/ MPOG, preconiza a utilização de métricas em contratos de software. Os Acórdãos do Tribunal de Contas da União (TCU) recomendam a utilização da métrica Pontos de Função Não Ajustados. A versão 4.3 do CPM também reconhece o PF Não Ajustado como método padrão do IFPUG.

## 2. Objetivo

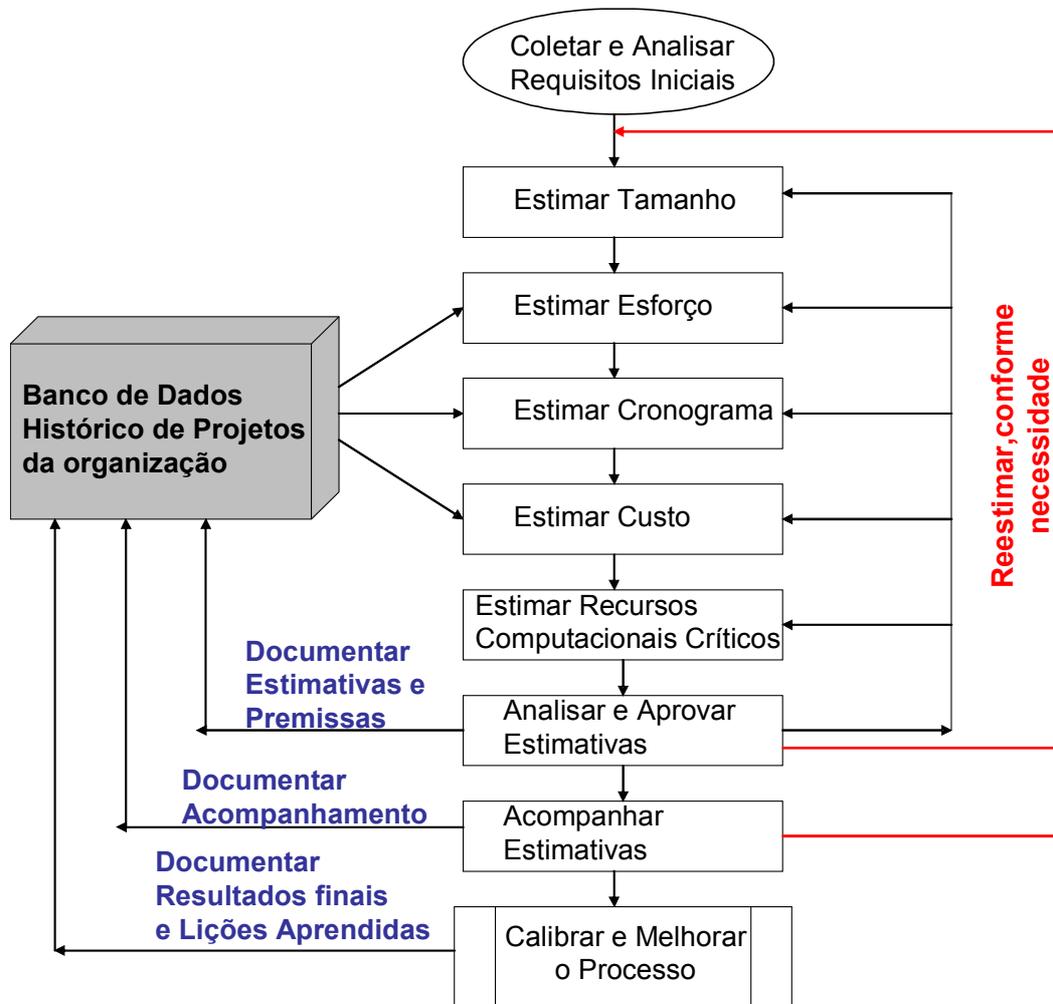
Este documento tem como propósito apresentar um roteiro contagem de Pontos de Função aderente ao Manual de Práticas de Contagem (CPM 4.3) e definir os tipos de projetos de manutenção e uma sistemática para dimensionar o tamanho de tais projetos, utilizando a métrica Pontos de Função. Além da contagem de Pontos de Função, este roteiro apresenta um processo de estimativas com base na métrica Pontos de Função, aderente ao modelo CMMI.

Este documento encontra-se organizado da seguinte maneira: O Capítulo 1 apresenta a motivação para a elaboração do documento; O Capítulo 2 mostra os objetivos aos quais este documento se propõe e a organização deste documento; O Capítulo 3 define o processo de estimativas de projetos de software integrado ao processo, indicando o momento de realização das estimativas e as análises a serem realizadas; O Capítulo 4 apresenta diretrizes para a estimativa e a contagem de Pontos de Função de todos os tipos de projetos de manutenção de software; O Capítulo 5 descreve as atividades associadas ao processo de desenvolvimento de software sem contagem de Pontos de Função; O Capítulo 6 apresenta algumas considerações importantes sobre utilização de métricas para dimensionar as mudanças de requisitos, redução de cronograma e atividades de negócio; O Capítulo 7 define um Guia para contagem de Múltiplas Mídias; O Capítulo 8 apresenta o processo de revisão deste guia de contagem; Finalmente, o Capítulo 9 conclui o documento apresentando sugestões para trabalhos futuros.

### 3. Estimativas de Projetos de Software

Este Capítulo tem como propósito descrever um processo de estimativas de projetos de software aderente ao CMMI. Nesse contexto, são apresentados: o método Contagem Estimativa de Pontos de Função (CEPF) para estimar o tamanho dos projetos de software em PF, o modelo simplificado de estimativas para estimar o esforço dos projetos em homem\_hora (HH), a fórmula de Capers Jones para estimar os prazos dos projetos.

A **Figura 1** ilustra um processo de Estimativas de Projetos de Software aderente à área de processo de Planejamento do Projeto do nível 2 do CMMI. Este processo é descrito em linhas gerais nos parágrafos seguintes.



**Figura 1: Processo de Estimativas de Projetos de Software [Hazan, 2008]**

O principal insumo (artefato de entrada) para um processo de estimativas é o documento de requisitos. Como as estimativas devem ser realizadas no início do processo de desenvolvimento de software, então, o artefato utilizado é um documento inicial de requisitos, por exemplo: Documento de Visão ou Formalização Simples de Requisitos. O estimador deve analisar os requisitos para garantir a qualidade e então estimar o tamanho do projeto de software. O próximo passo é a derivação das estimativas de esforço, prazo (cronograma), custo (orçamento) com base na estimativa de tamanho e nos dados históricos de projetos concluídos da organização, assim como o estabelecimento da estimativa de recursos computacionais críticos e dos recursos da equipe a ser alocada ao projeto. Neste ponto, as principais estimativas foram geradas e precisam ser documentadas. As premissas e suposições utilizadas na geração das estimativas, dentre outras: complexidade do projeto, plataforma de desenvolvimento, tipo do projeto, percentual de evolução de requisitos, também devem ser documentadas [Hazan, 2008].

A realização das estimativas por um analista de métricas que não atue na equipe do projeto, constitui uma prática recomendada. O analista de métricas deve analisar também a consistência da documentação utilizada na estimativa. No decorrer do processo de desenvolvimento, as estimativas devem ser acompanhadas conforme o refinamento dos requisitos. O projeto deve ser reestimado após a fase de requisitos, quando for gerada a especificação de casos de uso, e sempre ocorrerem mudanças significativas nos requisitos funcionais ou não funcionais. Quando o projeto é concluído, deve-se aferir e documentar o tamanho, prazo, custo, esforço e recursos realizados, assim como outros atributos relevantes do projeto, visando a coleta de dados para a melhoria do processo de estimativas. As lições aprendidas também devem ser documentadas [Hazan, 2008].

Portanto, para os contratos de projetos de software, baseados na métrica Pontos de Função, as estimativas devem ser realizadas em três marcos do processo de desenvolvimento de software, a saber:

- **Estimativa inicial:** realizada após o fechamento do escopo do projeto. Geralmente, é baseada em um documento inicial de requisitos, por exemplo Documento de Visão. Constitui uma boa prática a previsão de evolução de requisitos, especialmente em projetos de desenvolvimento de médio ou grande porte.

Nessa etapa é importante destacar os seguintes conceitos na área de estimativas: Uma **Estimativa** é obtida por meio de uma atividade técnica, utilizando métodos de estimativas. Não deve sofrer interferências políticas. A **Meta** é um desejo, em função de necessidades de negócio, estabelecida politicamente. Um **Compromisso** é um acordo da gerência com as equipes técnicas para alcançar uma meta [Parthasarathy,2007]. Em um cenário ideal os resultados da estimativa atendem as metas de negócio. Quando este cenário não é real, é fundamental a redução de escopo do projeto, de modo que a meta se adapte aos resultados da estimativa.

- **Contagem de Pontos de Função de Referência:** realizada após o aceite dos requisitos. Geralmente, leva em consideração a Especificação dos Casos de Uso e Regras de Negócio da aplicação.
- **Contagem de Pontos de Função Final:** realizada após a homologação da aplicação. Esta contagem leva em consideração as funcionalidades efetivamente entregues para o usuário pela aplicação.

Para fins de faturamento, que é realizado durante o desenvolvimento, deve-se considerar a Contagem de Referência e posteriormente considerar os ajustes no faturamento após a Contagem Final. É importante ressaltar que as mudanças de requisitos também serão consideradas no tamanho projeto a ser faturado, conforme descrito na seção 6.2. Além disso, se estas mudanças forem significativas, maiores que a evolução de requisitos (scope creep) prevista na estimativa inicial, o prazo do projeto deve ser reestimado. Toda mudança de requisito deve passar por uma análise de impacto do SERPRO e ser aprovada pelo cliente.

As seções seguintes apresentam os métodos de estimativas de tamanho prazo, custo e esforço utilizados nos projetos de software do SERPRO.

