

**MINISTÉRIO DO PLANEJAMENTO**  
**Secretaria de Logística e Tecnologia da Informação**

**Roteiro de Métricas de Software do SISP**  
**Versão 1.0**

Brasília, 29 de novembro de 2010.

**Presidente da República**

Luiz Inácio Lula da Silva

**Ministério do Planejamento, Orçamento e Gestão**

Paulo Bernardo Silva

**Secretaria de Logística e Tecnologia da Informação – SLTI**

Maria da Glória Guimarães dos Santos

**Departamento de Integração de Sistemas – DSI**

Nazaré Lopes Bretas

**Coordenação Geral de Gestão Corporativa – CGGC**

Claudio Muniz Machado Cavalcanti

**Equipe de Elaboração**

Carlos Renato dos Santos Ramos (SLTI/MP)

Claudia Hazan (SERPRO)

Claudio Muniz Machado Cavalcanti (SLTI/MP)

Emanuelle Monteiro Silva (SLTI/MP)

Fátima Saldanha Cesarino (CEF)

Gileno Dias dos Santos (SLTI/MP)

Jose Romildo Araujo de Andrade (SLTI/MP)

Lucinéia Turnes (SLTI/MP)

Marcelo Paiva Fernandes (SLTI/MP)

Maurício Koki Matsutani (DATAPREV)

Patrícia Oliveira de Carvalho (SUSEP)

Rafael Campos (SLTI/MP)

Rafael Monteiro dos Santos Escolástico (MEC)

Regiane Brito (BACEN)

Rosa Maria da Costa Medeiros (INEP)

# Sumário

1. Introdução.....	5
2. Objetivo.....	6
3. Contagem de Pontos de Função.....	6
3.1 Determinar Propósito, Tipo, Escopo e Fronteira de Contagem .....	7
3.2 Funções de Dados e Funções Transacionais.....	7
3.3 Cálculo de Pontos de Função.....	9
4. Itens Não Mensuráveis .....	11
4.1 Projeto de Melhoria.....	11
4.2 Projetos de Migração de Dados.....	14
4.3 Manutenção Corretiva.....	15
4.4 Redesenvolvimento de Projetos em outra Plataforma.....	16
4.5 Atualização de Plataforma.....	16
4.6 Manutenção em Interface.....	17
4.7 Adaptação de Funcionalidades a Requisitos Não Funcionais.....	17
4.8 Apuração Especial.....	18
4.8.1 Apuração Especial – Base de Dados.....	18
4.8.2 Apuração Especial – Geração de Relatórios.....	19
4.8.3 Reexecução de Apuração Especial.....	19
4.9 Atualização de Dados.....	19
4.10 Manutenção em Páginas Estáticas de Intranet, Internet ou Portal.....	20
4.11 Manutenção de Documentação de Sistemas Legados.....	20
4.12 Verificação de Erros.....	21
4.13 Pontos de Função de Teste.....	21
4.14 Manutenção de Componentes.....	22
5. Estimativas de Projetos de Software.....	22
5.1 Contagem Estimativa de Pontos de Função (CEPF).....	24
5.2 Estimativa de Esforço de Projetos de Software.....	28
5.2.1 Distribuição de Esforço por Fase do Projeto.....	28
5.3 Estimativa de Prazo de Projetos de Software.....	29
5.4 Alocação de Equipe ao Projeto.....	31
5.5 Método para Estimativa de Custo.....	32
5.6 Estimativa de Recursos Computacionais.....	32
6. Considerações para Contagem de Pontos de Função.....	33
6.1 Considerações sobre Mudança de Requisitos.....	33
6.2 Considerações sobre Projetos Cancelados.....	34
6.3 Considerações sobre Redução de Cronograma .....	34
6.4 Fator de Criticidade de Solicitação de Serviço.....	35
7. Contagem de Pontos de Função com Múltiplas Mídias e Dados de Código.....	35
7.1 Cenário 1: Mesmos dados apresentados em tela e impressos.....	36
7.2 Cenário 2: Mesmos dados de saída como dados em arquivo e relatório impresso.....	36
7.3 Cenário 3: Mesmos dados de entrada batch e on-line.....	37
7.4 Cenário 4: Múltiplos canais de entrega da mesma funcionalidade.....	37
7.5 Cenário 5: Relatórios em Múltiplos Formatos.....	37
7.6 Dimensionamento de Dados de Código.....	38
8. Atividades Sem Contagem de Pontos de Função.....	38
9. Processo de Revisão do Guia de Contagem.....	40
9.1 Revisão para Correção de Inconsistências e Situações não previstas.....	40

9.2 Revisão para Adoção de Novas Versões do CPM.....	40
10. Conclusão.....	40
Referências Bibliográficas.....	42
Anexo I - Formalização Simples de Requisitos.....	43

# 1. Introdução

Diversas instituições, públicas e privadas, 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, destacando: regras de contagem objetivas e independência da solução tecnológica utilizada. É importante ressaltar que a Instrução Normativa IN04 SLTI/MPOG 2010 recomenda o uso de métricas em soluções de software, restringindo o uso da métrica de esforço homem-hora. Os Acórdãos do Tribunal de Contas da União (TCU) recomendam a utilização da métrica Pontos de Função Não Ajustados em contratos de prestação de serviços de desenvolvimento e manutenção de sistemas.

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

Além disso, 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 e custo dos projetos de software da empresa, visando melhorar o gerenciamento dos projetos.

É importante frisar que o Manual de Práticas de Contagem (CPM) é um documento que se destina a mensurar o tamanho funcional de projetos de software, não tendo por objetivo principal suportar contratos de fábrica de software. Assim, torna-se necessário criar roteiros complementares, contemplando questões não abordadas pelo manual do IFPUG, mas vivenciadas pelos órgãos e entidades do SISP.

Este documento encontra-se organizado da seguinte maneira: o Capítulo 1 apresenta a motivação para a elaboração do documento e a sua organização; o Capítulo 2 mostra os objetivos aos quais este documento se propõe; o Capítulo 3 apresenta diretrizes para a Contagem de Pontos de Função de Projetos de Desenvolvimento e de Melhoria; o Capítulo 4 define os tipos de projetos de manutenção de software não mensuráveis pelo Manual de Práticas de Contagem (CPM) assim como métricas baseadas em Pontos de Função para o dimensionamento desses itens não mensuráveis; Capítulo 5 define um processo de estimativas de projetos de software aderente às melhores práticas do CMMI nível 2; o Capítulo 6 apresenta algumas considerações importantes sobre utilização de métricas para dimensionar as mudanças de requisitos e redução de cronograma; o Capítulo 7 estabelece diretrizes para contagem de Múltiplas Mídias e de Dados de Código (Code Data); o Capítulo 8 apresenta algumas atividades que não devem ser consideradas nas Contagens de Pontos de Função; o Capítulo 9

apresenta o processo de revisão deste guia de contagem; finalmente, o Capítulo 10 conclui o documento apresentando sugestões para trabalhos futuros.

