



**Instituto Brasileiro de Meio Ambiente e dos  
Recursos Naturais Renováveis – Ibama**

# **ROTEIRO DE MÉTRICAS DE SOFTWARE DO IBAMA**

**Versão 3.1**

## Histórico de Revisão

<b>Data</b>	<b>Versão</b>	<b>Descrição</b>	<b>Autor(es)</b>
07/12/15	1.0	Criação do documento	<ul style="list-style-type: none"> <li>• José Augusto Luís de Carvalho</li> <li>• Loidianne Alves Marinho</li> <li>• Márcio Pereira Lima</li> </ul>
28/07/16	1.1	Revisão do documento para contemplar atualizações da versão 2.2 do Roteiro de Métricas do SISP.	<ul style="list-style-type: none"> <li>• José Augusto Luís de Carvalho</li> <li>• Loidianne Alves Marinho</li> <li>• Márcio Pereira Lima</li> </ul>
19/06/17	1.2	Atualizações decorrentes de: advento da Portaria STI/MP nº 04/2017; edição do roteiro de métricas para projetos de data warehouse do SISP; Ajustes para maior clareza quanto ao uso de múltiplas mídias.	<ul style="list-style-type: none"> <li>• Márcio Pereira Lima</li> <li>• Loidianne Alves Marinho</li> <li>• José Augusto Luís de Carvalho</li> </ul>
09/08/19	2.0	Atualização do item Aferição de Pontos de Função em Ambiente Baseado em Geotecnologias	<ul style="list-style-type: none"> <li>• Rosana de Souza Ribeiro</li> </ul>
06/09/19	2.0	Atualização do Roteiro de Métricas do Ibama.	<ul style="list-style-type: none"> <li>• Mosar Rodrigues Rabelo Júnior</li> <li>• Liliane Pereira dos Santos</li> </ul>
03/05/21	3.0	Atualização do Roteiro de Métricas do Ibama	<ul style="list-style-type: none"> <li>• Felipe Ventorim Rodrigues Paulin</li> <li>• Liliane Pereira dos Santos</li> <li>• Mosar Rodrigues Rabelo Júnior</li> </ul>
18/11/21	3.1	Evolução das contagens envolvendo retrabalho no item 7.2.1 do roteiro.	<ul style="list-style-type: none"> <li>• Felipe Ventorim Rodrigues Paulin</li> <li>• Liliane Pereira dos Santos</li> </ul>

## Sumário

1. Introdução .....	6
2. Objetivo .....	8
8 Contagem de Pontos de Função .....	8
3.1 Determinar Propósito, Tipo e Escopo da Contagem e Fronteira da Aplicação .....	9
3.2 Identificar Funções de Dados e Funções Transacionais .....	10
3.3 Calcular Tamanho Funcional.....	11
4. Cálculo de Pontos de Função para o Ibama .....	12
4.1 Projeto de Desenvolvimento .....	13
4.2 Projeto de Melhoria .....	13
4.3 Projeto de Migração de Dados .....	16
4.4 Manutenção Corretiva .....	17
4.5 Mudança de Plataforma .....	18
4.5.1 Mudança de Plataforma – Linguagem de Programação .....	18
4.5.2 Mudança de Plataforma – Banco de Dados .....	19
4.6 Atualização de Versão .....	20
4.6.1 Atualização de Versão – Linguagem de Programação .....	20
4.6.2 Atualização de Versão – Browser .....	21
4.6.3 Atualização de Versão – Banco de Dados.....	21
4.7 Manutenção em Interface .....	22
4.8 Adaptação em Funcionalidades sem Alteração de Requisitos Funcionais .....	23
4.9 Apuração Especial .....	24
4.9.1 Apuração Especial – Base de Dados .....	25
4.9.2 Apuração Especial – Geração de Relatórios .....	26
4.9.3 Apuração Especial – Reexecução .....	26
4.10 Atualização de Dados .....	27
4.11 Manutenção de menus, telas e listas de seleção estáticas .....	27
4.12 Manutenção de Documentação de Sistemas Legados .....	27
4.13 Verificação de Erros .....	28
4.14 Pontos de Função de Teste .....	29
4.15 Componente Interno Reusável .....	29
5. Orientações Complementares para Contagem .....	30
5.1 Divisão de um mesmo Processo Elementar em Transações Distintas .....	30
5.2 Contagem de Pontos de Função com Múltiplas Mídias .....	31
5.2.1 Cenário 1: Mesmos dados apresentados em tela e impressos .....	32
5.2.2 Cenário 2: Mesmos dados de saída como dados em arquivo e relatório impresso .....	33
5.2.3 Cenário 3: Mesmos dados de entrada batch e on-line .....	33
5.2.4 Cenário 4: Múltiplos canais de entrega da mesma funcionalidade .....	34
5.2.5 Cenário 5: Relatório em múltiplos formatos .....	34
5.3 Log, Trilha de Auditoria e Histórico .....	35
5.3.1 Log .....	35
5.3.2 Trilha de Auditoria .....	35
5.3.3 Histórico .....	36
5.4 Aferição de Pontos de Função em ambiente BPMS .....	37
5.4.1 Fronteira de Aplicação .....	38