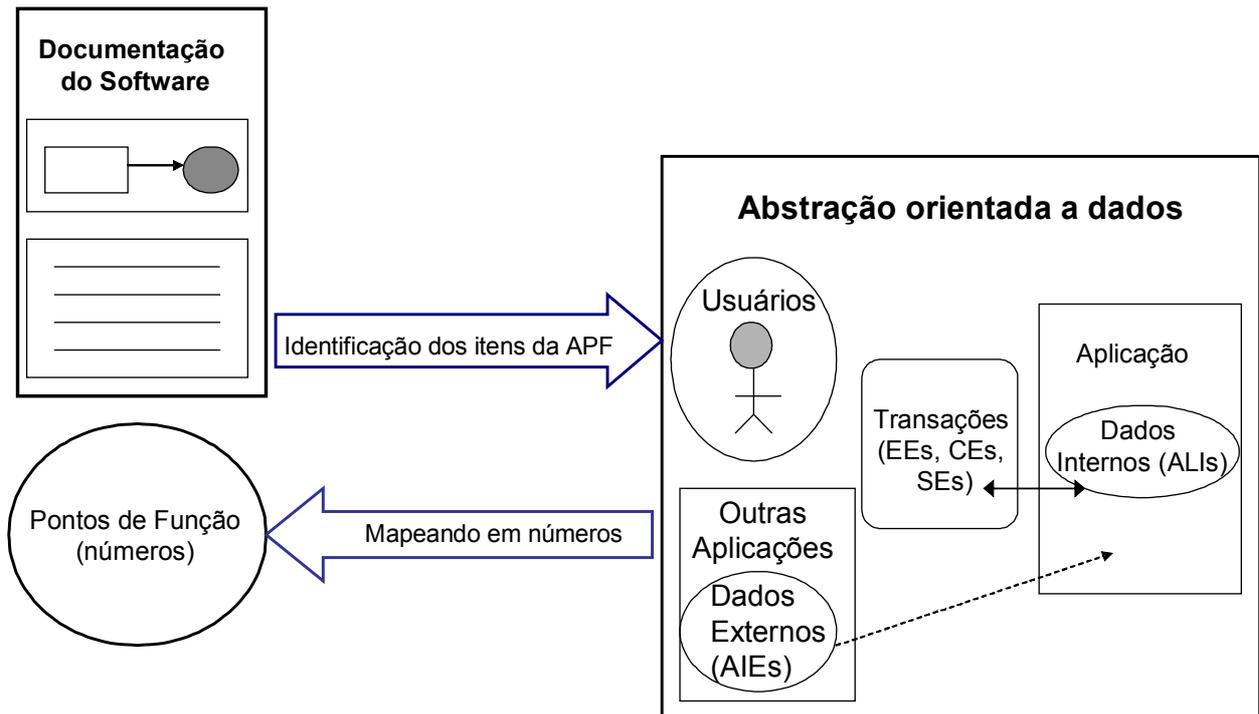
### 3.1 Contagem Estimativa de Pontos de Função (CEPF)

Antes de definir o método de estimativas – Contagem Estimativa de Pontos de Função (CEPF), é importante destacar que “estimar significa utilizar o mínimo de tempo e esforço para se obter um valor aproximado dos Pontos de Função do projeto de software investigado” [Meli, 1999]. Assim, para evitar confusão, é recomendável sempre fazer uma distinção entre os termos e conceitos: Contagem de Pontos de Função e Estimativa de Pontos de Função.

- **Contagem de Pontos de Função:** significa medir o tamanho do software por meio do uso das regras de contagem do IFPUG [IFPUG, 2010];
- **Estimativa de Pontos de Função:** significa fornecer uma avaliação aproximada do tamanho de um software utilizando métodos diferentes da Contagem de Pontos de Função do IFPUG.

O método CEPF visa aferir o tamanho em PF de maneira simplificada, com base no conhecimento dos requisitos iniciais do projeto. Inicialmente, os requisitos funcionais iniciais documentados nas propostas comerciais, nos documentos de visão, formalização simples de requisitos ou em qualquer especificação inicial do sistema do usuário são mapeados nos tipos funcionais da Análise de Pontos de Função: Arquivo Lógico Interno (ALI), Arquivo de Interface Externa (AIE), Entrada Externa (EE), Consulta Externa (CE) e Saída Externa (SE). Posteriormente, os Pontos de Função são associados a cada função identificada, baseando-se nas tabelas de complexidade e de contribuição funcional do CPM (Figura 2).

O estimador deve realizar uma leitura no documento inicial de requisitos, buscando informações relevantes para a identificação de processos elementares. O processo elementar é definido como a menor unidade de atividade significativa para o usuário. O processo elementar deve ser completo em si mesmo, independente e deixar a aplicação em um estado consistente [IFPUG, 2010]. Em outras palavras, os processos elementares são funções transacionais independentes, isto é, funções seqüenciais pertencem a um mesmo processo elementar e funções independentes constituem processos elementares diferentes.



**Figura 2: Modelo Lógico da Análise de Pontos de Função**

Uma vez identificado o processo elementar, o estimador deve buscar o entendimento deste para classificá-lo em Entrada Externa, Consulta Externa ou Saída Externa. Adicionalmente, o estimador deve descobrir os dados associados ao processo elementar, visando a determinação da complexidade funcional da função identificada. Caso não seja possível a identificação da complexidade da funcionalidade em questão, recomenda-se a utilização da complexidade Média. Na análise do processo elementar também são identificados, os grupos de dados lógicos da aplicação, que são classificados como Arquivos Lógicos Internos ou Arquivos de Interface Externa. Caso não seja possível a identificação da complexidade da função de dados em questão, recomenda-se a utilização da complexidade Simples. É importante ressaltar que se o estimador identificar mais de um Registro Lógico no Arquivo Lógico Interno, recomenda-se utilizar a complexidade Média.

A seguir são apresentadas dicas para ajudar no mapeamento dos requisitos funcionais da aplicação nos tipos funcionais da APF. As necessidades e funcionalidades especificadas para o projeto, contidas no documento inicial de requisitos, devem ser enquadradas em uma das seguintes tabelas:

**Tabela 1** - Contagem dos Arquivos Lógicos Internos (ALIs): Banco de Dados Lógico da Aplicação (tabelas e arquivos mantidos pela aplicação).

**Considerações:** Identifique os grupos de dados lógicos de aplicação nos modelos de dados ou diagrama de classes ou a partir dos requisitos funcionais, descritos nos documentos de requisitos (Documento de Visão, Relação de Casos de Uso, etc.). Não considere arquivos físicos, arquivos de índices, arquivos de trabalho e tabelas de relacionamento sem atributos próprios (tabelas que existem para quebrar o relacionamento nxn e apenas transportam as chaves estrangeiras). As entidades fracas também não são consideradas um ALI.

Se possível, tente descobrir os atributos lógicos, campos reconhecidos pelo usuário, e subgrupos de dados existentes para obter a complexidade funcional, segundo as regras de contagem do CPM. Caso não seja possível, a experiência tem mostrado que a maioria dos ALIs dos sistemas são de complexidade **Simples**.

<b>Nº ALIs Simples:</b>	<b>X 7 PF</b>
<b>Nº ALIs Médio:</b>	<b>X 10 PF</b>
<b>Nº ALIs Complexo:</b>	<b>X 15 PF</b>
<b>Total PF da Tabela 1:</b>	

**Tabela 1: Identificação dos Arquivos Lógicos Internos da Aplicação**

**Tabela 2** - Contagem de Arquivos de Interface Externa (AIEs): Banco de Dados de outras Aplicações, **apenas referenciados** pela aplicação que está sendo estimada (tabelas e arquivos mantidos por outra aplicação).

**Considerações:** Identifique os grupos de dados lógicos de outras aplicações referenciados pela aplicação que está sendo estimada. Frequentemente, o referenciamento de dados ocorre para a validação de informações em cadastros ou consultas. Algumas vezes, relatórios ou consultas referenciam dados externos de outras aplicações, também considerados AIEs. Não são considerados arquivos físicos, arquivos de índice, arquivos de trabalho, tabelas de relacionamento sem atributos próprios e entidades fracas.

Geralmente, os AIEs dos sistemas possuem a classificação de complexidade **Simples**. Porque, são considerados para a determinação da complexidade funcional do AIE apenas os atributos referenciados pela aplicação que está sendo contada.

<b>Nº AIEs Simples:</b>	<b>X 5 PF</b>
<b>Nº AIEs Médio:</b>	<b>X 7PF</b>
<b>Nº AIEs Complexo:</b>	<b>X 10 PF</b>
<b>Total PF da Tabela 2:</b>	

**Tabela 2: Identificação dos Arquivos de Interface Externa da Aplicação**

**Tabela 3** - Contagem de Entradas Externas (EEs): Funcionalidades que mantêm os Arquivos Lógicos Internos (ALIs) ou alteram o comportamento da aplicação.

**Considerações:** Identifique as funcionalidades de manutenção de dados. Conte separadamente a inclusão, alteração e exclusão de dados, isto é, cada função independente de inclusão ou alteração ou exclusão deve ser contada separadamente. A aplicação possui funções de entrada de dados que alteram o comportamento dela, por exemplo: processamentos *batch*, ou processamento de informações de controle? Caso positivo, estas funções também devem ser identificadas como Entradas Externas.

Se você não possui conhecimento sobre o processo elementar (funcionalidade analisada), considere as Entrada Externa identificada com complexidade **Média**.

<b>Nº EEs Simples:</b>	<b>X 3 PF</b>
<b>Nº EEs Média:</b>	<b>X 4 PF</b>
<b>Nº EEs Complexa:</b>	<b>X 6 PF</b>
<b>Total PF da Tabela 3:</b>	

**Tabela 3: Identificação das Entradas Externas da Aplicação**

**Tabela 4** - Contagem de Consultas Externas (CEs): funcionalidades que apresentam informações para o usuário **sem** a utilização de cálculos ou algoritmos.

São os processos elementares do tipo “lê - imprime”, “lê - apresenta dados”, incluindo consultas, relatórios, geração de arquivos pdf, xls, *downloads*, entre outros.

**Considerações:** Você está desenvolvendo uma função para apresentar informações para o usuário: uma consulta, relatório, *browse*, *listbox*, *download*, geração de um arquivo, geração de arquivo psd, xls? Esta função **não** possui cálculos ou algoritmos para derivação dos dados referenciados nem altera um Arquivo Lógico Interno, nem muda o comportamento do sistema? Caso positivo, estas funções devem ser identificadas como Consultas Externas.

Se você não possui conhecimento sobre o processo elementar (funcionalidade analisada), considere as Consultas Externas com complexidade **Média**.

<b>Nº CEs Simples:</b>	<b>X 3 PF</b>
<b>Nº CEs Média:</b>	<b>X 4 PF</b>
<b>Nº CEs Complexa:</b>	<b>X 6 PF</b>
<b>Total PF da Tabela 4:</b>	

**Tabela 4: Identificação das Consultas Externas da Aplicação**

**Tabela 5** - Contagem de Saídas Externas (SEs): Funcionalidades que apresentam informações para o usuário **com** utilização de cálculos ou algoritmos para derivação de dados ou atualização de Arquivos Lógicos Internos ou mudança de comportamento da aplicação. São as consultas ou relatórios com totalização de dados, relatórios estatísticos, gráficos, geração de arquivos com atualização *log*, *downloads* com cálculo de percentual, entre outros.

**Considerações:** Você está desenvolvendo uma funcionalidade para apresentar informações para o usuário: uma consulta ou relatório com totalização de dados, etiquetas de código de barras, gráficos, relatórios estatísticos, *download* com percentual calculado, geração de arquivo com atualização de *log*? Caso positivo, estas funções devem ser identificadas como Saídas Externas. Observe que esta função *deve* ter cálculos ou algoritmos para processar os dados referenciados nos arquivos lógicos *ou* atualizar campos (normalmente indicadores) nos arquivos *ou* mudar o comportamento da aplicação.

Caso não haja conhecimento da aplicação de APF ou sobre o processo elementar (funcionalidade analisada), considere as Saídas Externas com complexidade **Média**.

<b>Nº SEs Simples:</b>	<b>X 4 PF</b>
<b>Nº SEs Média:</b>	<b>X 5 PF</b>
<b>Nº SEs Complexa:</b>	<b>X 7 PF</b>
<b>Total PF da Tabela 5:</b>	

**Tabela 5: Identificação dos Saídas Externas da Aplicação**

A Estimativa de tamanho do projeto em PFs deve ser gerada totalizando-se os PFs obtidos nas **Tabelas 1, 2, 3, 4, e 5**.