## 2. Objetivo

Este documento tem como propósito apresentar um roteiro de métricas, considerando as regras de contagem Análise de Pontos de Função (APF), em aderência ao Manual de Práticas de Contagem (CPM 4.3), definir os outros tipos de projetos de manutenção, não abordados pelo CPM, e métricas baseadas na APF para dimensionar o tamanho de tais projetos, visando promover o uso de métricas objetivas em contratos de prestação de serviços de desenvolvimento e manutenção de software. 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, visando apoiar as organizações nas estimativas de tamanho, custo, prazo e esforço de seus projetos desenvolvidos internamente ou contratados.

O documento foi construído baseando-se no “Roteiro SERPRO de Métricas para Contratos de Software”. Também foram consultados outros roteiros já utilizados em órgãos que utilizam a métrica Pontos de Função em contratos de software. A proposta inicial do Roteiro foi submetida ao Grupo de Trabalho de Métricas do SISP, o qual contribuiu com sugestões de melhoria para o Roteiro. Finalmente, as propostas foram analisadas em reunião com especialistas em métricas de Órgãos do SISP. E então foi elaborada esta versão do Roteiro.

A implantação de um processo de medições de software e a mudança do modelo de contratação de software com base na métrica homem-hora para o novo paradigma de contratação com base em métricas requer uma mudança cultural. Este Roteiro tem como propósito apoiar os Órgãos do SISP nessa mudança cultural. Desta forma, duas premissas foram consideradas na elaboração desse Roteiro:

- **Simplicidade:** Este Roteiro deve ser simples para incentivar os Órgãos do SISP a utilizar a métrica Pontos de Função como medida padrão no estabelecimento de contratos de fábrica de software.
- **Consistência:** Este Roteiro deve definir critérios objetivos visando prover a consistência no uso de métricas em contratos de fábrica de software. Ou seja, dois profissionais ao aplicar o Roteiro no dimensionamento de um projeto de software devem chegar no mesmo resultado.

## 3. Contagem de Pontos de Função

A métrica PF mede o tamanho funcional de um projeto de software, observando as funcionalidades implementadas, considerando a visão do usuário. Tamanho funcional é definido como “tamanho do software derivado pela quantificação dos requisitos funcionais do usuário” [Dekkers, 2003]. A Análise de Pontos de Função (APF) é um método padrão para a medição de projetos de desenvolvimento e de manutenção de sistemas, visando estabelecer uma medida de tamanho do software em Pontos Função, com base na quantificação da funcionalidade solicitada e entregue, sob o ponto de vista do usuário. Assim, a APF tem como objetivo medir o que o software faz, por meio de uma avaliação padronizada dos requisitos de negócio do sistema.

O Manual de Práticas de Contagem (CPM) [IFPUG, 2010], publicado pelo IFPUG,

apresenta as regras de contagem de Pontos de Função de Projetos de Desenvolvimento, Projetos de Melhoria e Aplicações Implantadas. A Figura 1 ilustra o procedimento de contagem de Pontos de Função, descrito nas seções seguintes.

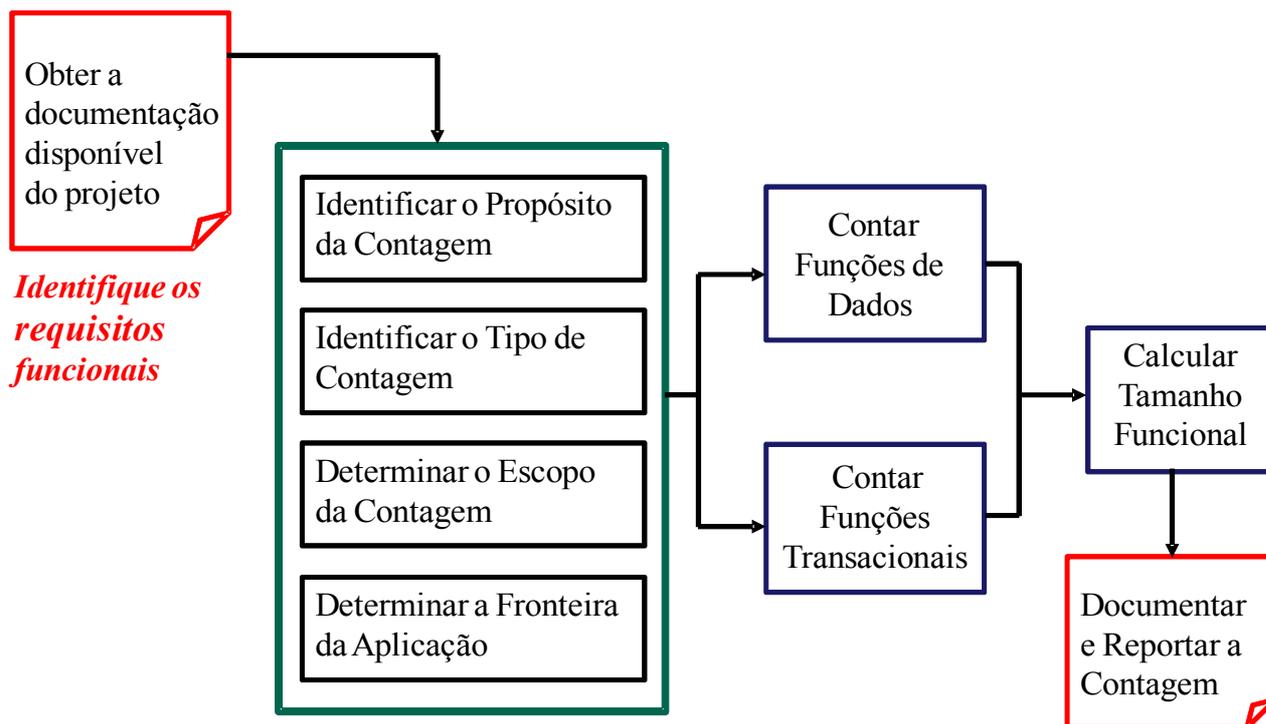


Figura 1: Procedimento de Contagem de Pontos de Função

### 3.1 Determinar Propósito, Tipo, Escopo e Fronteira de Contagem

A Contagem de Pontos de Função se inicia com a análise da documentação disponível do projeto em questão, visando a identificação dos requisitos funcionais. O próximo passo é o estabelecimento do propósito da contagem, o qual fornece uma resposta para uma questão de negócio a ser resolvida, por exemplo: necessidade de dimensionar um projeto de um novo sistema para a contratação do mesmo. Com base no propósito da contagem são definidos o escopo da contagem, que identifica quais funcionalidades serão incluídas na contagem de Pontos de Função, e o tipo de contagem, que pode ser projeto de desenvolvimento, projeto de melhoria ou aplicação instalada. A fronteira da aplicação, que é a interface conceitual que delimita o sistema sendo medido e os usuários e as outras aplicações deve ser definida com base na visão do usuário, desconsiderando questões de implementação.

Em Editais para contratação de projetos de manutenção é fortemente recomendado a definição das fronteiras de todas as aplicações a serem contratadas no Edital. Os Roteiros de Contagem dos Órgãos também devem definir as fronteiras das aplicações implantadas em um anexo e este deve ser atualizado sempre que for implantada uma nova aplicação no Órgão.