5.4.2	Critérios de Contagem .....	39
5.5	Aferição de Pontos de Função em ambiente de Apoio à Decisão (Business Intelligence e Data Warehouse) .....	40
5.5.1	Introdução .....	40
5.5.2	Modelo de Arquitetura de Referência de Data Warehouse / Data Mart .....	40
5.5.3	Estimativa de Projeto de Data Warehouse .....	44
5.5.4	Contagem de Projeto de Data Warehouse.....	45
5.5.3.1	Contagem de Arquivos Lógicos Internos (ALI) .....	47
5.5.3.2	Contagem de Arquivos de Interface Externa (AIE) .....	48
5.5.3.3	Contagem de Funções Transacionais (EE, CE e SE) .....	48
5.5.3.4	Regras Resumo sobre Contagem em DW .....	50
5.5.3.5	Considerações sobre Contagem em DW .....	51
5.5.3.6	Atividades Sem Contagem de Pontos de Função .....	55
5.5.5	Fronteira de Aplicação .....	56
5.5.6	Critérios de Contagem .....	57
5.6	Aferição de Pontos de Função em Ambiente Baseado em Geotecnologias .....	59
5.6.1	Medição de camadas georreferenciadas .....	59
5.6.2	Apresentação de dados em formatos diferentes (multiple instance) .....	60
5.6.3	Apresentação simultânea de vários elementos visuais .....	60
5.6.4	Funcionalidades complexas de georreferenciamento .....	62
6.	Considerações Especiais para Planejamento e Acompanhamento de Projetos .....	63
6.1	Diretrizes para Planejamento: Estimativas de Projetos de Software .....	63
6.1.1	Contagem Estimativa de Pontos de Função (CEPF) .....	65
6.1.2	Distribuição de Esforço por Fase do Projeto .....	70
6.2	Definição de Prazo de Projetos de Software .....	71
Diretrizes para Acompanhamento de Projetos.....	72	
6.2.1	Considerações sobre Mudança de Requisitos .....	72
6.2.2	Considerações sobre Projetos Cancelados .....	73
6.2.3	Considerações sobre Redução de Cronograma .....	74
7.	Atividades Sem Contagem de Pontos de Função .....	75
8.	Processo de Revisão do Roteiro de Contagem .....	77
8.1	Revisão para Correção de Inconsistências e Situações não Previstas .....	77
8.2	Revisão para Adoção de Novas Versões do CPM e do Roteiro de Métricas de Software do SISP .....	77
9.	Dicas para otimizar o custo das manutenções .....	78
10.	Conclusão .....	78
11.	Referencias Bibliográficas .....	79
Orientações para preenchimento da Planilha de Contagem .....	80	
ANEXO 1 – Identificação da Planilha de Contagem de Ponto de Função .....	80	
ANEXO 2 – Planilha de Contagem de Ponto de Função .....	81	
ANEXO 3 – Fundamentação da Contagem de Ponto de Função .....	83	

## Índice de tabelas

Tabela 1: Contribuição Funcional dos Tipos Funcionais (Fonte: CPM 4.3.1) .....	11
Tabela 2: Resumo da contagem estimada em PF de projeto de DW .....	45
Tabela 3: Identificação dos Arquivos Lógicos Internos da Aplicação.....	68
Tabela 4: Identificação dos Arquivos de Interface Externa da Aplicação .....	68
Tabela 5: Identificação das Entradas Externas da Aplicação .....	69
Tabela 6: Identificação das Consultas Externas da Aplicação .....	69
Tabela 7: Identificação das Saídas Externas da Aplicação .....	70
Tabela 8: Distribuição de Esforço por Macroatividades do Projeto .....	71
Tabela 9: Definição de Prazo de Projetos para OS de desenvolvimento ou manutenção evolutiva.	72
Tabela 10: Percentuais de redução de prazo e aumento de custo. ....	75

## 1. Introdução

O Instituto Brasileiro do Meio Ambiente e dos Recursos Naturais Renováveis – Ibama – têm utilizado a métrica Ponto de Função (PF) nas estimativas e dimensionamento de tamanho funcional de projetos de software devido aos diversos benefícios de utilização desta métrica, destacando-se: regras de contagem objetivas, independência da solução tecnológica utilizada e facilidade de estimativa nas fases iniciais do ciclo de vida do software.