A fórmula de contagem ou de estimativa de Pontos de Função para Projetos de Desenvolvimento é a seguinte:

$$\text{PF\_Desenvolvimento} = \text{PF\_Não\_Ajustado} + \text{PF\_Conversão}$$

**Observação:** PF\_Conversão: Pontos de Função associados às funcionalidades de conversão de dados dos projetos. Exemplos de funções de conversão incluem: migração ou carga inicial de dados para popular as novas tabelas criadas no sistema e relatórios associados à migração de dados.

### 3.2 Estimativa de Esforço de Projetos de Software

Uma vez que o tamanho do projeto está estimado em Pontos de Função, o próximo passo é estimar o esforço de desenvolvimento projeto, bem como sua distribuição pelas fases do ciclo de vida do desenvolvimento do software. A Engenharia de Software possui vários modelos para estimar esforço de projetos de software, baseados em Pontos de Função, sendo o Modelo Simplificado de Estimativas [Vazquez, 2010] e o Modelo COCOMO II [Boehm, 2000] os mais utilizados. No SERPRO é adotado o modelo Simplificado de Estimativas. Futuramente, pretende-se utilizar também o COCOMO II.

O modelo simplificado de estimativas consiste em obter um índice de produtividade em horas/PF para o projeto específico em questão, e então multiplicar o tamanho em PF do Projeto pelo índice de produtividade, conforme a fórmula [Vazquez, 2010]:

$$\text{Esforço (horas)} = \text{Tamanho (PF)} \times \text{Índice de Produtividade (HH/PF)}$$

O índice de produtividade depende de diversos atributos dos projetos, dentre outros: plataforma tecnológica, complexidade do domínio, segurança, desempenho, usabilidade, tamanho do projeto, tipo de manutenção, desenvolvimento de componentes. Assim, com base em análise de dados históricos de projetos do SERPRO, Benchmarking e análise de literatura específica, foi definida uma Tabela de Produtividade do SERPRO para ser utilizada nas estimativas de esforço da empresa. Esta tabela é revisada com a periodicidade bimestral, considerando o *feedback* das equipes de desenvolvimento da empresa e novas análises de dados. A política de preços do SERPRO foi estabelecida com base na análise de dados históricos dos projetos da empresa.

Considerando um projeto de desenvolvimento de uma solução, além da construção do sistema, dimensionada por meio da métrica Pontos de Função, podem ser necessárias outras atividades de consultoria ou análise. A fórmula utilizada para o cálculo de esforço total de um projeto (EP) é a seguinte [SERPRO, 2008]:

$$EP = QHC + QHA + (QPF \times EPF)$$

Onde:

**QHC** = Quantidade de Horas Perfil Consultor

**QHA** = Quantidade de Horas Perfil Analista

**QPF** = Tamanho do Projeto em PF

**EPF** = Esforço para implementar um PF na plataforma em questão

### 3.2.1 Distribuição de Esforço por Fase do Projeto

O próximo passo é a definição da distribuição de esforço pelas macroatividades do projeto, visando definir o valor agregado ao projeto após cada fase do ciclo de vida (Tabela 5).

<b>Macro Atividades do Processo de Desenvolvimento de Software</b>	<b>Percentual de esforço (%)</b>
Engenharia de Requisitos	25%
Design, Arquitetura	15%
Implementação	40%
Testes	10%
Homologação	5%
Implantação	5%

**Tabela 5: Distribuição de Esforço por Macroatividades do Projeto**

A Tabela 5 é uma sugestão de distribuição de esforço em projetos típicos, no entanto, em se tratando um projeto com características específicas, estes percentuais devem ser alterados para o projeto em questão. Nesses casos, o estimador deve justificar, com observações no documento de estimativas, as premissas utilizadas para a alteração dos percentuais.

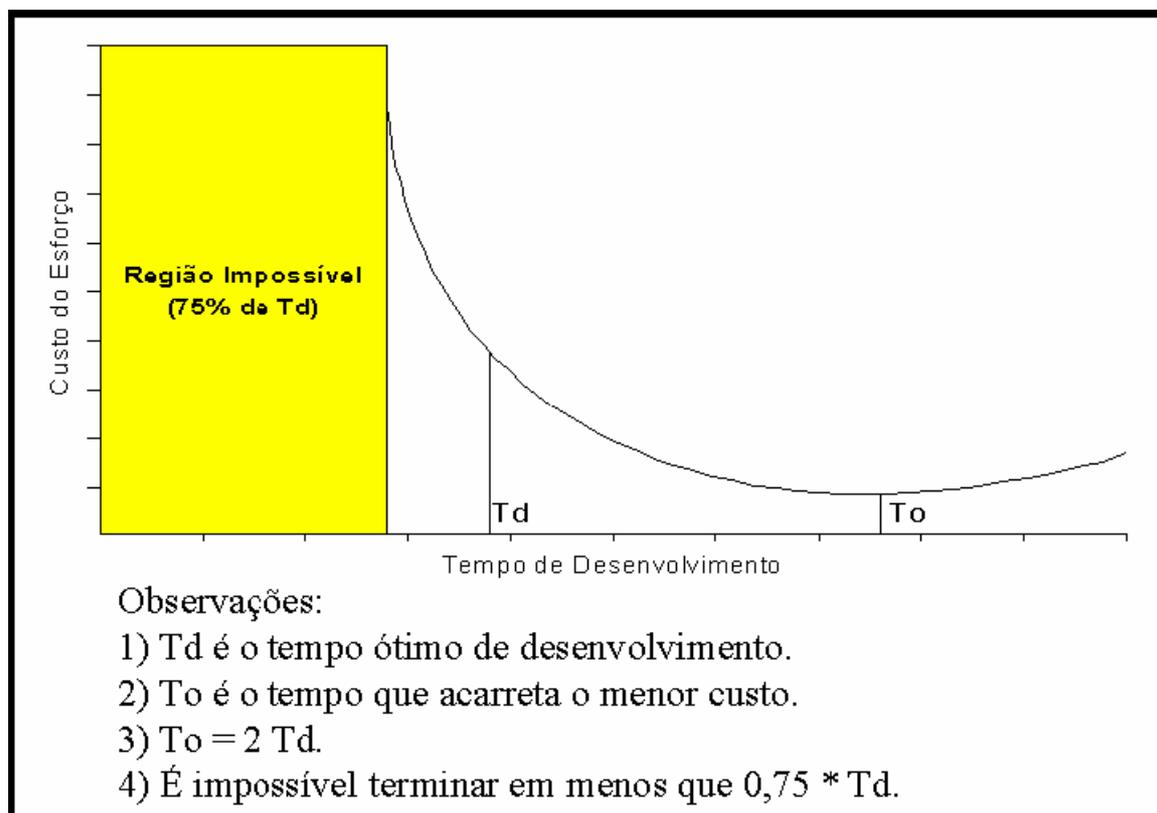
Os contratos estabelecidos com os clientes devem determinar o processo de desenvolvimento a ser seguido com percentual de esforço por fases.

### 3.3 Estimativa de Prazo de Projetos de Software

As estimativas de prazo não são lineares com o tamanho do projeto. O melhor tempo desenvolvimento, no qual há uma melhor relação custo x benefício de alocação de recursos e menor prazo de desenvolvimento, dado o tamanho de um projeto específico, é utilizado nas estimativas de prazo do SERPRO. Jones [Jones, 2007] propõe uma fórmula para o cálculo do melhor tempo de desenvolvimento, denominado Td e de Região Impossível (RI) de desenvolvimento (Figura 3).

Na Região Impossível (RI), a adição de mais recursos ao projeto não implicará em redução no prazo. Note que a curva (Figura 3) mostra que quanto menor o prazo almejado para a conclusão do projeto, maior será o esforço requerido e

consequentemente maior o custo do projeto. O aumento do esforço para reduzir o prazo acontece através da realização de horas-extras e da inclusão de pessoal adicional, gerando retrabalho. No entanto, a redução de prazo tem um limite, como demonstra a região impossível da Figura 3 .



**Figura 3: Relação entre a Estimativa de Prazo e de Esforço**

O método utilizado para estimar o prazo dos projetos ( $T_d$ ) é baseado na fórmula de Capers Jones [Jones, 2007]. Posteriormente, pretende-se implantar o modelo COCOMO II para obtenção de mais de uma estimativa de prazo para o projeto. A fórmula de Capers Jones estima o prazo, baseando-se no tamanho do projeto em Pontos de Função, da seguinte maneira:

$$T_d = V^t$$

Onde:

**Td:** prazo de desenvolvimento

**V:** tamanho do projeto em Pontos de Função

**t:** o expoente  $t$  é definido de acordo com a Tabela 6

<b>Tipo de Sistema</b>	<b>Expoente t</b>
Sistema Comum – Mainframe (desenvolvimento de sistema com alto grau de reuso ou manutenção evolutiva)	0,32 a 0,33
Sistema Comum – WEB ou Cliente Servidor	0,34 a 0,35
Sistema OO (se o projeto OO não for novidade para equipe, não tiver o desenvolvimento de componentes reusáveis, considerar sistema comum)	0,36
Sistema Cliente/Servidor (com alta complexidade arquitetural e integração com outros sistemas)	0,37
Sistemas Gerenciais complexos com muitas integrações, Datawarehousing, Geoprocessamento, Workflow	0,39
Software Básico, Frameworks, Sistemas Comerciais	0,40

**Tabela 7: Expoente t por tipo de Projeto**

É importante destacar que o método só funciona para projetos com mais de 100 PFs. Caso o projeto seja menor, o prazo deve ser obtido por meio de WBS. O prazo calculado considera todo o ciclo de vida do projeto, desde a fase de requisitos até a implantação. Assim, caso a estimativa tenha sido realizada ao final da fase de requisitos, descontar do prazo restante, o tempo gasto com a fase de requisitos.

Caso o cliente precise receber o projeto em um prazo menor que o Td calculado, recomenda-se propor um processo de desenvolvimento incremental, priorizando funcionalidades em cada iteração de acordo com a necessidade dele. Caso, ainda assim, a estimativa não atenda às necessidades do cliente, então pode-se reduzir o Td em até 25%, observando-se a Região Impossível. No entanto, quanto mais perto da Região Impossível, o esforço e o custo do projeto aumentam de maneira exponencial. Assim, de um modo geral, a redução de prazo de 10 % implica no aumento de esforço de 25%; a redução de prazo de 20% implica no aumento de esforço de 50% ; a redução de prazo de 25% implica em um aumento de esforço de 60%.

Não é recomendada a redução de prazo em mais de 20%.

Na seção seguinte é abordada a questão da distribuição de esforço e alocação de pessoas ao projeto em questão.

### 3.3.1 Alocação de Equipe ao Projeto

Na alocação de equipe, deve ser considerada a estimativa de prazo e a de esforço. A fórmula utilizada é a seguinte:

$$\text{Equipe} = \text{Esforço (HH)} / (21 \times \text{prod. diária} \times \text{Prazo})$$

Onde:

prazo = Td em meses

Prod. Diária = 6h/dia ou 7h/dia (recomenda-se considerar 6 horas/dia)

21 = dias úteis contidos em 1 mês

O tamanho da equipe é obtido em quantidade de recursos para o desenvolvimento do projeto, deve-se considerar percentuais de alocação. Por exemplo, suponha uma Equipe de 2,2 recursos. Esta equipe pode conter 5 pessoas, sendo que 4 pessoas com 50% de alocação e um líder de projeto com 20% de alocação ao projeto.