### 3.2 Funções de Dados e Funções Transacionais

Uma vez estabelecida a fronteira da contagem, o próximo passo é o mapeamento dos requisitos de dados e de funções transacionais para os tipos funcionais da APF, a saber:

- **Arquivo Lógico Interno (ALI):** é um grupo de dados, logicamente relacionados, reconhecido pelo usuário, mantido por meio de um processo elementar da aplicação que está sendo contada.
- **Arquivo de Interface Externa (AIE):** é um grupo de dados, logicamente relacionados, reconhecido pelo usuário, mantido por meio de um processo elementar de uma outra aplicação e referenciado pela aplicação que está sendo contada. O AIE é obrigatoriamente um ALI de outra aplicação.
- **Entrada Externa (EE):** é um processo elementar que processa dados ou informação de controle que entram pela fronteira da aplicação. Seu objetivo principal é manter um ou mais ALI ou alterar o comportamento do sistema.
- **Consulta Externa (CE):** é um processo elementar que envia dados ou informação de controle para fora da fronteira da aplicação. Seu objetivo principal é apresentar informação para o usuário através da recuperação de dados ou informação de controle de ALI ou AIE.
- **Saída Externa (SE):** é um processo elementar que envia dados ou informação de controle para fora da fronteira da aplicação. Seu objetivo principal é apresentar informação para um usuário ou outra aplicação através de um processamento lógico adicional à recuperação de dados ou informação de controle. O processamento lógico deve conter cálculo, ou criar dados derivados, ou manter ALI ou alterar o comportamento do sistema.

Após a identificação do tipo funcional, deve-se avaliar a complexidade (Baixa, Média, Alta) e a contribuição funcional do mesmo para a contagem de Pontos de Função, observando as regras de contagem de Pontos de Função descritas no CPM. A Tabela 1 apresenta a contribuição dos tipos funcionais na contagem de Pontos de Função.

Tipo Funcional	Complexidade		
	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

**Tabela 1: Contribuição Funcional dos Tipos Funcionais**

### 3.3 Cálculo de Pontos de Função

Caso o tipo de contagem seja projeto de desenvolvimento, o cálculo do tamanho funcional é definido no CPM conforme a fórmula abaixo:

$$\text{PF\_Desenvolvimento} = \text{PF INCLUIDO} + \text{PF\_CONVERSÃO}$$

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

**PF\_CONVERSÃO** = Pontos de Função associados às funcionalidades de conversão de dados. 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.

Caso o tipo de contagem seja projeto de melhoria, o cálculo do tamanho funcional é definido no CPM conforme a fórmula abaixo:

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

**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.

#### Observação1: Função Alterada

Uma função de dados (Arquivo Lógico Interno ou Arquivo de Interface Externa) é considerada alterada quando houver inclusão ou exclusão de tipo de dados. Também é considerada alterada se algum tipo de dado sofrer mudança de tamanho (número de posições) ou tipo de campo (por exemplo: mudança de numérico ou alfanumérico), caso a mudança decorra de alteração de regra de negócio.

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

- Mudança de tipos de dados;
- Mudança de arquivos referenciados;
- Mudança de lógica de processamento.

O CPM 4.3 define lógica de processamento 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

### **Observação 2: Outros Tipos de Funções Alteradas**

Este Roteiro considera função alterada, qualquer mudança em funcionalidades da aplicação, devido às mudanças de Regras de Negócio. Por exemplo, uma funcionalidade de cadastro implicava na inclusão de um telefone do gerente. Devido a mudanças no processo de negócio, a funcionalidade deve sofrer uma manutenção adaptativa para cadastrar dois telefones do gerente. Desta forma, o Roteiro considera esta função como uma Entrada Externa alterada, mesmo que não exista mudança de lógica, mudança de Tipos de Dados e mudança de arquivos referenciados. Serão tratadas como manutenções adaptativas apenas as manutenções que implicarem em apenas mudanças em requisitos não funcionais. Se uma mesma funcionalidade tiver mudanças em requisitos funcionais e não funcionais, esta deve ser contada apenas uma vez, como função alterada em um Projeto de Melhoria.

### **Observação3: Migração de Dados – PF\_Conversão**

As características do esforço de desenvolvimento ou de melhoria de um sistema são bastante diferentes daquelas do esforço de migração de dados, exigindo habilidades diferentes e geralmente resultando em equipes distintas para esses dois tipos de tarefas. Devido a isso, é fortemente recomendado que os projetos de migração de dados sejam tratados como objetos separados dos projetos de desenvolvimento e de melhoria e tenham precificação própria. Dessa forma, este Roteiro recomenda suprimir o PF\_Conversão das fórmulas de desenvolvimento e de melhoria e considerar estas funcionalidades como um projeto de migração de dados, que será dimensionado como um novo projeto de desenvolvimento, conforme a Seção 4.2 deste documento.

O capítulo seguinte define métricas para tratar os itens não mensuráveis pelo CPM, ou seja os demais tipos de projetos de manutenção de software.

## 4. Itens Não Mensuráveis

O manual de práticas de contagem – CPM não contempla projetos de manutenção (maintenance) apenas o de Melhoria (enhancement). Este capítulo tem como propósito descrever os diversos projetos de manutenção e definir métricas baseadas nas regras de contagem de Pontos de Função do CPM para seu dimensionamento.

Quanto à documentação de projetos de manutenção pequenos (menores que 100 PF), deve-se registrar a solicitação e documentar os requisitos do projeto de manutenção e 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 de Documento de Requisitos para projetos de manutenção.

### 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.

Segue a Fórmula de Contagem de Pontos de Função de Projetos de Melhoria.

$$PF = PF\_INCLUIDO + (0,40 \times PF\_EXCLUIDO) + (FI \times PF\_ALTERADO)$$

Definições:

**PF\_INCLUIDO** = 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.

**FI = Fator de Impacto** pode variar de 50% a 100% de acordo com o seguinte:

- Sistema ou funcionalidade desenvolvido pela empresa contratada : **FI = 50%**
- Sistema não desenvolvido pela empresa contratada **com documentação atualizada: FI = 75%**
- Sistema não desenvolvido pela empresa contratada **sem documentação atualizada: FI =100%**. A contratada deve redocumentar a funcionalidade mantida, gerando a documentação completa da mesma, aderente ao processo de software da contratante. Tendo em vista que o FI equivale a inclusão de funcionalidade, se houver uma nova demanda de projeto de melhoria na funcionalidade em questão será considerado que a contratada desenvolveu a funcionalidade. Observe que o percentual de 100% apenas será considerado na primeira demanda de projeto de melhoria em cada funcionalidade.

Os percentuais de multiplicação propostos são estimados, podendo ser reajustados conforme avaliação da base histórica dos serviços realizados no Órgão.