Por permitir a medição objetiva de serviços de desenvolvimento de soluções de software, a utilização da métrica Pontos de Função é uma boa prática na contratação de serviços e está aderente ao estabelecido na Instrução Normativa n.º 04/2014 STI/MP, na Portaria nº 04/2017 STI/MP e na Instrução Normativa SGD/ME n.º 1/2019.

Esta Portaria, no seu Art. 3º, também determina que os órgãos integrantes do Sistema de Administração dos Recursos de Tecnologia da Informação (SISP) elaborem Roteiro ou modelo de métricas o qual determinará as regras de contagem específicas para aplicação no órgão.

O Manual de Práticas de Contagem de Pontos de Função (CPM 4.3.1) [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 Ponto de Função foi concebida como uma medida de tamanho funcional para projetos de desenvolvimento e de melhoria (manutenção evolutiva) de software.

Além disso, a contagem de pontos de função é baseada no projeto lógico da aplicação (logical design). Nas fases iniciais do ciclo de vida do software, o insumo para a definição das estimativas do projeto é um documento inicial de requisitos, por exemplo: documento de visão ou algum outro tipo de especificação elaborada pelo analista de negócios. Assim, torna-se importante o estabelecimento de métodos para estimar o tamanho 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 a partir do tamanho funcional estimado do projeto. Considerando que o objetivo principal do CPM não é esse, o Roteiro de Métricas de Software do Ibama foi criado para contemplar questões não abordadas pelo manual do IFPUG, mas vivenciadas pelo Instituto, como a utilização da métrica em contratos administrativos.

O restante deste documento encontra-se organizado da seguinte forma:

- o capítulo 2 descreve os objetivos e as referências consultadas para a elaboração deste documento;
- o capítulo 3 apresenta algumas definições básicas para a contagem de pontos de função;

- o capítulo 4 define métricas baseadas em Ponto de Função para dimensionar projetos de desenvolvimento e vários tipos de projetos de manutenção de software; o capítulo 5 estabelece diretrizes para contagem de múltiplas mídias; o capítulo 6 define um processo de estimativas e recomendações para o gerenciamento de projetos contratados com base em métricas, tais como: tratamento de mudanças de requisitos, projetos cancelados e redução de cronograma; o capítulo 7 apresenta algumas atividades que não devem ser consideradas nas contagens de pontos de função;
- o capítulo 8 apresenta o processo de revisão deste guia de contagem;
- o capítulo 9 dicas para otimizar o custo das manutenções;
- o capítulo 10 conclui o documento.

## 2. Objetivo

Este documento tem como objetivo principal apresentar um roteiro de métricas para ser utilizado na mensuração de softwares, sistemas e aplicativos do Ibama, incluindo as estimativas de tamanho de projetos de software e para utilização em contratos administrativos relacionados a desenvolvimento e manutenção de software.

Para fins de contagem de pontos de função, neste documento os termos software e sistema são utilizados indistintamente, representando todos os seus derivados, como sistemas de informação, aplicativos, programas, web services, entre outros.

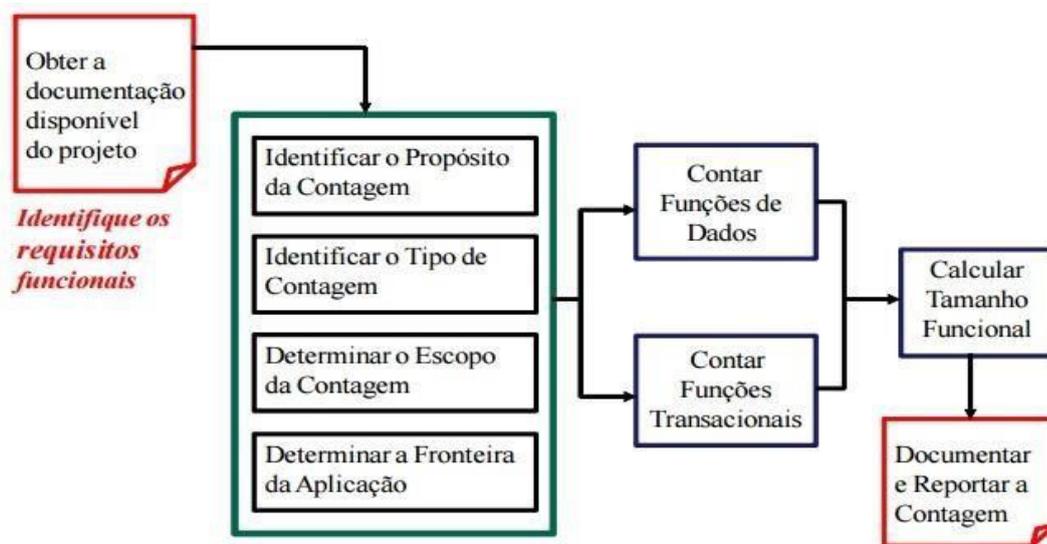
Este documento usa como base o roteiro de métricas do SISP 2.2 e as regras de contagem de pontos de função do Manual de Práticas de Contagem (CPM 4.3.1) para vários tipos de projetos de desenvolvimento e de manutenção de software, promovendo o uso de métricas objetivas nos contratos de prestação de serviços desses projetos.