### 3.4. Método para Estimativa de Custo

A estimativa de custo do projeto deve levar em consideração o custo da mão de obra, considerando o esforço e o custo da hora de todos os profissionais envolvidos no desenvolvimento da solução de software. Além do custo da mão de obra, devem ser considerados outros custos, tais como: treinamento, consultoria, viagens, licenças de software, custos indiretos etc. Também devem ser considerados os custos dos recursos computacionais descritos na seção seguinte.

Sugere-se a seguinte fórmula para calcular o custo relativo a mão de obra para o desenvolvimento da solução (CP – Custo do Projeto).

$$\text{CP} = (\text{QHC} \times \text{VPC}) + (\text{QHA} \times \text{VPA}) + (\text{QPF} \times \text{EPF} \times \text{VPA}) + \text{Outros Custos}$$

Onde:

**QHC** = Quantidade de Horas Perfil Consultor

**VPC** = Valor da Hora do Perfil Consultor

**QHA** = Quantidade de Horas Perfil Analista

**VPA** = Valor da Hora do Perfil Analista

**QPF** = Tamanho do Projeto em PF

**EPF** = Esforço para implementar um Ponto de Função na plataforma em questão

Caso o contrato seja de preço fixo por Ponto de Função, então pode-se considerar o seguinte:

$$CP = (QHC \times VPC) + (QHA \times VPA) + (QPF \times VPF)$$

Onde:

**VPF** = Valor Unitário do PF para o projeto em questão - Identificado de acordo com a Tabela de Serviço Padrão do Sistema de Orçamento Técnico do SERPRO.

É importante destacar que como o esforço para a construção do PF é variável, o preço do PF também é variável de acordo com os requisitos não funcionais do projeto. A política de preços do SERPRO define o preço do Ponto de Função (**VPF**) para os diversos tipos de projetos da empresa.

### 3.5 Estimativa de Recursos Computacionais

A Estimativa de Recursos Computacionais também deve ser considerada, esta constitui um componente importante para as estimativas de custos dos projetos. Um recurso computacional é um hardware que se precisa adquirir; ou que se possui, mas precisa-se configurar. Exemplos de recursos computacionais incluem, dentre outros: espaço em disco para o sistema entrar em produção, um servidor específico para teste ou homologação do sistema. Devem ser registradas as seguintes informações associadas aos recursos computacionais críticos:

•**Nome do Recurso Computacional:** [considere exclusivamente hardware: micro, periférico, expansão de memória, área em disco, banda de rede, etc]

•**Descrição:**

•**Responsável pela Disponibilização:**

•**Data Limite:**

•**Parâmetros:** [características do recurso: quantidade, perfil, configuração, etc]

•**Tipo do Recurso:** [D: recurso para ambiente de Desenvolvimento; P: recurso para ambiente de Produção; H: recurso para ambiente de Homologação]

•**Custo (Opcional):** [Custo do recurso computacional. Não considerar custos de processamento ou custos operacionais de produção.]

Caso o projeto a ser desenvolvido não possua nenhum recurso computacional crítico, deve ser registrado no documento de estimativas que o projeto não possui nenhum recurso computacional crítico.

## 4. Contagem de Pontos de Função de Projetos de Manutenção

Esta seção tem como propósito descrever os diversos tipos de projetos de manutenção e mostrar uma solução para o seu dimensionamento em Pontos de Função, visto que o manual de práticas de contagem – CPM não contempla projetos de manutenção (*maintenance*) apenas o de Melhoria (*enhancement*).

Quanto à documentação de projetos de manutenção pequenos (menores que 100 PF), deve-se registrar a solicitação do cliente (demanda do SIGOP ou Solicitação do Serviço ou Solicitação de Mudança) e documentar os requisitos da aplicação impactada pela demanda, de forma detalhada, visando apoiar a contagem de Pontos de Função da demanda. É importante também documentar as estimativas e a contagem de Pontos de Função. O Anexo I apresenta um modelo para a documentação dos requisitos. O Anexo II apresenta um modelo para a documentação da Contagem de Pontos de Função.

### 4.1 Projeto de Melhoria

O Projeto de Melhoria (*enhancement*), também denominada de projeto de melhoria funcional, ou manutenção evolutiva, está associado às mudanças em requisitos funcionais da aplicação, ou seja, a inclusão de novas funcionalidades, alteração ou exclusão de funcionalidades em aplicações implantadas.

Segundo o padrão IEEE Std 1229 [IEEE 1229], esta manutenção seria um tipo de manutenção adaptativa, definida como: modificação de um produto de software concluído após a entrega para mantê-lo funcionando adequadamente em um ambiente com mudanças. O projeto de melhoria é considerado um tipo de projeto de manutenção adaptativa com mudanças em requisitos funcionais da aplicação, ou seja com funcionalidades incluídas, alteradas ou excluídas na aplicação, segundo o CPM 4.3.

Este documento separa o projeto de melhoria, quando as mudanças são associadas aos requisitos funcionais e a manutenção adaptativa quando as mudanças estão associadas aos requisitos não funcionais da aplicação.

Um projeto de melhoria consiste em demandas de criação de novas funcionalidades (grupos de dados ou processos elementares), demandas de exclusão de funcionalidades (grupos de dados ou processos elementares) e demandas de alteração de funcionalidades (grupos de dados ou processos elementares) em aplicações implantadas em produção.

Uma função de dados (Arquivo Lógico Interno ou Arquivo de Interface Externa) é considerada alterada, quando a alteração contemplar mudanças de item de dados, inclusão ou exclusão de item de dados. Ou mudança de tamanho (número de posições) ou tipo de campo (por exemplo: mudança de numérico ou alfanumérico), sendo que esta ocorre por mudança de regra de negócio do usuário.

Uma função transacional (Entrada Externa, Consulta Externa e Saída Externa) é considerada alterada, quando a alteração contemplar:

- Mudança de itens de dados em uma função existente;
- Mudança de arquivos referenciados;
- Mudança de lógica de processamento, segundo as ações das lógicas e processamento do CPM 43:

A **Lógica de Processamento** é definida como requisitos especificamente solicitados pelo usuário para completar um processo elementar. Esses requisitos devem incluir as seguintes ações:

- Validações são executadas
- Fórmulas matemáticas e cálculos são executados
- Valores equivalentes são convertidos
- Dados são filtrados e selecionados através da utilização de critérios
- Condições são analisadas para verificar quais são aplicáveis
- Um ou mais ALIs são atualizados
- Um ou mais ALIs e AIEs são referenciados
- Dados ou informações de controle são recuperados
- Dados derivados são criados através da transformação de dados existentes, para criar dados adicionais
- O comportamento do sistema é alterado
- Preparar e apresentar informações para fora da fronteira
- Receber dados ou informações de controle que entram pela fronteira da aplicação

- Dados são reordenados

A contagem ou estimativa de Pontos de Função de projetos de manutenção evolutiva deve seguir a fórmula de *enhancement* do CPM 4.3:

$$PF = (PF \text{ INCLUIDO} + PF \text{ ALTERADO} + PF\_CONVERSÃO + PF \text{ EXCLUIDO} )$$

#### Definições:

PF\_INCLUÍDO = Pontos de Função associados às novas funcionalidades que farão parte da aplicação.

PF\_ALTERADO = Pontos de Função associados às funcionalidades existentes na aplicação que serão alteradas no projeto de manutenção.

PF\_EXCLUÍDO = Pontos de Função associados às funcionalidades existentes na aplicação que serão excluídas no projeto de manutenção.

PF\_CONVERSÃO = Pontos de Função associados às funcionalidades de conversão de dados dos projetos de melhoria. Exemplos de funções de conversão incluem: migração ou carga inicial de dados para popular as novas tabelas criadas e relatórios associados à migração de dados.

## 4.2 Manutenção Corretiva

Mesmo com a execução de atividades de garantia da qualidade, o cliente pode identificar defeitos na aplicação entregue. A manutenção corretiva altera o software para correção de defeitos. Encontra-se nesta categoria, as demandas de correção de erros (bugs) em funcionalidades em sistemas em produção.

É importante destacar que as demandas de manutenção corretiva freqüentemente precisam ser atendidas com urgência. Assim, o grau de criticidade do projeto poderá trazer impacto nas estimativas de custo e esforço. O padrão IEEE [IEEE,1998] define um tipo de manutenção corretiva, denominado de Manutenção Emergencial como “manutenção corretiva não programada executada para manter o sistema em estado operacional”.

**Quando o sistema em produção for desenvolvido pelo SERPRO, a manutenção corretiva será do tipo Garantia, não sendo cobrada do cliente, desde que o prazo da solicitação seja inferior a 180 dias da data do recebimento da funcionalidade em questão.**

Nestes casos, a aferição do tamanho em Pontos de Função da funcionalidade ou das funcionalidades corrigidas considera 30% do PF\_Alterado, observando os conceitos do CPM 4.3, apresentados na seção 4.1.

$$PF = PF\_ALTERADO \times 0,30$$

Quando o sistema não for desenvolvido pelo SERPRO, esta manutenção deverá ser cobrada do cliente. A estimativa e dimensionamento de tamanho de projetos de manutenção corretiva em Pontos de Função devem levar em consideração a documentação do sistema disponível e os artefatos a serem mantidos. Seguem as fórmulas a serem consideradas:

**a) Aplicação sem documentação ou com documentação desatualizada ou documentação incompleta e sem redocumentação de requisitos**

Nestes casos, a aferição do tamanho em Pontos de Função da funcionalidade ou das funcionalidades corrigidas considera 70% do PF\_Alterado, observando os conceitos do CPM 4.3, apresentados na seção 4.1.

$$PF = PF\_ALTERADO \times 0,70$$

**b) Aplicação sem documentação ou com documentação desatualizada ou incompleta ou completa e com redocumentação de requisitos**

Nestes casos, a aferição do tamanho em Pontos de Função da funcionalidade ou das funcionalidades corrigidas considera 80% do PF\_Alterado, seguindo os conceitos do CPM 4.3, apresentados na seção 4.1.

Deve-se destacar que além da correção as funcionalidades em questão e da documentação do projeto de manutenção corretiva realizado, a documentação das funcionalidades deve ser atualizada.

$$PF = PF\_ALTERADO \times 0,80$$

### **c)Aplicação com documentação completa**

Nestes casos, a aferição do tamanho em Pontos de Função da funcionalidade ou das funcionalidades corrigidas considera 60% do PF\_Alterado, seguindo os conceitos do CPM 4.3, mostrados na seção 4.1. Deve-se ressaltar que não há necessidade de correção da documentação do sistema, apenas dos artefatos associados à correção do código.

$$PF = PF\_ALTERADO \times 0,60$$

## **4.3 Manutenção Cosmética**

São consideradas manutenções cosméticas ou Adaptativas – Mudança de Interface, as demandas associadas à alterações de interface, por exemplo, fonte de letra, cores de telas, logotipos, mudança de botões na tela, mudança de posição de campos ou texto na tela.