Para os Órgãos que já possuem base histórica de contagem de Pontos de Função Detalhada de projetos concluídos e um processo de desenvolvimento implantado com documentação detalhada das aplicações a serem mantidas podem utilizar o método de Contagem de Pontos de Função de Projetos de Melhoria publicado pela NESMA – Netherlands Software Metrics Users Association no documento "Function Point Analysis for Software Enhancement Guidelines" [NESMA, 2009].

Para o cálculo dos Pontos de Função Incluídos, Alterados e Excluídos, segue as fórmulas estabelecidas abaixo [NESMA, 2009]:

**PF\_INCLUIDO** = QPI;

**PF\_EXCLUIDO** = QPE x 0.40;

**PF\_FD\_ALTERADO** = FD-QPA x FFD, sendo FFD entre 0,25 e 1,00, conforme tabela abaixo (para funções de dados);

**PF\_FT\_ALTERADO** = FT-QPA x FFT, sendo FFT entre 0,25 e 1,50, conforme tabela abaixo (para funções transacionais);

**PF\_ALTERADO** = PF\_FD\_ALTERADO + PF\_FT\_ALTERADO.

Onde:

QPI = Quantidade de Pontos de Função incluídos;

QPE = Quantidade de Pontos de Função excluídos;

PF\_FD\_ALTERADO = Pontos de Função alterados para Funções de Dados;

PF\_FT\_ALTERADO = Pontos de Função alterados para Funções Transacionais;

FD-QPA = Quantidade de Pontos de Função alterados em funções de dados;

FT-QPA = Quantidade de Pontos de Função alterados em funções transacionais;

FFD = Fator de impacto de alterações em funções de dados;

FFT = Fator de impacto de alterações em funções transacionais

O cálculo dos fatores de impacto são obtidos através dos percentuais de alterações conforme abaixo:

**a) Funções de Dados:**

Percentual de alterações em funções de dados:  $PFD = QTDA / QTDT \times 100$

QTDA = Quantidade de Tipos de Dados Alterados

QTDT = Quantidade de Tipos de Dados Totais na Função Original

Com o valor obtido em PFD, procura-se na tabela qual o valor do fator de impacto:

Fator de Impacto em Funções de Dados Alteradas (FFD)				
PFD	Entre 0 e 33%	Entre 33% e 66%	Entre 66% e 100%	Maior que 100%
Fator de Impacto (FFD)	0,25	0,5	0,75	1

**b) Funções Transacionais:**

Percentual alterações em funções transacionais (PFT1) =  $QTDA / QTDT \times 100$

Percentual alterações em funções transacionais (PFT2) =  $QFTRA / QFTRT \times 100$

QTDA = Quantidade de Tipos de Dados Alterados

QTDT = Quantidade de Tipos de Dados Totais na Função Original

QARA = Quantidade de Arquivos Referenciados Alterados

QART = Quantidade de Arquivos Referenciados Totais na Função Original

Fator de Impacto em Funções Transacionais Alteradas (FFT)			
PFT2 / PFT1	Entre 0% e 66%	Entre 66% e 100%	Maior que 100%
Entre 0% e 33%	0,25	0,5	0,75
Entre 33% e 66%	0,5	0,75	1
Entre 66% e 100%	0,75	1	1,25
Maior que 100%	1	1,25	1,5

Caso FFT seja maior que 1, recomenda-se reconsiderar a melhoria.

**Observação Importante**

Embora este método seja utilizado por algumas organizações, é importante ressaltar que em algumas simulações realizadas com base no método da NESMA, os resultados ficaram inconsistentes.

Exemplo 1: Suponha uma aplicação com leitura de um Arquivo de Interface Externa

com leitura de 2 Tipos de Dados (nome, taxa). Na manutenção, o usuário solicita a leitura de um novo Tipo de Dados, por exemplo uma Sigla. Então, considerando o método NESMA, a contagem de PF da função de dados seria a seguinte:

Aplicação Implantada : AIE: Taxa – RL:1 TD:2 - 5 PF

Manutenção: inclusão do TD – Sigla

$PFD = QTDA / QTDT \times 100$

$PFD = (1 / 2) \times 100 = 50\%$

$FFD = 0,5$

$PF\_Alterado = 5 \times 0,5 = 2,5 \text{ PFs}$

Exemplo 2: Suponha uma aplicação com leitura de um Arquivo de Interface Externa com leitura de 47 Tipos de Dados. Na manutenção, o usuário solicita a leitura de um novo Tipo de Dados, por exemplo uma Sigla. Então, considerando o método NESMA, a contagem de PF da função de dados seria a seguinte:

Aplicação Implantada : AIE: Taxa – RL:1 TD:47 - 5 PF

Manutenção: inclusão do TD – Sigla

$PFD = QTDA / QTDT \times 100$

$PFD = (1 / 47) \times 100 = 2,1 \%$

$FFD = 0,25$

$PF\_Alterado = 5 \times 0,25 = 1,25 \text{ PFs}$

Observe que o esforço para a alteração foi o mesmo ou superior na Aplicação do exemplo 2. No entanto, pelo método da NESMA, a contagem de PF do exemplo 2 é a metade (1,25 PFs) da contagem de PF do exemplo 1 (2,5 PFs). Além disso, nos casos de ausência de documentação atualizada da contagem inicial, a operacionalização da contagem de PF por esse método pode ser bastante trabalhosa.

Assim, considerando a inconsistência do método, suas dificuldades de operacionalização e o fato de que na prática é observado que o esforço para alteração de uma funcionalidade depende do conhecimento da equipe sobre a aplicação implantada e da qualidade da sua documentação e do seu código, este Roteiro não recomenda o uso do método sem base histórica de projetos e documentação atualizada e propõe o uso dos mesmos fatores de impacto levando em conta a familiaridade com o código e a qualidade da documentação.

## 4.2 Projetos de Migração de Dados

Conforme mencionado, este Roteiro recomenda a supressão do PF\_Conversão das fórmulas de contagem de Pontos de Função de Projetos de Desenvolvimento e de Melhoria e o tratamento das funções de migração de dados como projetos separados de migração de dados.

Os projetos de migração de dados devem ser contados como um novo projeto de desenvolvimento de um sistema, contemplando minimamente: os ALIs mantidos pela migração, Entradas Externas – considerando as cargas de dados nos ALIs e caso seja solicitado pelo usuário relatórios gerenciais das cargas estes serão contados como Saídas Externas. Todas as contagens de PF devem ser realizadas com base nas

funcionalidades requisitadas e recebidas pelo usuário.

### **4.3 Manutenção Corretiva**

Mesmo com a execução de atividades de garantia da qualidade, pode-se 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 tiver sido desenvolvido pela contratada, a manutenção corretiva será do tipo Garantia, conforme prazos e demais cláusulas do contrato em questão. Caso não exista cláusula contratual de Garantia, deve ser considerada a garantia de 6 meses, preconizada por lei (Código do Consumidor).

Quando o sistema não tiver sido desenvolvido pela contratada, deverá ser estimado e calculado o tamanho do projeto de manutenção corretiva. 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 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.

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

#### **b) 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.

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

#### **c) 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 pela contratada.

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

Em todos os três itens acima, os percentuais de multiplicação são estimados, podendo ser reajustados conforme avaliação da base histórica dos serviços realizados no Órgão.