## 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. O tamanho funcional é definido como “tamanho do software derivado pela quantificação dos requisitos funcionais do usuário” [Dekkers, 2003].

A métrica PF é independente da metodologia e tecnologia utilizadas. 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 de função, com base na quantificação das funcionalidades solicitadas e entregues, 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, 2010b] 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.



*Procedimento de Contagem de Pontos de Função*

### 3.1 Determinar Propósito, Tipo e Escopo da Contagem e Fronteira da Aplicação

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 auxiliar o processo de contratação do mesmo.

Com base no propósito da contagem são definidos o escopo da contagem e o tipo de contagem. O escopo da contagem identifica quais funcionalidades serão incluídas na contagem de pontos de função, e o tipo de contagem identifica se o projeto é de desenvolvimento, de melhoria ou aplicação instalada.

A fronteira da aplicação, que é a interface conceitual que indica o limite lógico entre o sistema sendo medido e os usuários (também entre outras aplicações), deve ser definida com base na visão do usuário, desconsiderando questões de implementação. Deve-se ressaltar que toda contagem de pontos de função é realizada dentro de uma fronteira estabelecida.

### 3.2 Identificar 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 dos tipos funcionais para cada requisito funcional definido no documento de requisitos do sistema, 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 identificação e a avaliação das complexidades dos tipos funcionais não podem ser realizadas de maneira subjetiva. A contagem de pontos de função deve seguir rigorosamente as regras de contagem do CPM e as definições complementares deste roteiro de métricas.

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

### 3.3 Calcular Tamanho Funcional

O Manual de Práticas de Contagem do IFPUG define dois tipos de projetos de software, a saber:

**Projeto de Desenvolvimento:** projeto para desenvolver e entregar a primeira versão de uma aplicação de software. Seu tamanho funcional é a medida das funcionalidades entregues ao usuário no final do projeto. Também considera-se as funcionalidades de conversão de dados, caso seja requisitado no projeto a migração ou carga inicial de dados para a nova aplicação.

**Projeto de Melhoria:** projeto de manutenção evolutiva ou melhoria funcional. Seu tamanho funcional é a medida das funcionalidades incluídas, alteradas e excluídas ao final do projeto. Também considera-se as funcionalidades de conversão de dados, caso seja requisitado a migração ou carga inicial de dados no projeto de melhoria.

Seguem abaixo as definições dos termos técnicos da Análise de Pontos de Função utilizados nas fórmulas de dimensionamento de projetos de software propostas neste roteiro:

**PF\_INCLUÍDO:** pontos de função associados às novas funcionalidades que farão parte da aplicação após um projeto de desenvolvimento ou de manutenção.

**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 desenvolvimento ou de manutenção. Exemplos de funções de conversão incluem: migração ou carga inicial de dados para popular as novas tabelas criadas (Entradas Externas) e relatórios associados à migração de dados, caso requisitado pelo usuário (Saídas Externas ou Consultas Externas). Observe que os dados carregados em um processo de migração não devem ser contados como Arquivos de Interface Externa.

Quando for caracterizado um esforço relativamente maior da atividade de conversão de dados em projetos de desenvolvimento e de melhoria, a atividade deve ser tratada como projeto separado de migração de dados, conforme seção 4.3. Por exemplo, os projetos que envolvem a migração de dados de banco de dados hierárquico para banco de dados relacional e o tratamento de funções complexas de migração de dados. Nesses casos, recomenda-se tratá-los como projetos separados de migração de dados, descritos na seção 4.3.

## 4. Cálculo de Pontos de Função para o Ibama

Este capítulo tem como propósito descrever os diversos tipos de projetos de software e definir métricas para seu dimensionamento baseado nas regras de contagem de pontos de função do CPM.

Quanto à documentação de projetos de manutenção, 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 e futuras manutenções. É importante também documentar as estimativas e a contagem de pontos de função.

Quando não houver necessidade de contratar todas as fases do ciclo de vida do software a contratada será remunerada pela contagem de pontos de função considerando apenas os percentuais das fases contratadas, conforme os níveis percentuais sugeridos na Tabela 7. Exemplo: para um novo projeto de desenvolvimento de um sistema de treinamentos, que não exista a intenção de contratar a fase de requisito, a contratada será remunerada pela contagem de pontos de função desconsiderando o percentual dessa fase.