Nestes casos, a aferição do tamanho em Pontos de Função da funcionalidade ou das funcionalidades corrigidas considera 10% do PF\_Alterado, seguindo os conceitos do CPM 4.3. Não será contemplada a redocumentação das funcionalidades da aplicação impactadas pela manutenção nas demandas desta categoria.

$$PF = PF\_ALTERADO \times 0,10$$

## **4.4 Manutenção Adaptativa em Requisitos Não Funcionais**

Seguindo os conceitos da IEEE, existem vários tipos de Manutenção Adaptativa. Quando há mudança em requisitos funcionais, estes projetos são denominados de projetos de Melhoria, descritos na seção 4.1. Quando há mudança em requisitos não funcionais de interface, estes projetos são denominados de Manutenção Cosmética ou Manutenção Adaptativa – Mudança de Interface.

Esta seção visa apresentar alguns tipos manutenções adaptativas associadas às mudanças em requisitos não funcionais da aplicação, a saber: Redesenvolvimento de projetos em outra plataforma, Atualização de plataforma, Adequação de funcionalidades às mudanças de negócio. Caso sejam identificados outros tipos de projetos de manutenção adaptativa em requisitos não funcionais, estes devem ser definidos e incorporados nesse Roteiro de Contagem.

#### 4.4.1. Redesenvolvimento de Projetos em outra Plataforma

São considerados nesta categoria, projetos que precisam ser migrados para outra plataforma. Por exemplo, um sistema legado em COBOL precisa ser re-desenvolvido em JAVA.

Como estes projetos legados, freqüentemente, encontram-se sem documentação, então serão considerados como novos projetos de desenvolvimento. Assim, será utilizada a fórmula de Projetos de Desenvolvimento do CPM 4.3.

$$PF = PF\_NÃO\_AJUSTADO + PF\_CONVERSÃO$$

PF\_CONVERSÃO = Pontos de Função associados às funcionalidades de conversão de dados dos projetos de desenvolvimento. Exemplos de funções de conversão incluem: migração ou carga inicial de dados para popular as novas tabelas criadas e relatórios associados à migração de dados.

#### 4.4.2. Atualização de Plataforma

São consideradas nesta categoria, demandas para uma aplicação existente ou parte de uma aplicação existente executar em versões mais atuais de *browsers* (ex: versão atual do *Internet Explorer*, *Mozilla*, *Firefox*,...) ou de linguagens de programação (ex: versão mais atual do *JAVA* ou do Banco de Dados). Também são consideradas nesta categoria aplicações *Web* desenvolvidas para executar em *Internet Explorer* que precisam executar também em *browser* em software livre. Nesta categoria foram observadas demandas dos seguintes tipos:

### **A) Atualização de Plataforma com necessidade de redocumentação de requisitos**

Nestes casos, a aferição do tamanho em Pontos de Função da aplicação ou da parte da aplicação que sofreu impacto considera 50% dos PFs, seguindo a fórmula de projetos de desenvolvimento do CPM 4.3 e as funções de conversão de dados, se aplicável. Deve-se destacar que além da adequação as funcionalidades em questão e da documentação do projeto de manutenção adaptativa realizado, a documentação das funcionalidades deve ser atualizada.

$$PF = (PF\_NÃO\_AJUSTADO \times 0,50) + PF\_CONVERSÃO$$

### **B) Atualização de Plataforma sem necessidade de redocumentação de requisitos**

Nestes casos, a aferição do tamanho em Pontos de Função da aplicação ou da parte da aplicação que sofreu impacto considera 40% dos PFs, seguindo a fórmula de desenvolvimento do CPM 4.3 e as funções de conversão de dados, se aplicável.

$$PF = (PF\_NÃO\_AJUSTADO \times 0,40) + PF\_CONVERSÃO$$

#### **4.4.3. Adequação de Funcionalidades às Mudanças de Negócio**

São consideradas nesta categoria as demandas de manutenção adaptativa associadas à adequação de funcionalidades às mudanças de regras de negócio ou de Legislação ou requisitos de usabilidade **que não se enquadram nas funções alteradas do Projeto de Melhoria**, seguindo as regras de contagem do CPM. Observe que tais solicitações envolvem aspectos não funcionais, sem alteração em requisitos funcionais. Por exemplo: replicação de funcionalidade (chamar uma consulta existente na aplicação de outra tela, por demanda do usuário); replicação de base de dados ou criação de base temporária para resolver problemas de performance ou segurança; Alteração no *software* para adaptação às alterações

realizadas em rotinas de integração com outros *software* (ex: alteração em sub-rotinas chamadas por este *software*). Nesta categoria foram observadas demandas dos seguintes tipos:

#### **A) Adequação de funcionalidades com necessidade de redocumentação**

Nestes casos, a aferição do tamanho em Pontos de Função da funcionalidade ou das funcionalidades que sofreram impacto deve considerar 80% do PF\_Alterado, seguindo os conceitos do CPM 4.3, apresentados na seção 4.1. Deve-se destacar que além da adequação das funcionalidades em questão e da documentação do projeto de manutenção adaptativa realizado, a documentação das funcionalidades deve ser atualizada.

$$PF = PF\_ALTERADO \times 0,80$$

#### **B) Adequação de funcionalidades sem necessidade de redocumentação de requisitos**

Nestes casos, a aferição do tamanho em Pontos de Função da funcionalidade ou das funcionalidades que sofreram impacto deve considerar 70% do PF\_Alterado, seguindo os conceitos do CPM 4.3, apresentados na seção 4.1. Não será contemplada a documentação das funcionalidades nas demandas desta categoria.

$$PF = PF\_ALTERADO \times 0,70$$

Para outros tipos de projetos de manutenção adaptativa não definidos neste documento, considerar um percentual do PF\_Alterado, variando de 30% a 80%, de acordo com as características do requisito não funcional alterado. As premissas utilizadas devem ser acordadas entre o SERPRO e o cliente e documentadas no documento de estimativas do projeto.

## 4.5 Apuração Especial

São funcionalidades executadas apenas uma vez para: corrigir problemas de dados incorretos na base dados das aplicações ou atualizar dados em bases de dados de aplicações, detalhado no item 4.5.1; gerar um relatório específico ou arquivo para o usuário por meio de recuperação de informações nas bases da aplicação, detalhado no item 4.5.2.

Caso a apuração seja de correção de dados, devido a erros de funcionalidades de aplicações desenvolvidas pelo SERPRO. Esta não será cobrada do cliente, desde que a solicitação ocorra dentro do prazo de garantia de 180 dias após a entrega da funcionalidade em questão.

### 4.5.1. Apuração Especial – Base de Dados

Uma apuração especial é um projeto que inclui a geração de procedimentos para atualização da base de dados. Deve-se destacar que estas funções são executadas apenas uma vez, não fazendo parte da aplicação, visando a correção de dados incorretos na base de dados da aplicação ou atualização em função de modificação da estrutura de dados, por exemplo inclusão do indicador de matriz – sim ou não para um CNPJ. Nestes casos, a contagem de Pontos de Função das funcionalidades desenvolvidas. Geralmente, estas funcionalidades são classificadas como Entradas Externas.

$$PF = PF\_NÃO\_AJUSTADO$$

Deve-se ressaltar que as funções de conversão de dados (carga inicial de dados que ocorre na implantação de projetos de desenvolvimento ou de melhoria) não são apurações especiais. Estas funções fazem parte do projeto de desenvolvimento ou de melhoria em questão, portanto devem ser contadas junto com estes projetos e não como apuração especial. Assim, nestes casos, considera-se as fórmulas de contagem de Pontos de Função dos projetos em questão.

Em alguns casos de Apuração Especial – Atualização de dados, o usuário solicita uma consulta prévia das informações atualizadas para validação. De fato, é

uma prática interessante para evitar informações errôneas na base de produção dos sistemas. Esta Consulta Prévia, classificada como Consulta Externa ou Saída Externa deve ser dimensionada, considerando-se 60% do tamanho da funcionalidade em questão, conforme a fórmula abaixo:

$$PF = PF\_NÃO\_AJUSTADO \times 0,60$$

#### 4.5.2. Apuração Especial – Geração de Relatórios

Uma apuração especial é um projeto que inclui a geração de relatórios em uma ou mais mídias para o usuário. Em alguns casos, são solicitadas extrações de dados e envio dos dados para outros sistemas. Caso neste envio de dados sejam requisitadas atualizações no sistema de origem, então estas funções são Saídas Externas, devido à atualização do Arquivo Lógico Interno.

Deve-se destacar que estas funções são executadas apenas uma vez, não fazendo parte da aplicação. Nestes casos, considera-se contagem de Pontos de Função das funcionalidades desenvolvidas. Frequentemente, estas funcionalidades são classificadas como Saídas Externas. Também podem ser classificadas como Consultas Externas, caso não possuam cálculos ou criação de dados derivados.

$$PF = PF\_NÃO\_AJUSTADO$$

#### 4.6 Manutenção em Páginas Estáticas de *Intranet*, *Internet* ou *Portal*

Nesta seção são tratadas manutenções específicas em páginas estáticas de Portais, *Intranets* ou *Websites*. A demanda consiste na publicação de páginas HTML. Estas demandas são consideradas como desenvolvimento de consultas com a utilização de ferramentas para apoiar a publicação. Nestes casos, considera-se 50% dos Pontos de Função das consultas desenvolvidas. Cada página é contada como uma consulta. As consultas são consideradas Consultas Externas Simples (3 PF).

$$PF = PF\_NÃO\_AJUSTADO \times 0,50$$



## 4.7 Manutenção de Documentação de Sistemas Legados

Nesta seção são tratadas demandas de documentação ou atualização de documentação de sistemas legados. Observe que o desenvolvedor deve realizar uma Engenharia Reversa da aplicação para gerar a documentação. Para este tipo de projeto, caso a demanda seja apenas a documentação de requisitos, devem ser considerados 20% dos Pontos de Função da aplicação em questão, conforme a fórmula abaixo.

$$PF = PF\_NÃO\_AJUSTADO \times 0,20$$

Caso a demanda seja a geração de artefatos de documentação de todas as fases do processo de desenvolvimento, deve-se considerar um percentual mais alto de 30% a 50%, dependendo dos artefatos a serem gerados. As premissas utilizadas devem ser acordadas entre o SERPRO e o cliente e documentadas no documento de estimativas do projeto.

## 4.8 Verificação de Erros

São consideradas verificações de erro ou análise e solução de problemas as demandas referentes a todo comportamento anormal ou indevido apontado pelo cliente nos sistemas aplicativos. Neste caso, a equipe de desenvolvimento do SERPRO se mobilizará para encontrar a(s) causa(s) do problema ocorrido. Se for constatado erro de sistema, a demanda será atendida como manutenção corretiva.