#### **4.4 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 redesenvolvido em JAVA.

Como estes projetos legados, frequentemente, 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, com a supressão do PF\_CONVERSÃO, considerando que as Migrações de Dados serão tratadas como um projeto a parte, conforme a Seção 4.2.

$$PF = PF\_NÃO\_AJUSTADO$$

#### **4.5 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.

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

## **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.

$$\text{PF} = \text{PF\_NÃO\_AJUSTADO} \times 0,40$$

O PF\_CONVERSAO deve ser considerado como um projeto de migração a parte, conforme descrito na seção 4.2.

Nos dois itens acima, os percentuais de multiplicação são estimados, podendo ser reajustados conforme avaliação da base histórica dos serviços realizados no Órgão.

## **4.6 Manutenção em Interface**

A manutenção em Interface, denominada na literatura manutenção cosmética, é associada às demandas de 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. Também se enquadram nessa categoria as mudanças de texto em mensagens de erro, validação, aviso, alerta ou conclusão de processamento.

Nestes casos, a aferição do tamanho em Pontos de Função das funcionalidades alteradas será aferido com a aplicação de um fator de modo a considerar 10% da contagem de uma função transacional de baixa complexidade (3 PFs), independentemente da complexidade da funcionalidade alterada. Caso seja utilizada uma mesma tela para duas ou mais funcionalidades, deve ser contada **APENAS UMA** função transacional de baixa complexidade.

Não será contemplada a redocumentação das funcionalidades da aplicação impactadas pela manutenção nas demandas desta categoria.

$$\text{PF} = \text{PF\_Alterado\_Função\_Transacional\_Complexidade\_Baixa} \times 0,10$$

O percentual de multiplicação é estimado, podendo ser reajustado conforme avaliação da base histórica dos serviços realizados no Órgão.

## **4.7 Adaptação de Funcionalidades a Requisitos Não Funcionais**

São consideradas nesta categoria as demandas de manutenção adaptativa associadas a solicitações que envolvem aspectos não funcionais, sem alteração em requisitos funcionais. Por exemplo: replicação de funcionalidade (chamar uma consulta existente em outra tela da aplicação); replicação de base de dados ou criação de base temporária para resolver problemas de performance ou segurança; alteração na aplicação para adaptação às alterações realizadas na interface com rotinas de integração com outros softwares. Nesta categoria foram observadas demandas dos seguintes tipos:

### **a) Adaptação de funcionalidades sem necessidade de redocumentação**

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.

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

#### **b) Adaptação 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.

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

Nos dois itens acima, os percentuais de multiplicação são estimados, podendo ser reajustados conforme avaliação da base histórica dos serviços realizados no Órgão.

### **4.8 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 4.8.1; gerar um relatório específico ou arquivo para o usuário por meio de recuperação de informações nas bases da aplicação, detalhado no item 4.8.2. A seção 4.8.3 considera os casos de reexecução de uma apuração especial.

Caso a apuração seja de correção de dados, devido a erros de funcionalidades de aplicações desenvolvidas pela contratada, observar as cláusulas contratuais com relação a garantias e prazos de correção.

#### **4.8.1 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, considera-se a contagem de Pontos de Função das funcionalidades desenvolvidas. Geralmente, estas funcionalidades são classificadas como Entradas Externas. É importante ressaltar que as funções de dados associadas aos dados atualizados **Não** devem ser contadas, considerando que não há mudanças nas estruturas dos Arquivos Lógicos.

$$PF = PF\_NÃO\_AJUSTADO$$

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:

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

#### 4.8.2 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. É importante ressaltar que as funções de dados associadas aos dados atualizados **Não** devem ser contadas, considerando que não há mudanças nas estruturas dos Arquivos Lógicos.

$$PF = PF\_NÃO\_AJUSTADO$$

#### 4.8.3 Reexecução de Apuração Especial

Em alguns casos a empresa contratante pode ter interesse em executar uma apuração especial mais de uma vez. Nestes casos, ela deve solicitar formalmente à contratada o armazenamento do script executado. Desta forma, se for solicitada a reexecução de uma apuração especial, esta deve ser dimensionada com aplicação de um fator considerando 10% na contagem de Pontos de Função da apuração especial em questão, da seguinte maneira:

$$PF = PF\_NÃO\_AJUSTADO \times 0,10$$

O percentual de multiplicação proposto no item acima é estimado podendo ser reajustado conforme avaliação da base histórica dos serviços realizados no Órgão.

### 4.9 Atualização de Dados

Em alguns casos, as demandas de correção de problemas em base de dados estão associadas à atualizações em um único registro. Por exemplo, atualização do nome de um Fornecedor cadastrado erradamente. Essa atualização realizada manualmente, sem a criação de script, enquadra-se na categoria de atualização de dados.

Nestes casos, a aferição do tamanho em Pontos de Função deve considerar 10% do PF\_ALTERADO de uma Entrada Externa, os Tipos de Dados da Entrada Externa são os campos atualizados.

$$PF = PF\_NÃO\_AJUSTADO \times 0,10$$

O percentual de multiplicação é estimado, podendo ser reajustado conforme avaliação da base histórica dos serviços realizados no Órgão.

#### **4.10 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 com conteúdo estático. Estas demandas são consideradas como desenvolvimento de consultas. Nestes casos, considera-se 50% dos Pontos de Função das consultas desenvolvidas. Por exemplo: alteração de página de estilo, criação de página html, atualização de menu, atualização de texto ou banner em páginas html existentes. Cada página é contada como uma consulta. As consultas são consideradas Consultas Externas Simples (3 PF).

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

As demandas de criação de Logomarcas ou identidade visual ou outras demandas de criação de Arte, associadas à área de Comunicação Social não são enquadradas nessa categoria. Tais demandas não se referem a contratos de Fábrica de Software, portanto não são consideradas neste Roteiro.

É recomendado a construção de Portais com ferramentas que apoiem a construção de conteúdo pelo usuário, por exemplo: Joomla, Zope/Plone, de modo a minimizar as demandas de criação de páginas estáticas.

O percentual de multiplicação proposto acima é estimado, podendo ser reajustado conforme avaliação da base histórica dos serviços realizados no Órgão.

#### **4.11 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 conforme cláusulas contratuais e documentadas no documento de estimativas do projeto.

O percentual de multiplicação proposto acima é estimado, podendo ser reajustado conforme avaliação da base histórica dos serviços realizados no Órgão.

#### **4.12 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 da contratada se mobilizará para encontrar as causas 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:

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

É importante ressaltar que a demanda de verificação de erros deve ser associada a uma funcionalidade específica. Os casos de sistema fora do ar por conta de problemas em rede ou banco de dados devem ser tratados como serviços de suporte e não de Fábrica de Software. Esses serviços de suporte não fazem parte do escopo desse Roteiro de métricas.

O percentual de multiplicação proposto acima é estimado, podendo ser reajustado conforme avaliação da base histórica dos serviços realizados no Órgão.

#### **4.13 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, isto é, 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 documentadas. Observe que estas funções farão parte do escopo do projeto de manutenção.