### 4.1 Projeto de Desenvolvimento

Projeto de Desenvolvimento é o projeto para desenvolver e entregar a primeira versão de uma aplicação de software. Seu tamanho funcional é a medida das funcionalidades entregues ao usuário no final do projeto. Também consideram-se as funcionalidades de conversão de dados. Segue a fórmula de cálculo utilizada no dimensionamento de projetos de desenvolvimento de software:

$$PF\_DESENVOLVIMENTO = PF\_INCLUIDO + PF\_CONVERSÃO$$

Quando for caracterizado um esforço relativamente maior da atividade de conversão de dados em projetos de desenvolvimento, a atividade deve ser tratada como projeto separado de migração de dados, conforme seção 4.3.

### 4.2 Projeto de Melhoria

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

Segundo o padrão IEEE Std 1219 [IEEE, 1998], esta manutenção seria um tipo de manutenção adaptativa, definida como: modificação de um produto de software existente para mantê-lo funcionando adequadamente em um ambiente que sofre 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 4.3.1.

Este roteiro separa o projeto de melhoria (quando as mudanças são associadas aos requisitos funcionais) do projeto de 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 cálculo utilizada no dimensionamento de projetos de melhoria de software:

$$PF\_MELHORIA = PF\_INCLUIDO + (FI \times PF\_ALTERADO) + (0,30 \times PF\_EXCLUIDO) + PF\_CONVERSÃO$$

**FI (Fator de Impacto)** pode variar de 50% a 90% conforme condições abaixo:

**FI = 50%** para **funcionalidade de sistema desenvolvida ou mantida** por meio de um projeto de melhoria pela empresa contratada.

FI = 75% para funcionalidade de sistema não desenvolvida ou mantida por meio de um projeto de melhoria pela empresa contratada e sem necessidade de redocumentação da funcionalidade.

**FI = 90%** para **funcionalidade de sistema não desenvolvida ou mantida** por meio de um projeto de melhoria pela empresa contratada e **com necessidade de redocumentação da funcionalidade**. FI = 90% representa a adição de 15% como fator de redocumentação ao Fator de Impacto anterior (75%). Nesse caso, **a contratada deve redocumentar a funcionalidade mantida, gerando a documentação completa da mesma, aderente ao processo de software da contratante**. 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 90% apenas será considerado na primeira demanda de projeto de melhoria em cada funcionalidade.

Este roteiro define um fator de redocumentação menor para projetos de manutenção (melhoria, corretiva e adaptativa) do que o fator proposto em projetos específicos de redocumentação (seção 4.12 deste roteiro). Isso porque, em projetos de manutenção de uma funcionalidade sem documentação, é necessário realizar o entendimento da funcionalidade para poder modificá-la e testá-la, ou seja, é necessário realizar a engenharia reversa da funcionalidade para executar os testes corretamente. Assim sendo, a redocumentação requisitada em projetos de melhoria requer um esforço menor do que em projetos de redocumentação, descritos na seção 4.12, onde é

necessário remunerar todo o esforço de engenharia reversa e a atividade de documentação. Em projetos de manutenção, o fator de 15% está remunerando apenas a atividade de documentação.

Os percentuais de FI acima correspondem à contratação de todas as fases do processo de desenvolvimento de software. Caso alguma fase não seja contratada, deve-se aplicar ao FI um redutor que corresponde ao percentual da fase não contratada, conforme percentuais sugeridos na Tabela 7.

Quando for caracterizado um esforço relativamente maior da atividade de conversão de dados em projetos de melhoria, a atividade deve ser tratada como projeto separado de migração de dados, conforme seção 4.3.

Seguem algumas considerações importantes para serem analisadas em projetos de melhoria.

#### Observação 1: Função Alterada

Uma função de dados (Arquivo Lógico Interno ou Arquivo de Interface Externa) é considerada alterada quando houver inclusão ou exclusão de Tipos de Dados (TD). De acordo com o glossário do CPM 4.3.1, um Tipo de Dados (DET – *Data Element Type*) é um atributo único, reconhecido pelo usuário e não repetido. 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 4.3.1 define **lógica de processamento** como requisitos especificamente solicitados pelo usuário para completar um processo elementar. Esses requisitos devem incluir uma ou mais das 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 ou 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 como função alterada qualquer mudança em funcionalidades da aplicação devido às mudanças de regras de negócio. Por exemplo, uma funcionalidade de cadastro envolvia a inclusão de um telefone do gerente. Devido a mudanças no processo de negócio, a funcionalidade deve sofrer uma manutenção para cadastrar dois telefones do gerente. Desta forma, o roteiro considera esta função como uma Entrada Externa alterada, PF\_ALTERADO em um projeto de melhoria, mesmo que não existam mudanças de lógica de processamento, de tipos de dados ou de arquivos referenciados.

Serão tratadas como manutenções adaptativas apenas as manutenções que implicarem exclusivamente em 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.