Entretanto, uma vez não constatado o problema apontado pelo cliente ou o mesmo for decorrente de regras de negócio implementadas ou utilização incorreta das funcionalidades, será realizada a aferição do tamanho em Pontos de Função das funcionalidades verificadas, e será considerado 25% do tamanho funcional das funcionalidades analisadas, segundo a fórmula abaixo:

$$PF = PF\_NÃO\_AJUSTADO \times 0,25$$

## 4.9 Fator de Criticidade de Solicitação de Serviço

Em função da criticidade e da necessidade de alocação de recursos extras para atendimento da demanda no prazo estipulado pelo cliente, será adotado um Fator de Criticidade de 1,35 (um vírgula trinta e cinco), que deverá ser multiplicado pelo tamanho funcional da demanda considerada crítica, de modo a remunerar adequadamente o aumento do esforço de atendimento. **Este fator é considerado para demandas que devem ser atendidas em finais de semana, feriados e fora do horário comercial. Entende-se como horário comercial o horário de 08:00 às 18:00 h, horário de Brasília.**

## 4.10 Pontos de Função de Teste

Muitas vezes, em projetos de manutenção o conjunto de funções de dados e funções transacionais a serem testadas é maior do que a quantidade de funções a serem implementadas, i.e., além das funcionalidades que são afetadas diretamente pelo projeto de manutenção, outras precisam ser testadas [NESMA, 2009].

O tamanho das funções a serem testadas deve ser aferido em Pontos de Função de Teste (PFT). Não considerar as funcionalidades incluídas, alteradas ou excluídas do projeto de manutenção na contagem de Pontos de Função de Teste.

A contagem de PFT deve considerar o seguinte [NESMA, 2009]:

- Determinar o tamanho em Pontos de Função de cada função de dados ou transacional envolvida no teste.
- Calcular o tamanho em Pontos de Função de todas as funções de dados ou transacionais envolvidas no teste.

A conversão do PFT em Ponto de Função deve ser feita de acordo com a fórmula abaixo:

$$PF = PFT \times 0,20$$

É importante ressaltar que as funções testadas consideradas no PFT devem ser requisitadas pelo cliente e documentadas. Observe que estas funções farão parte do escopo do projeto de manutenção, sendo consideradas para efeito de estimativa de esforço e faturamento junto ao cliente.

## 5. Atividades Sem Contagem de Pontos de Função

Deve-se ressaltar que o processo de desenvolvimento de soluções possui várias atividades que devem ser consideradas como um projeto separado, levando-se em conta as horas realizadas, dentre outras:

- Treinamentos em Tecnologia, Metodologias, Métricas, etc.: encontram-se nesta categoria as demandas de treinamentos em linguagens de programação, ferramentas de gestão, processos, modelos da qualidade, métricas, etc. Estes serviços são executados por consultores do SERPRO, especialistas no assunto em questão. Assim, devem ser consideradas as horas de consultoria para preparação e execução do curso e o custo do deslocamento do instrutor, se for o caso.
- Desenvolvimento de Cursos para EaD: encontram-se nesta categoria as demandas de desenvolvimento de um curso na modalidade de Ensino a Distância (EaD). Estas demandas devem ser remuneradas em horas.
- Mapeamento de Processos de Negócio: encontram-se nesta categoria as demandas de elaboração de documentação contendo o mapeamento de processos de negócio de uma organização ou de parte de uma organização. Estes serviços são executados por consultores do SERPRO, especialistas em BPM (*Business Process Modeling*).
- Elaboração de Plano Diretor de Tecnologia da Informação (PDTI): encontram-se nesta categoria demandas para elaboração de PDTIs para clientes. Estes serviços são executados por consultores do SERPRO, especialistas nas atividades associadas à elaboração de um PDTI.
- Definição de Processo de Desenvolvimento de Soluções: encontram-se nesta categoria demandas para definição de Processos de Software, aderentes às melhores práticas do CMMI e IN04. Estes serviços são executados por consultores do SERPRO, especialistas nas atividades de processos de software e na customização da ferramenta para criação do site do processo.

Outros serviços prestados que também não possuem Pontos de Função associados são os seguintes:

- Administração de Dados: este serviço requer uma equipe de ADs com um número de profissionais definido junto ao Cliente, dedicada para atender as demandas associadas à definição e manutenção do modelo de dados de negócio do cliente. Esta equipe fica disponível em horário comercial para atendimento das demandas. Assim, estes serviços não possuem contagem de PF associada. **É importante ressaltar que as atividades de banco de dados associadas ao projeto de desenvolvimento ou de manutenção, por exemplo, preparação de ambiente (testes, homologação, implantação), desempenhadas pelos DBAs da equipe de desenvolvimento, já estão consideradas dentro do projeto de software, não cabendo cobrança adicional.**

- Análise de Solução: Serviço de apoio destinado à análise de regras de negócio implementadas em soluções de TI. Estas demandas não possuem contagem de PF associada.

- Consultoria: Serviço de apoio destinado à análise de regras de negócio a serem implementadas em soluções de TI realizado por consultores especialistas do SERPRO. As demais modalidades de consultoria também podem ser enquadradas neste item, por exemplo, Consultoria em Métricas. Estas demandas não possuem contagem de PF associada.

Outras atividades contidas em um processo de software devem ser gerenciadas dentro do projeto de desenvolvimento ou de manutenção, no entanto o esforço deve ser considerado separadamente da estimativa de esforço derivado da contagem de Pontos de Função. **Estas atividades também devem ser precificadas a parte.** São elas:

- Treinamento para Implantação: são demandas de treinamentos sobre utilização do sistema a ser implantado para os gestores de solução do cliente e usuários. O esforço deste serviço deve ser considerado separadamente da estimativa de esforço derivada da contagem de PF. O preço deste serviço deve ser calculado, levando-se em conta o preço da hora dos consultores do

SERPRO que estarão realizando atividades de preparação de treinamento e de instrutoria. Em alguns casos, pode ocorrer também o deslocamento do instrutor, que também deve ser cobrado do cliente. Deve-se ressaltar que este treinamento para implantação pode ser definido na modalidade de EaD, sendo tratado como um projeto de treinamento a parte. O esforço deste é considerado dentro do projeto de EaD que não faz parte do projeto de desenvolvimento ou manutenção em questão.

- Especificação de Negócio: esta atividade é a primeira atividade a ser executada em uma demanda de projeto de desenvolvimento e/ou de manutenção. O objetivo desta atividade é gerar a Especificação da demanda. O principal produto gerado nesta atividade é o artefato: Documento de Visão do Projeto (DV), que deve ser validado pelo cliente, por meio da assinatura do termo de aceite. Além do DV podem ser gerados outros documentos, tais como: atas de reunião, documento de requisitos não funcionais e glossário da especificação de negócio. O esforço desta atividade deve ser considerado separadamente da estimativa de esforço derivada da contagem de PF. É importante ressaltar que esta atividade é de responsabilidade dos Analistas de Negócios da empresa contratante, de acordo com a Instrução Normativa (IN 04). Portanto. Caso o cliente demande para o SERPRO a realização desta atividade, esta deve ser cobrada em horas de consultoria. Observe que o Documento de Visão gerado nessa atividade é o insumo para o planejamento (estimativas) e a atividade de Engenharia de Requisitos do processo de desenvolvimento de software. No capítulo seguinte - Considerações Especiais – são propostas métricas para tratar esta atividade (**Seção 6.5**).

## 6. Considerações para Contagem de Pontos de Função

Esta seção apresenta considerações especiais sobre o dimensionamento em Pontos de Função de mudança de requisitos, projetos cancelados e redução de cronograma. E sugere métricas para o dimensionamento de atividades de negócio.

### 6.1 Considerações sobre Mudança de Requisitos

Em projetos de desenvolvimento e manutenção de software é bastante comum as mudanças de requisitos no decorrer do projeto, conforme o usuário e o desenvolvedor adquirem mais conhecimento sobre o negócio [Sommerville, 2007]. O CPM denomina este fenômeno de Scope Creep. Como os requisitos não podem ser congelados, então temos que gerenciá-los de forma efetiva. Nas estimativas iniciais de tamanho de projetos de desenvolvimento, após a fase de especificação, considerando-se o documento de visão inicial do projeto, recomenda-se utilizar um percentual para evolução de requisitos de 30% a 40%. Nas estimativas, após a fase de requisitos, utilizando-se como insumo as especificações de casos de uso, deve-se considerar um percentual de 20% a 30%. Por exemplo, suponha que após a análise do Documento de Visão de um Projeto, aplicando-se a CEPF, foi obtido o tamanho de 200 PFs, então o tamanho estimado do projeto considerado é de 270 PFs (200 + 35%), utilizando-se a premissa de evolução de requisitos em 35%. Esta premissa deve ser documentada.

Uma mudança de requisito gera retrabalho da equipe de desenvolvimento, aumentando assim o esforço e o custo do projeto. Por exemplo, suponha um relatório de clientes em que no final da fase de implementação foi solicitada a exibição de uma nova informação. A equipe de desenvolvimento, terá um retrabalho de várias fases do ciclo de vida. Para tratar o dimensionamento das mudanças de requisitos torna-se importante definir a distribuição de esforço pelas macroatividades do projeto, visando definir o valor agregado ao projeto após cada fase do ciclo de vida.

A Tabela 7 estabelece os percentuais por atividade de forma a permitir a contagem de mudança conforme o estágio do projeto. Esta distribuição percentual de esforço deve ser definida no contrato de software.

Macro Atividades do Processo de	Percentual de esforço
---------------------------------	-----------------------

<b>Desenvolvimento de Software</b>	<b>(%)</b>
Engenharia de Requisitos	25%
Design, Arquitetura	15%
Implementação	40%
Testes	10%
Homologação	5%
Implantação	5%

Por exemplo, suponha um relatório de clientes em que no final da fase de implementação foi solicitada a exibição de uma nova informação. A equipe de desenvolvimento terá um retrabalho de várias fases do ciclo de vida. Assim, o tamanho dessa mudança deve ser calculado da seguinte maneira:

-Tamanho do relatório de clientes (original) – SE – M – 5 PF

-Tamanho do relatório de clientes (alterado) – SE – M- 5 PF

O requisito alterado será considerado 100% do PF, supondo que este será entregue ao cliente sem passar por novas alterações.

Para o requisito original será considerado o seguinte:

Engenharia de Requisitos	25%
Design, Arquitetura	15%
Implementação	40%

Assim, o tamanho da mudança é de 4 PFs (5 PF x 80% = 4 PFs).

## 6.2 Considerações sobre Projetos Cancelados

Em alguns casos, devido às mudanças no ambiente do cliente, uma demanda ou parte de um projeto de desenvolvimento ou manutenção pode ser cancelado pelo cliente. Nestes casos, o tamanho funcional das funcionalidades canceladas será aferido por meio da contagem de Pontos de Função das funcionalidades canceladas e um Fator de Impacto.

O Fator de Impacto é definido com base no percentual de esforço alocado a construção da funcionalidade em questão, observando as tabelas de distribuição de [www.serpro.gov.br](http://www.serpro.gov.br)

esforço contidas na Seção 6.1 ou alguma diretriz específica de distribuição de esforço do contrato em questão. O Fator de Impacto deve ser aplicado na contagem de Pontos de Função das funcionalidades em questão. É importante ressaltar que em um processo de desenvolvimento incremental uma funcionalidade pode por exemplo estar em fase de requisitos e de testes, porque o plano de testes é construído na fase de requisitos. O Progresso das atividades executadas em cada funcionalidade do projeto deve ser obtido por meio do acompanhamento do plano do projeto.