#### 4.14 Manutenção de Componentes

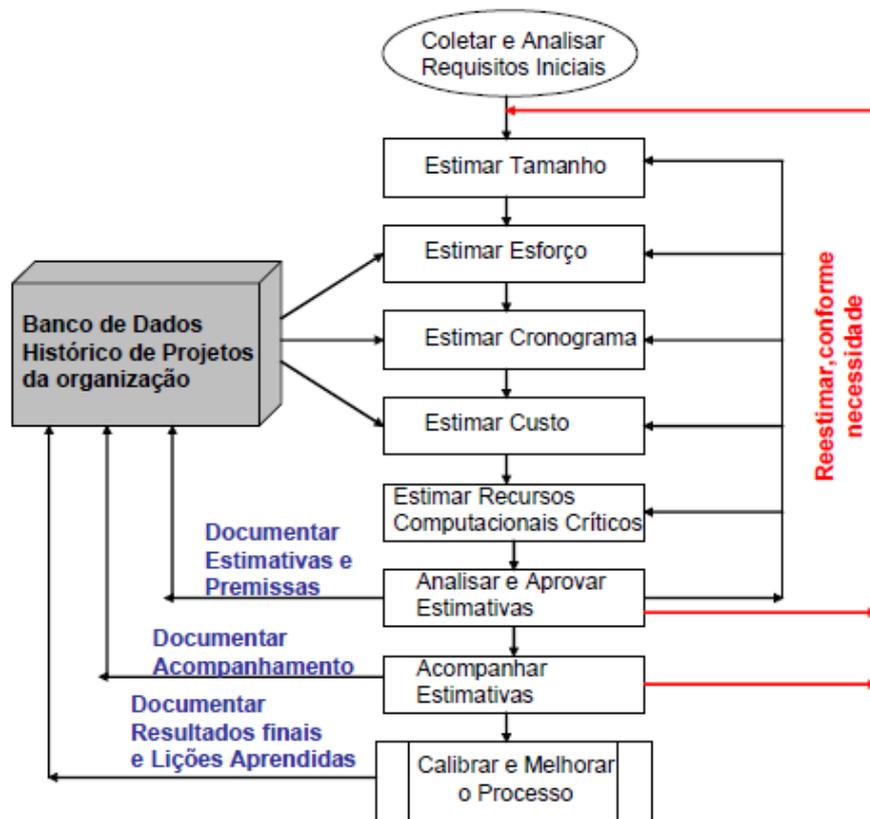
Em alguns casos são demandadas manutenções em componentes específicos de uma aplicação e estes são reusados por várias funcionalidades da aplicação. Por exemplo, suponha uma mudança em uma rotina de validação de um CPF usada em várias telas. Se considerarmos o método de contagem de projetos de melhoria do CPM, seriam contadas todas as funcionalidades impactadas por esta mudança. No entanto, este Roteiro propõe que o componente seja considerado um processo elementar independente e contado como uma funcionalidade. As funcionalidades que necessitem de teste devem ser requisitadas pela contratante e dimensionadas por meio da métrica Pontos de Função de Testes proposta em 4.12.

$$PF = PF\_NÃO\_AJUSTADO$$

### 5. 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) e a fórmula de Capers Jones para estimar os prazos dos projetos.

A **Figura 2** 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 2: 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 que 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 no capítulo 6. 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 entre contratante e contratada.

As seções seguintes apresentam os métodos de estimativas de tamanho, prazo, custo e esforço a serem utilizados nos projetos de software em contratos.

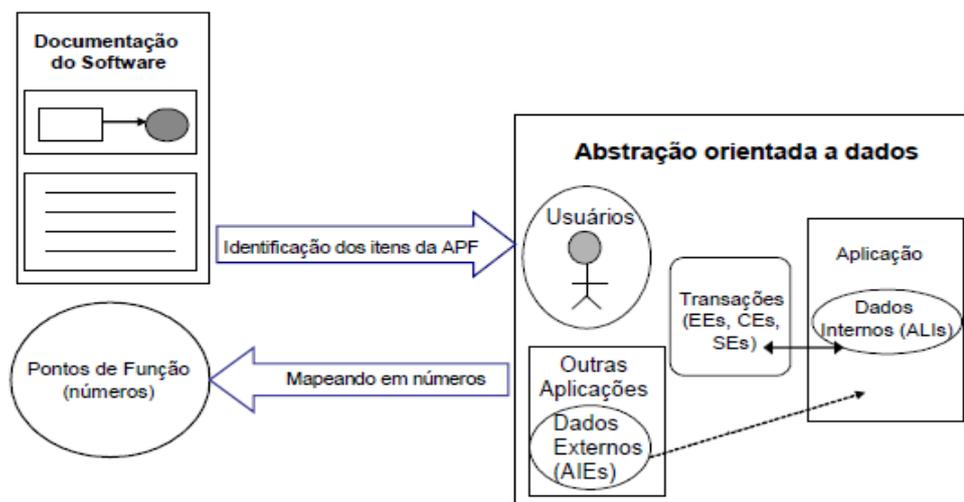
## 5.1 Contagem Estimativa de Pontos de Função (CEPF)

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, 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 (Tabela 1).

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 sequenciais pertencem a um mesmo processo elementar e funções independentes constituem processos elementares diferentes.



**Figura 3: 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 2** - 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**.

Nº ALIs Simples:	X 7 PF
Nº ALIs Médio:	X 10 PF
Nº ALIs Complexo:	X 15 PF
Total PF:	

**Tabela 2: Identificação dos Arquivos Lógicos Internos da Aplicação**

**Tabela 3** - 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.

Nº AIEs Simples:	X 5 PF
Nº AIEs Médio:	X 7 PF
Nº AIEs Complexo:	X 10 PF
Total PF:	

**Tabela 3: Identificação dos Arquivos de Interface Externa da Aplicação**

**Tabela 4** - 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**.

N° EEs Simples:	X 3 PF
N° EEs Média:	X 4 PF
N° EEs Complexa:	X 6 PF
Total PF:	

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

**Tabela 5** - 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, *listbox*, *download*, geração de um arquivo, geração de arquivo pdf, 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**.

N° CEs Simples:	X 3 PF
N° CEs Média:	X 4 PF
N° CEs Complexa:	X 6 PF
Total PF:	

**Tabela 5: Identificação das Consultas Externas da Aplicação**

**Tabela 6** - 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**.

N° SEs Simples:	X 4 PF
N° SEs Média:	X 5 PF

N° SEs Complexa:	X 7 PF
Total PF:	

**Tabela 6: Identificação dos Saídas Externas da Aplicação**

A Estimativa de tamanho do projeto em PFs deve ser gerada totalizando-se os PFs obtidos nas **Tabelas 2, 3, 4, 5 e 6**.

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

$$\text{PF\_DESENVOLVIMENTO} = \text{PF\_NÃO\_AJUSTADO}$$

**Observação:** O PF\_Conversão foi suprimido da fórmula. Nesse Roteiro é recomendado que o projeto de migração seja tratado como um novo projeto de desenvolvimento. Portanto, este deve ser estimado separadamente.