### 4.3 Projeto de Migração de Dados

Os projetos de migração de dados devem ser contados como um novo projeto de desenvolvimento de um sistema, seguindo a fórmula abaixo:

$$\text{PF\_CONVERSÃO} = \text{PF\_INCLUIDO}$$

Um projeto de migração deve contemplar minimamente: os ALI mantidos pela migração, as

Entradas Externas – considerando as cargas de dados nos ALI – e, caso seja solicitado pelo usuário, os relatórios gerenciais das cargas, que serão contados como Saídas Externas. Todas as contagens de PF devem ser realizadas com base nas funcionalidades requisitadas e recebidas pelo usuário.

## 4.4 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 de sistemas.

Quando o sistema em produção tiver sido desenvolvido pela contratada, a manutenção corretiva será do tipo Garantia se estiver no período de cobertura e em conformidade com as demais condições de garantia previstas em contrato.

Quando o sistema estiver fora da garantia ou não tenha sido desenvolvido pela empresa contratada, deverá ser estimado e calculado o tamanho do projeto de manutenção corretiva. Nestes casos, a aferição do tamanho em pontos de função da funcionalidade ou das funcionalidades corrigidas deve considerar um fator de impacto (FI) sobre o PF\_ALTERADO.

$$\text{PF\_CORRETIVA} = \text{FI} \times \text{PF\_ALTERADO}$$

Fator de Impacto (FI):

50% quando estiver fora da garantia e a correção for feita pela mesma empresa que desenvolveu a funcionalidade.

75% quando estiver fora da garantia e a correção for feita por empresa diferente daquela que desenvolveu a funcionalidade.

As demandas de manutenção corretiva não contemplam atualização de documentação da funcionalidade corrigida, pois este roteiro considera que, normalmente, manutenção corretiva não se refere a erros de requisitos. Caso seja erro em requisitos, essa demanda deve ser tratada como projeto de melhoria (alteração de funcionalidade), descrito na seção 4.2. Porém, quando o erro for causado por documentação dúbia ou imprecisa (elaborada pela contratada) da funcionalidade corrigida, a manutenção corretiva poderá contemplar os ajustes na documentação, mesmo fora da garantia.

Caso seja demandada a redocumentação da funcionalidade corrigida, porque a documentação não existe ou está desatualizada, deve-se adicionar ao FI um fator de redocumentação de 15%, conforme descrito na seção 4.2.

## **4.5 Mudança de Plataforma**

São considerados nesta categoria, projetos que precisam ser migrados para outra plataforma. Por exemplo, um sistema legado em COBOL que necessita ser redesenvolvido em JAVA; o banco de dados de um sistema legado que precisa ser migrado para o DB2.

Recomenda-se enfaticamente a realização da análise de impacto das mudanças propostas, para efeito de determinação do percentual adequado para aplicação sobre o total de pontos de função das funcionalidades impactadas. Por exemplo, em uma análise de impacto pode ser identificado que não haverá mudanças no código-fonte ou em função transacional, sendo necessário apenas testar o sistema, então deve-se utilizar um percentual contemplando apenas a fase de testes. No caso do teste apontar a necessidade de atualizar alguma função transacional, não deve ser contado o esforço do teste, mas sim o esforço abordado nesta seção, conforme as fórmulas apresentadas nos tópicos seguintes.

As próximas subseções apresentam os tipos de projetos de mudança de plataforma. Os projetos de mudança de plataforma que se enquadram em mais de uma subseção, devem ser contados apenas uma vez, considerando o tipo de projeto com maior contagem de pontos de função.

### **4.5.1 Mudança de Plataforma – Linguagem de Programação**

Nesta categoria encontram-se as demandas de redesenvolvimento de sistemas em outra linguagem de programação. Como os projetos legados, frequentemente, não possuem documentação, devem ser considerados como novos projetos de desenvolvimento. Assim, será utilizada a fórmula de projetos de desenvolvimento do CPM 4.3.1.

Observe que caso não exista mudança nas funções de dados, ou seja, o banco de dados da aplicação seja mantido, as funções de dados não devem ser consideradas para fins de pagamento. No entanto, nesse caso, deve ser realizada a contagem das funções de dados a fim de compor a documentação da contagem final do projeto.

Outro ponto a ser observado são as fases contratadas. Caso o projeto já possua documentação de requisitos, a fase de requisitos não será contratada. Deve-se considerar apenas os percentuais das fases contratadas.

### **PF\_REDESENVOLVIMENTO\_LINGUAGEM = PF\_INCLUÍDO + PF\_CONVERSÃO**

Quando for caracterizado um esforço relativamente maior da atividade de conversão de dados em projetos de redesenvolvimento, a atividade deve ser tratada como projeto separado de migração de dados, conforme seção 4.3.