### 6.3 Considerações sobre Redução de Cronograma

As estimativas de prazo não são lineares com o tamanho do projeto, assim pretende-se pesquisar mais sobre o melhor tempo desenvolvimento (onde há uma melhor relação custo x benefício de alocação de recursos e menor prazo de desenvolvimento), dado o tamanho de um projeto específico. Jones [Jones, 2007] propõe uma fórmula para o cálculo do melhor tempo de desenvolvimento, descrita na **seção 3.3**.

Alguns projetos devido à legislação e a outros fatores externos ao SERPRO possuem um prazo imposto pelo cliente. Se este prazo for igual ou superior ao prazo calculado pela Fórmula de *Capers Jones* (expoente t) ou em caso de projetos pequenos (menores que 100 PF) a um prazo calculado considerando o trabalho da equipe de 8 horas/dia nos dias úteis, então este é tratado como um projeto normal.

No entanto, se o projeto tiver um prazo imposto pelo cliente inferior ao prazo calculado, então se deve considerar o seguinte:

- Redução de prazo de 10%: aumento de esforço de 20% (projetos urgentes)
- Redução de prazo de 20%: aumento de esforço de 50% (projetos críticos)
- Redução de prazo de 25%: aumento de esforço de 60% (projetos de alta criticidade)

Deve-se ressaltar que não é possível uma redução de prazo maior que 25 %, devido aos cálculos de Região Impossível e ainda conforme nos aproximamos da região impossível, o esforço e o custo do projeto aumentam de maneira exponencial.

Como os riscos da redução de cronograma também são altos, não é recomendada a redução de cronograma. Deve-se tentar priorizar funcionalidades trabalhando com o ciclo de vida incremental.

Caso o contrato seja baseado em preço por Pontos de Função, este aumento de esforço será refletido na contagem de PF. Assim, um aumento de esforço de 20% implica em aumento de 20% na contagem de PF; aumento de esforço de 50% implica em aumento de 50% na contagem de PF; o aumento de esforço de 60% implica em aumento de 60% na contagem de PF.

## 6.4 Métricas para Atividades de Negócio

Segundo a Instrução Normativa – IN 04 (MPOG/SLTI), o processo de negócios deve ser realizado pelo Analista de Negócios da empresa contratante. No entanto, por se tratar de empresa pública, o SERPRO pode ser contratado por alguns clientes para atividades de negócio. Essas atividades antecedem a fase de requisitos – primeira fase do processo de software e devem ser faturadas em horas de consultoria.

Sugere-se estimar as horas desta atividade da seguinte maneira: Estimar as horas das reuniões de especificação de negócio realizadas com a equipe do cliente, também denominadas de reunião de repasse de especificação; Cada hora de reunião com a participação de analista de negócio corresponde a um esforço de 5 horas do analista (participação da reunião: 1 hora + elaboração de documentação: 4 horas). Por exemplo, suponha uma reunião com duração de 2 horas e a participação de um analista de negócio, assim temos o esforço de 10 horas (2 x 5).

Nesta atividade, deve ser gerado um Documento de Visão (DV), Documento de Requisitos não funcionais e Glossário (opcional). É importante ressaltar que o DV é o artefato utilizado como insumo para a estimativa de tamanho funcional (em Pontos de Função) do projeto e para o desenvolvimento do sistema.

Observando as preferências dos gestores de contratos dos órgãos públicos de adoção de métricas distintas da métrica de esforço homem-hora para todos os tipos de projeto, a Coordenação Estratégica de Tecnologia do SERPRO (CETEC) tem buscado definir métricas objetivas para a precificação de atividades de negócio. Neste contexto foi definida a métrica Pontos de Negócio (PN) descrita a seguir.

- **Pontos de Negócios (PN):** esta métrica deve ser aferida considerando 10% dos Pontos de Função Não Ajustados estimados com base nas funcionalidades identificadas e analisadas nos artefatos de negócio.

**Exemplo Projeto de Data Warehouse:** De posse do modelo de dados do DW, devem ser contadas as Tabelas Fato e Tabelas Dimensão, caso não seja possível identificar a complexidade das mesmas, devido a ausência dos atributos das tabelas, considera-se a complexidade Simples. Deve-se contar duas entradas externas associadas às cargas das tabelas Fato e das tabelas Dimensão, a complexidade de tais funcionalidades deve ser avaliada como Média, considerando a ausência de definição detalhada das funcionalidades. Para cada Estrela, deve-se considerar uma Saída Externa Complexa, considerando a geração do Contexto de Análise. Caso, os relatórios estejam definidos nesta fase, estes devem ser contados como Saídas Externas médias. Caso contrário, não serão contados.

Deve-se ressaltar que a métrica definida está em fase experimental. Assim, esta pode ser utilizada em casos específicos, substituindo a métrica homem\_hora, em acordo entre as partes Contratante e Contratada (SERPRO).

## 7. Contagem de Pontos de Função com Múltiplas Mídias

Esta seção tem como propósito apresentar as diretrizes de Contagem de Pontos de Função utilizadas no SERPRO em relação ao tema Múltiplas Mídias. Esta abordagem é reconhecida pelo IFPUG. As definições apresentadas têm como base o artigo “*Considerations for Counting with Multiple Media*” Release 1.0 publicado pelo IFPUG [IFPUG, 2009].

Considerando-se a contagem de PF de funcionalidades entregues em mais de uma mídia, a aplicação das regras de contagem de Pontos de Função definidas no CPM tem levado a duas abordagens alternativas, a saber: *single instance* e *multiple instance*.

A abordagem *single instance* considera que a entrega de uma função transacional em múltiplas mídias não deve ser utilizada na identificação da unicidade da função.

A abordagem *multiple instance* leva em consideração que a mídia utilizada na entrega da funcionalidade é uma característica de identificação da unicidade da função. Assim, funcionalidades únicas são reconhecidas no contexto da mídia na qual elas são requisitadas para operar.

É importante enfatizar que o IFPUG reconhece ambas abordagens *single instance* e *multiple instance* para a aplicação das regras definidas no CPM. A determinação de da contagem de PF seguindo a abordagem *multiple instance* ou *single instance* depende da avaliação da Coordenação de Métricas da organização. As estimativas e contagens de PF realizadas pelo SERPRO serão baseadas na abordagem *multiple instance*, com exceção dos casos de consultas em .pdf, .doc, .xls e consultas idênticas em tela e papel, que serão consideradas uma única funcionalidade.

A seguir são descritos os termos comuns definidos pelo IFPUG [IFPUG, 2009]:

- **Canal:** também refere-se a mídia. Múltiplos canais é sinônimo de múltiplas mídias.
- **Mídia:** descreve a maneira que os dados ou informações se movimentam para dentro e para fora de uma fronteira de aplicação, por exemplo, apresentação de dados em tela, impressora, arquivo, voz. Este termo é

utilizado para incluir, dentre outros: diferentes plataformas técnicas e formatos de arquivos como diferentes mídias.

- **Múltiplas Mídias:** quando a mesma funcionalidade é entregue em mais de uma mídia. Frequentemente, somente uma mídia é requisitada para um usuário específico em um determinado momento, por exemplo consulta de extrato bancário via internet como oposto a consulta de extrato bancário via terminal do banco.
- **Multi-Mídia:** quando mais de uma mídia é necessária para entregar a função, por exemplo, uma nova notícia publicada na Internet que é apresentada em vídeo e texto. Observe que a notícia completa só é apresentada para o usuário se ele ler o texto e assistir o vídeo.
- **Abordagem *Single Instance*:** esta abordagem não reconhece que a mídia utilizada na entrega da função transacional é uma característica de diferenciação na identificação da unicidade da função transacional. Se duas funções entregam a mesma funcionalidade usando mídias diferentes, elas são consideradas a mesma funcionalidade em uma contagem de Pontos de Função.
- **Abordagem *Multiple Instance*:** esta abordagem especifica que o tamanho funcional é obtido no contexto de objetivo da contagem, permitindo uma função de negócio ser reconhecida no contexto das mídias que são requisitadas para a funcionalidade ser entregue. A abordagem *multiple instance* reconhece que a mídia para entrega constitui uma característica de diferenciação na identificação da unicidade da função transacional.

Os cenários descritos nas seções seguintes não representam uma lista completa de situações de múltiplas mídias. O entendimento destes exemplos facilitará o entendimento de outros cenários envolvendo múltiplas mídias. Este Roteiro deve ser atualizado considerando a publicação de novas diretrizes do IFPUG e novos cenários que emergirão nas contagens de PFs dos projetos dos clientes do SERPRO.

## 7.1 Cenário 1: Mesmos dados apresentados em tela e impressos

Neste cenário, uma aplicação apresenta uma informação em uma consulta em tela. A mesma informação pode ser impressa caso requisitado pelo usuário na tela em questão.

Nesses casos, o SERPRO utiliza a abordagem *single instance*, considerando que dados idênticos sendo apresentados em tela e relatório impresso devem ser contados como uma única função. Caso as lógicas de processamento da consulta em tela e do relatório em papel sejam distintas, o processo elementar não é único e portanto a funcionalidade será contada duas vezes.

Observe que a abordagem *multiple instance* considera que a contagem de PF de dados idênticos sendo apresentados usando mais de um tipo de mídia deve incluir toda instância da função em cada tipo de mídia. Neste exemplo, duas funções são contadas – apresentação de dados em tela; apresentação de dados impressos.

## 7.2 Cenário 2: Mesmos dados de saída como dados em arquivo e relatório impresso

Uma aplicação grava dados em um arquivo de saída e imprime um relatório com informações idênticas as gravadas no arquivo.

Nesses casos, o SERPRO utiliza a abordagem *single instance* considerando que os dados impressos e os dados apresentados no arquivo de saída sejam idênticos. Assim, apenas uma funcionalidade será incluída na contagem de Pontos de Função. Caso as lógicas de processamento da geração do arquivo de saída e do relatório em papel sejam distintas, o processo elementar não é único e portanto a funcionalidade será contada duas vezes.

Observe que a abordagem *multiple instance* considera que dados idênticos estão sendo entregues em mais de um tipo de mídia e a contagem de PF incluirá todas as instâncias de tipos de mídia. Neste cenário, duas funções são contadas – geração arquivo e apresentação dos dados impressos.

### **7.3 Cenário 3: Mesmos dados de entrada batch e on-line**

Uma informação pode ser carregada na aplicação por meio de dois métodos: arquivo batch e entrada on-line. O processamento do arquivo batch executa validações durante o processamento. O processamento on-line também executa validações das informações.

A abordagem *single instance* conta apenas uma funcionalidade.

No SERPRO é utilizada a abordagem *multiple instance* que conta duas funcionalidades: a entrada de dados batch e a entrada de dados on-line. Geralmente, a lógica de processamento utilizada nas validações em modo batch é diferente da lógica de processamento das validações nas entradas de dados on-line. Portanto, o SERPRO contará duas funcionalidades.