## 5.2 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. Neste Roteiro é adotado o modelo Simplificado de Estimativas.

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.

Cada órgão deverá possuir sua própria tabela de produtividade para cada linguagem, considerando-se sempre dados históricos dos projetos já realizados no órgão.

### 5.2.1 Distribuição de Esforço por Fase do Projeto

O próximo passo é a definição da distribuição de esforço pelas macro atividades do projeto, visando definir o valor agregado ao projeto após cada fase do ciclo de vida (**Tabela 7**).

Macro Atividades do Processo de Desenvolvimento de Software	Percentual de esforço (%)
Engenharia de Requisitos	25%

Design, Arquitetura	15%
Implementação	40%
Testes	10%
Homologação	5%
Implantação	5%

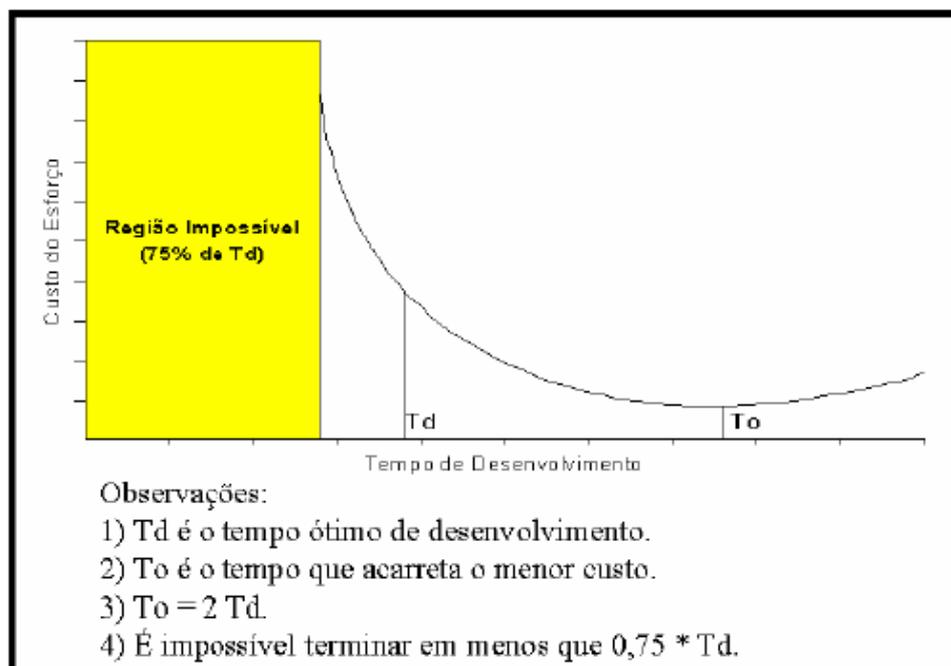
**Tabela 7: Distribuição de Esforço por Macro atividades do Projeto**

A Tabela 7 é uma sugestão de distribuição de esforço em projetos típicos, no entanto, em se tratando de 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 devem determinar o processo de desenvolvimento a ser seguido com percentual de esforço por fases.

### 5.3 Estimativa de Prazo de Projetos de Software

As estimativas de prazo não são lineares com o tamanho do projeto. O melhor tempo de 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 sugerido e utilizado nas estimativas de prazo deste manual. Jones [Jones, 2007] propõe uma fórmula para o cálculo do melhor tempo de desenvolvimento, denominado  $T_d$  e de Região Impossível (RI) de desenvolvimento (Figura 4). Na Região Impossível (RI), a adição de mais recursos ao projeto não implicará em redução no prazo. Note que a curva (Figura 4) mostra que quanto menor o prazo almejado para a conclusão do projeto, maior será o esforço requerido e conseqüentemente 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 4.



**Figura 4: Relação entre a Estimativa de Prazo e de Esforço**

O método utilizado para estimar o prazo dos projetos (Td) é baseado na fórmula de Capers Jones [Jones, 2007]. 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 8

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

**Tabela 8: 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 da definição de prazo máximo por tamanho funcional com base em dados históricos do órgão, conforme a Tabela 9.

Tamanho do Projeto	Prazo máximo (em dias úteis)
Até 10 PF	10 dias
De 11 PF a 20 PF	20 dias
De 21 PF a 30 PF	30 dias
De 31 PF a 40 PF	40 dias
De 41PF a 50 PF	50 dias
De 51 PF a 60 PF	60 dias
De 61 PF a 70 PF	70 dias
De 71 PF a 85 PF	88 dias
De 86 PF a 99 PF	104 dias

**Tabela 9. Estimativa de Prazo de Projetos menores que 100 PF**

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 seja necessário receber o projeto em um prazo menor que o tudo 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 custo de 20%; 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 70%. Não é recomendada a redução de prazo em mais de 20%.

Os percentuais de aumento de esforço são estimados, podendo ser reajustados conforme avaliação da base histórica dos serviços realizados no órgão.

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

## 5.4 Alocação de Equipe ao Projeto

Na alocação de equipe, deve ser considerada a estimativa de prazo e a de esforço. Sugere-se utilizar a fórmula 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.

## 5.5 Método para Estimativa de Custo

A estimativa de custo do projeto deve levar em consideração o custo de um ponto de função. A contratada já deverá considerar o custo da hora de todos os profissionais envolvidos no desenvolvimento da solução de software. O cálculo do custo do projeto (CP) será então da seguinte forma:

$$\text{CP} = \text{QPF} \times \text{CPF}$$

Onde:

**QPF** = Tamanho do Projeto em PF

**CPF** = Custo para implementar um Ponto de Função na plataforma em questão

## 5.6 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.

## 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, redução de cronograma e fator criticidade.

### 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%. Este percentual é sugerido, ficando a critério da instituição estabelecer este percentuais contratualmente. 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.

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 ( $5 \text{ PF} \times 80\% = 4 \text{ PFs}$ ).

## 6.2 Considerações sobre Projetos Cancelados

Em alguns casos, devido às mudanças no ambiente da contratante, uma demanda ou parte de um projeto de desenvolvimento ou manutenção pode ser cancelado a critério da contratante. 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 a tabela de distribuição de esforço contida 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 5.3.

Alguns projetos devido à legislação e a outros fatores externos já se iniciam com um prazo imposto. 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 inferior ao prazo calculado, então se deve considerar o seguinte:

- Redução de prazo de 10%: aumento de custo de 20% (projetos urgentes)
- Redução de prazo de 20%: aumento de custo de 50% (projetos críticos)
- Redução de prazo de 25%: aumento de custo de 70% (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% no custo de PF; aumento de esforço de 50% implica em aumento de 50% no custo de PF; o aumento de esforço de 70% implica em aumento de 70% no custo de PF.

## **6.4 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.

## **7. Contagem de Pontos de Função com Múltiplas Mídias e Dados de Código**

Este capítulo tem como propósito apresentar as diretrizes de Contagem de Pontos de Função 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]. Também é

abordado neste Capítulo algumas diretrizes para a mensuração de Dados de Código ou Code Data.