## **4.5.2 Mudança de Plataforma – Banco de Dados**

Nesta categoria encontram-se as demandas de redesenvolvimento de sistemas para utilizar um outro sistema gerenciador de banco de dados.

Observe que caso não exista mudança nas funções de dados, ou seja, o banco de dados da aplicação seja mantido, então as funções de dados não devem ser consideradas para fins de pagamento. No entanto, nesse caso, deve ser realizada a contagem das funções de dados a fim de compor a documentação da contagem final do projeto.

Caso a demanda de redesenvolvimento seja de um sistema gerenciador de banco de dados relacional para outro relacional, deve ser utilizada a seguinte fórmula:

$$\text{PF\_REDESENVOLVIMENTO\_BD\_RELACIONAL} = (\text{PF\_ALTERADO} \times 0,30) + \text{PF\_CONVERSÃO}$$

O PF\_ALTERADO deve considerar apenas as funcionalidades impactadas. As funcionalidades que possuem apenas demandas de testes, devem ser contadas usando o percentual da fase de testes (ver Tabela 7).

Nos projetos de redesenvolvimento de banco de dados relacional para outro relacional, recomenda-se tratar o PF\_CONVERSÃO dentro do mesmo projeto.

Na mudança de banco relacional para relacional, geralmente a estrutura de dados não é alterada, desta forma não contamos as funções de dados.

## **4.6 Atualização de Versão**

São consideradas nesta categoria, demandas para uma aplicação existente – ou parte de uma aplicação existente – executar em versões diferentes de *browsers* (ex: *Internet Explorer*, *Firefox*, *Chrome*, etc) ou de linguagens de programação (ex: versão mais atual do JAVA). Também são consideradas nesta categoria atualização de versão de banco de dados.

Nesta categoria foram observadas demandas de diferentes tipos de projetos, descritos nas próximas subseções.

Outro ponto a ser observado é a classificação, em alguns casos, dessas demandas como componente interno reusável (seção 4.15).

Recomenda-se enfaticamente a realização da análise de impacto das mudanças propostas para efeito de determinação do percentual adequado para aplicação sobre o total de pontos de função das funcionalidades impactadas. Por exemplo, em uma análise de impacto, pode ser identificado que não haverá mudanças no código-fonte ou em função transacional, sendo necessário somente testar o sistema, então deve-se utilizar um percentual contemplando apenas a fase de testes. No caso do teste apontar a necessidade de atualizar alguma função transacional, não deve ser contado o esforço do teste, mas sim o esforço abordado nesta seção, conforme as fórmulas apresentadas nas subseções seguintes.

#### **4.6.1 Atualização de Versão – Linguagem de Programação**

Nesta categoria encontram-se as demandas de atualização de versão de linguagem de programação de sistemas. As funções de dados não devem ser contadas. Estas demandas devem ser dimensionadas de acordo com a fórmula abaixo.

$$\text{PF\_ATUALIZAÇÃO\_VERSÃO\_LINGUAGEM} = \text{PF\_ALTERADO} \times 0,30$$

O PF\_ALTERADO deve considerar apenas as funcionalidades impactadas. As funcionalidades que possuem apenas demandas de testes, devem ser contadas usando o percentual da fase de testes (ver Tabela 7).

#### **4.6.2 Atualização de Versão – Browser**

Nesta categoria encontram-se as demandas de atualização de aplicações Web para executar em novas versões de um mesmo *browser* e para suportar a execução em mais de um *browser*. É importante destacar que este tipo de procedimento usualmente é realizado quando é necessário resolver algum problema de incompatibilidade. As funções de dados não devem ser contadas. Estas demandas devem ser dimensionadas de acordo com a fórmula abaixo.

$$\text{PF\_ATUALIZAÇÃO\_VERSÃO\_BROWSER} = \text{PF\_ALTERADO} \times 0,30$$

O PF\_ALTERADO deve considerar apenas as funcionalidades impactadas. As funcionalidades que possuem apenas demandas de testes, devem ser contadas usando o percentual da fase de testes (ver Tabela 7).

Essas atualizações podem implicar em manutenções em componentes específicos da plataforma utilizada. Nesse caso, a demanda deve ser contada como componente interno reusável, descrita na seção 4.15 deste roteiro.

### 4.6.3 Atualização de Versão – Banco de Dados

Nesta categoria encontram-se as demandas de atualização de versão do sistema gerenciador de banco de dados. As funções de dados não devem ser contadas. Estas demandas devem ser dimensionadas de acordo com a fórmula abaixo.

$$\text{PF\_ATUALIZAÇÃO\_VERSÃO\_BD} = \text{PF\_ALTERADO} \times 0,30$$

O PF\_ALTERADO deve considerar apenas as funcionalidades impactadas. As funcionalidades que possuem apenas demandas de testes, devem ser contadas usando o percentual da fase de testes (ver Tabela 7).