### **7.4 Cenário 4: Múltiplos canais de entrega da mesma funcionalidade**

Uma funcionalidade deve ser disponibilizada em múltiplos canais, por exemplo consulta de dados em página Web e consulta de dados no telefone celular.

A abordagem *single instance* conta apenas uma funcionalidade.

No SERPRO é utilizada a abordagem *multiple instance* que conta duas funcionalidades: a consulta de dados na Web e a consulta de dados via celular. Considera-se que a funcionalidade é desenvolvida duas vezes para os dois canais. Algumas vezes, são até projetos de desenvolvimento distintos, um projeto relativo ao sistema Web e outro para o sistema via celular. Portanto, o SERPRO contará duas funcionalidades.

### **7.5 Cenário 5: Relatórios em Múltiplos Formatos**

Um relatório deve ser entregue em diferentes formatos, por exemplo em um arquivo *html* e um formato de valores separados por vírgula.

Nestes casos, conforme sugerido na abordagem *multiple instance*, o SERPRO considera a ferramenta utilizada na geração dos relatórios. Se a equipe de

desenvolvimento precisar desenvolver o relatório nos dois formatos na ferramenta em questão, serão contadas duas funcionalidades. Porque, a lógica de processamento de análise de condições para verificar quais são aplicáveis é identificada. No entanto, se a ferramenta de desenvolvimento suportar um gerador de relatórios que o usuário visualize o relatório em tela e o gerador permita ao usuário imprimir o relatório, salvar em *html* ou salvar no formado de valores separados por vírgula, então o SERPRO contará apenas uma vez, observando que a funcionalidade será da ferramenta e não da aplicação.

## **8. Processo de Revisão do Guia de Contagem**

### **8.1 Revisão para Correção de Inconsistências e Situações não previstas**

A revisão deste guia será feita sempre que o cliente ou o SERPRO verificarem inconsistências entre uma definição do CPM e uma regra constante deste documento e situações não previstas neste guia. Para situações não previstas neste guia, dever-se-á recorrer à equipe de contagem do cliente e a coordenação da área de métricas do SERPRO que decidirão pela atualização deste guia ou do contrato.

### **8.2 Revisão para Adoção de Novas Versões do CPM**

A adoção de nova versão do CPM como referência para este Guia de Contagem não será imediata à sua publicação. Nesse caso deverá haver uma avaliação da nova versão pelo SERPRO e o cliente para se decidir sobre a atualização do guia.

## 9. Conclusão

Este documento apresentou um guia para o dimensionamento de tamanho de todos os tipos de projetos de software desenvolvidos pelo SERPRO, visando a aderência de todos os projetos desenvolvidos na empresa ao processo de Planejamento de Projetos de nível 2 do CMMI e as diretrizes da Instrução Normativa – IN04. A estimativa de tamanho utiliza a métrica de Pontos de Função Não Ajustados como unidade de medida, conforme recomendado nos Acórdãos do Tribunal de Contas da União (TCU).

Como trabalho futuro recomenda-se a revisão e atualização deste roteiro sempre que se verificar inconsistência entre alguma definição do IFPUG, seja publicada em versões futuras do CPM ou em *White Paper*, ou quando for detectado um novo tipo de serviço associado ao desenvolvimento de software não previsto neste trabalho.

Também, pretende-se implantar outros modelos de estimativa de esforço e prazo, por exemplo o COCOMO II, visando a comparação das estimativas de prazo e esforço por mais de um método. O COCOMO II tem sido utilizado pela CETEC na elaboração de Parecer Técnico de Estimativas para clientes, quando requisitado.

## Referências Bibliográficas

- [Boehm, 2000] BOEHM, B.W. **Software Cost Estimation With COCOMO II**. Prentice Hall, New Jersey, 2000.
- [Hazan, 2008] HAZAN, C. **Análise de Pontos de Função: Uma Aplicação nas Estimativas de Tamanho de Projetos de Software**. Engenharia de Software Magazine, Edição 2, Devmedia, pp.25-31.
- [IEEE,1998] IEEE Computer Society. **IEEE Standard for Software Maintenance**. IEEE Std 1219, 1998.
- [Meli, 1999] MELI, R.; SANTILLO, L. **Function Point Estimation Methods: A Comparative Overview**. Proceedings of FESMA 99, Amsterdam, Netherlands, October 1999, pp. 271-286.
- [IEEE,1998] IEEE Computer Society. **IEEE Standard for Software Maintenance**. IEEE Std 1219, 1998.
- [IFPUG,2009] IFPUG. **Considerations for Counting with Multiple Media**. Release 1.0, September, 2009.
- [IFPUG,2010] IFPUG. **Counting Practices Manual**. Version 4.3, January, 2010.
- [Jones, 2007] JONES, C. **Estimating Software Costs**. Second Edition, Mc Graw Hill, 2007.
- [NESMA, 2009] NESMA. **Function Point Analysis for Software Enhancement Guidelines**. Version 2.2.1, 2009
- [Parthasarathy,2007] PARTHASARATHY, M. A. **Practical Software Estimation: function point methods for insourced and outsourced projects**. Addison Wesley, New York, 2007.
- [Roetzheim, 2005] ROETZHEIM, W. **Estimating and Managing Project Scope for New Development**. CrossTalk, Vol. April, 2005.

- [SERPRO, 2008] SERPRO. **Métodos para Estimativa de Projetos de Software Baseado em Pontos de Função**. Relatório do Grupo de Trabalho para Definição da Utilização de Pontos de Função nos Serviços de Desenvolvimento e Manutenção de Sistemas. 2008.
- [Sommerville, 2007] SOMMERVILLE, I. **Software Engineering**. Pearson Education Limited, 8th Edition, 2007.
- [Vazquez, 2010] VAZQUEZ, C. E.; SIMÕES, G. S.; ALBERT, R. M. **Análise de Pontos de Função: Medição, Estimativas e Gerenciamento de Projetos de Software**. 9ª Edição. Editora Érica, São Paulo.

**ANEXO I: Documento de Requisitos de Projetos de Manutenção  
Pequenos (< 100 PFs)**

<Nome do Sistema>

**Documento de Requisitos de Projetos de Manutenção  
Pequenos**

Versão 1.0

**Histórico da Revisão**

<b>Data</b>	<b>Versão</b>	<b>Descrição</b>	<b>Autor</b>	<b>Aprovador</b>

## Formalização Simples de Requisitos

Dados Gerais	
Número do SIGOP/ Demanda/ RT	
Nome do Sistema Mantido	
Tecnologia Adotada	
Data do Início do Serviço	DD/MM/AAAA
Data do Término do Serviço	DD/MM/AAAA
Descrição da Solicitação	

### Descrição do Serviço Executado

Requisito	Detalhamento
1.	1.1 1.2...
2.	2.1 2.2...

## Identificação da Manutenção

Tipo	
	<b>Melhoria (Evolutiva)</b>
	<b>Corretiva – Funcionalidade Fora do Prazo de Garantia</b>
	<b>Corretiva – Funcionalidade Não Desenvolvida pelo SERPRO</b>
	<b>Verificação de erros (Análise e Solução de Problemas)</b>
	<b>Cosmética – Adaptação de Interfaces</b>
	<b>Adaptativa – Requisitos Não Funcionais</b>
	<b>Tipo:</b>
	<b>Apuração Especial – Base de Dados</b>
	<b>Apuração Especial – Base de Dados – Consulta Prévia</b>
	<b>Apuração Especial – Emissão de Relatório</b>
	<b>Publicação de Página em Intranet / Portal / Internet</b>

Foi demandada a redocumentação da funcionalidade mantida?

**Sim**       **Não**

Aplicar Fator Criticidade?

**Sim**       **Não**

Observações relevantes quanto ao tipo de manutenção:

Descrição dos Requisitos de Manutenção (para cada funcionalidade alterada, utilizar um quadro)

a) Tabelas Modificadas pela Manutenção

Nome da Tabela	
A Tabela é atualizada por alguma funcionalidade da aplicação: <input type="checkbox"/> Sim <input type="checkbox"/> Não	
A Tabela é atualizada por outra aplicação: <input type="checkbox"/> Sim <input type="checkbox"/> Não	
A Tabela foi: <input type="checkbox"/> Incluída <input type="checkbox"/> Alterada <input type="checkbox"/> Excluída	
Total de Campos da Tabela após a Manutenção =	
Campos Incluídos/Alterados/Excluídos	

A funcionalidade será **apenas** testada?

Sim  Não

b) Entradas de Dados Afetadas pela Manutenção (telas ou arquivos de carga)

Nome da Entrada	
Total de Campos na Entrada =	
Nome das Tabelas Acessadas (Lidas e Gravadas) =	
Campos Incluídos/Alterados/Excluídos	

Houve mudança na regra de negócio (validações, lógica de processamento, regras de cálculo)?

Sim  Não

A funcionalidade será **apenas** testada?

Sim  Não

c) Consultas Afetadas pela Manutenção

Considere a tela de parâmetros e a de resultados da consulta como apenas uma única Consulta. Caso a consulta seja do tipo lista e consulta detalhes, considere como funções independentes e preencha quadros diferentes.

Nome da Consulta	
Total de Campos da Consulta	
Tabelas Acessadas	
Total de Campos Afetados =	
Total de Campos Calculados ou Totalizadores =	
Existe atualização de dados (log, indicador...) <input type="checkbox"/> Sim <input type="checkbox"/> Não	
Campos Incluídos/Alterados/Excluídos	

Houve mudança na regra de negócio (validações, lógica de processamento, regras de cálculo, campos de filtro)?

Sim  Não

A funcionalidade será **apenas** testada?

Sim  Não

#### d) Relatórios Afetados pela Manutenção

Considere a tela de parâmetros e a de resultados do relatório como apenas um único Relatório.

Nome do Relatório	
Total de Campos no Relatório	

<b>Tabelas Acessadas</b>	
<b>Total de Campos Afetados =</b>	
<b>Total de Campos Calculados ou Totalizadores =</b>	
Existe atualização de dados (log, indicador...) <input type="checkbox"/> Sim <input type="checkbox"/> Não	
<b>Campos Incluídos/Alterados/Excluídos</b>	

Houve mudança na regra de negócio (validações, lógica de processamento, regras de cálculo, campos de filtro)?

Sim  Não

A funcionalidade será **apenas** testada?

Sim  Não

## **ANEXO II: Modelo de Documento de Contagem de Pontos de Função para Projetos de Manutenções Pequenos (< 100 PFs)**

### **Documento de Contagem Pontos de Função de Projetos de Manutenção Pequenos**

**Cliente: Secretaria do Tesouro Nacional**

#### **Histórico da Revisão**

<b>Data</b>	<b>Versão</b>	<b>Descrição</b>	<b>Autor</b>	<b>Aprovador</b>

Nome Projeto:

Número do SIGOP / Demanda / RT:

Responsável pela Contagem:

Descrição da Solicitação de Mudança:

<b>Descrição da Atividade</b>	<b>Contagem PF</b>	<b>Tipo de Manutenção / Total PFs</b>

Observações Relevantes:

Conforme a tabela de atividades acima, o total de Pontos de Função realizados no Projeto \_\_\_\_\_ na SM \_\_\_\_\_ é de \_\_\_\_\_ PFs Ajustados.