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 abordadas neste documento serão baseadas em *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 órgãos do SISP.

## **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, sugere-se 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, sugere-se que se utilize a abordagem *single instance* considerando que os dados impressos e os dados apresentados no arquivo de saída sejam idênticos e que a ferramenta de desenvolvimento apoie a geração dessas múltiplas saídas. 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. E ainda, se a geração das múltiplas saídas não seguirem o padrão da ferramenta de desenvolvimento e tiverem que ser customizadas para o cliente, então será utilizada a abordagem *multiple instance*.

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. Sugere-se que seja 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.

#### **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. Geralmente se utiliza 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. Lembrando que caso o projeto é claro o suficiente para dizer que o desenvolvimento é o mesmo, poderá ser utilizada a abordagem *single instance*.

#### **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*, considera-se 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 formato de valores separados por vírgula, então se contará apenas uma vez, observando que a funcionalidade será da ferramenta e não da aplicação.

#### **7.6 Dimensionamento de Dados de Código**

As Tabelas com atributos de Código e Descrição devem ser analisadas com muito cuidado. Caso estas tabelas não sejam mantidas pela aplicação, o que geralmente acontece, elas não serão contadas.

Entretanto, caso seja justificadamente necessário manter estas informações, elas serão consideradas requisitos funcionais do usuário. Assim, caso sejam requisitados Casos de Uso e a implementação de funcionalidades para manter tais tabelas, elas serão contadas como dados de negócio (Arquivo Lógico Interno).

### **8. 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 da contratada, 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 da contratada, 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 da contratada, 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 da contratada, 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 administradores de dados (ADs) com um número de profissionais definido junto a contratante, 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 da contratada. 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 da contratada 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 contratante. 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 contratante, 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. No entanto, por falta de pessoal, alguns Órgãos tem contratado estas atividades. Essas atividades antecedem a fase de requisitos – primeira fase do processo de software e devem ser faturadas em horas de consultoria.

Nesta atividade, deve ser gerado um Documento de Visão (DV) ou outro documento de requisitos de negócios, Documento de Requisitos não funcionais e Glossário (opcional). O DV é o artefato utilizado como insumo para o Planejamento do Projeto - estimativa de tamanho funcional (em Pontos de Função) do projeto e para o processo de desenvolvimento - atividade de Engenharia de Requisitos do processo de desenvolvimento de software.

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

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

A revisão deste guia será feita sempre que se 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 documentar a novidade em documentos posteriores, complementando este Roteiro.

## 9.2 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 pela contratante e contratada para se decidir sobre a atualização do guia.

## 10. Conclusão

Este documento apresentou um guia para o dimensionamento de tamanho de todos os tipos de projetos de software da contratante, visando a aderência de todos os projetos desenvolvidos na instituição 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).

É importante ressaltar que o uso de métricas em contrato de software é uma excelente prática, visando proporcionar uma gestão efetiva dos contratos com base em dados quantitativos e objetivos. A implantação desta modalidade de contrato implica na definição de processos de gestão de requisitos e de gestão de projetos baseados nas melhores práticas dos modelos de Qualidade de Software. Outro ponto a ser destacado é a implantação de um Escritório de Métricas com Servidores capacitados para realizar contagens de Pontos de Função e Estimativas. Estes Servidores serão responsáveis pela revisão das contagens de Pontos de Função e Estimativas realizadas pelo Escritório de Métricas da empresa contratada e pela manutenção do Roteiro de Métricas do Órgão.

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.



## Referências Bibliográficas

- [Boehm, 2000] BOEHM, B.W. **Software Cost Estimation With COCOMO II**. Prentice Hall, New Jersey, 2000.
- [Dekkers, 2003] DEKKERS, C. “Measuring the logical or functional” Size of Software Projects and Software Application”. Spotlight Software, ISO Bulletin May 2003, pp10-13.
- [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 fo 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 - Formalização Simples de Requisitos

Dados Gerais	
Número da OS	
Descrição da OS	
Nome do Sistema Mantido	
Tecnologia Utilizada	
Responsável pela Manutenção:	<nome do desenvolvedor, telefone de contato e e-mail>
Data do Início do Serviço	DD/MM/AAAA
Data do Término do Serviço	DD/MM/AAAA

### Descrição do Serviço a ser Executado

Descrição dos Requisitos	Critérios de Aceite
1.	1.1 1.2...
2.	2.1 2.2...
N.....	N.1... N.2...

Anexar a Tela da Funcionalidade (Opcional)

Identificação da Manutenção:

<b>Tipo</b>	
<input type="checkbox"/>	Projeto de Melhoria – Criação de Funcionalidades
<input type="checkbox"/>	Projeto de Melhoria – Alteração de Funcionalidades
<input type="checkbox"/>	Projeto de Melhoria – Exclusão de Funcionalidades
<input type="checkbox"/>	Projetos de Migração de Dados
<input type="checkbox"/>	Garantia – Manutenção Corretiva em Funcionalidade no Prazo de Garantia (sem custo)
<input type="checkbox"/>	Manutenção Corretiva – Funcionalidade Não Desenvolvida pela Contratada ou Fora do Prazo de Garantia
<input type="checkbox"/>	Redesenvolvimento de Projetos em outra Plataforma
<input type="checkbox"/>	Atualização de Plataforma
<input type="checkbox"/>	Manutenção em Interface (Cosmética)
<input type="checkbox"/>	Adaptação de Funcionalidades a Requisitos Não Funcionais
<input type="checkbox"/>	Apuração Especial – Base de Dados
<input type="checkbox"/>	Apuração Especial – Geração de Relatórios
<input type="checkbox"/>	Reexecução de Apuração Especial
<input type="checkbox"/>	Atualização de Dados
<input type="checkbox"/>	Manutenção em Páginas Estáticas de <i>Intranet, Internet ou Portal</i>
<input type="checkbox"/>	Manutenção de Documentação de Sistemas Legados
<input type="checkbox"/>	Verificação de Erros
<input type="checkbox"/>	Pontos de Função de Teste (Execução de testes em funcionalidades fora do escopo da manutenção)
<input type="checkbox"/>	Manutenção de Componentes

Observações relevantes quanto ao tipo de manutenção:

--

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

a) Tabelas Modificadas pela Manutenção

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

b) Entradas de Dados Afetadas pela Manutenção (telas ou arquivos de carga) ou apuração especial de atualização de Base de Dados

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

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

**c Sim      c 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.

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

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

**c Sim      c Não**

d) Relatórios Afetados pela Manutenção ou Apuração Especial – Extração de Dados para arquivo ou Geração de arquivos

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

<b>Nome do Relatório</b>	
<b>Total de Campos no Relatório</b>	
<b>Tabelas Acessadas</b>	
<b>Total de Campos Afetados =</b>	
<b>Total de Campos Calculados ou Totalizadores =</b>	
<b>Existe atualização de dados (log, indicador...) c Sim    c Não</b>	
<b>Campos Incluídos/Alterados/Excluídos</b>	

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

**c Sim    c Não**