## 4.7 Manutenção em Interface

A manutenção em interface, denominada na literatura de manutenção cosmética, é associada às demandas de alterações de interface, por exemplo: fonte de letra, cores de telas, cores de letras, logotipos, mudança de botões na tela, mudança de posição de campos ou texto na tela. Também se enquadram nessa categoria as seguintes manutenções:

Mudanças de texto em mensagens de erro, validação, aviso, alerta, confirmação de cadastro ou conclusão de processamento;

Mudança em texto estático de e-mail enviado para o usuário em uma funcionalidade de cadastro. A demanda deve ser contada como manutenção em interface na funcionalidade de cadastro;

Alteração de título de relatório, de janela ou de caixas de texto;

Alteração de *labels* (rótulos) de uma tela ou de relatório ou de outra mídia.

Nestes casos, a aferição do tamanho em pontos de função das funções transacionais impactadas será realizada com a aplicação de um fator de redução de modo a considerar 20% da

contagem de uma função transacional de mais baixa complexidade (3 PF), ou seja 0,6 PF, independentemente da complexidade da funcionalidade alterada. Neste tipo de manutenção não são contadas funções de dados.

### **PF\_INTERFACE = 0,6 PF x QUANTIDADE DE FUNÇÕES TRANSACIONAIS IMPACTADAS**

Está contemplada a atualização da documentação das funcionalidades da aplicação impactadas pela manutenção nas demandas desta categoria. Assim, a documentação (documento de requisitos, documento de interface, protótipo, entre outros) das funcionalidades alteradas deve ser atualizada. Caso não exista documentação para as funcionalidades alteradas, não será contemplada a redocumentação das funcionalidades da aplicação impactadas pela manutenção nas demandas desta categoria.

**Observação 1 – Help:** As demandas de projetos de desenvolvimento de sistemas ou de manutenção de funcionalidades já contemplam o desenvolvimento ou atualização do *help* da funcionalidade em questão, sendo tratada como uma atividade de documentação no processo de software.

No caso de demandas específicas de desenvolvimento ou atualização de *help* estático de funcionalidades, estas podem ser enquadradas nesta seção e deverá ser usado o valor de multiplicação de 0,3 PF. Em caso de requisitos de usuário para o desenvolvimento de funcionalidades de manutenção de *help*, deve-se contar a função de dados de *help* e as funcionalidades de manutenção de *help* (por exemplo: incluir *help* de tela, consultar *help* de campo) de acordo com o CPM 4.3.1.

## **4.8 Adaptação em Funcionalidades sem Alteração de Requisitos Funcionais**

São consideradas nesta categoria as demandas de manutenção adaptativa associadas a solicitações que envolvem aspectos não funcionais, sem alteração em requisitos funcionais.

Seguem alguns exemplos de demandas de requisitos não funcionais:

Aumentar a quantidade de linhas por página em um relatório;

Colocar paginação em um relatório;

Limitar a quantidade de linhas por página em uma consulta existente;

Permitir exclusões múltiplas em uma funcionalidade que antes só possibilitava a exclusão de um item;

Adaptação de uma funcionalidade para possibilitar a chamada por um *web service* ou para outro tipo de integração com outros sistemas;

Replicação de funcionalidade: chamar uma consulta existente em outra tela da aplicação;

Alteração na aplicação para adaptação às alterações realizadas na interface com rotinas de integração com outros softwares, por exemplo, alteração em sub-rotinas chamadas por este software;

Modificar o servidor a ser acessado em uma funcionalidade de *download* de arquivo;

Adequar mensagem do sistema que em algumas telas apresenta “Usuário Não está Habilitado a ver esta Página”, para que passe a enviar uma mensagem mais adequada ao fato do usuário não possuir mais uma sessão ativa e ainda navegar no sistema. A demanda deve ser contada como manutenção adaptativa considerando as funcionalidades impactadas. Observe que trata-se de mudança em validação com regra de negócio não funcional.

Nestes casos, a aferição do tamanho em pontos de função da funcionalidade ou das funcionalidades que sofreram impacto deve considerar um fator de impacto (FI) sobre o PF\_ALTERADO, seguindo os conceitos do CPM 4.3.1, apresentados na seção 4.2.

$$\text{PF\_ADAPTATIVA} = \text{FI} \times \text{PF\_ALTERADO}$$

**FI (Fator de Impacto)** pode variar conforme condições abaixo:

**FI = 50%** para **funcionalidade de sistema desenvolvida ou mantida** por meio de um projeto de melhoria pela empresa contratada.

**FI = 75%** para **funcionalidade de sistema não desenvolvida ou mantida** por meio de um projeto de melhoria pela empresa contratada.

Caso exista a documentação das funcionalidades impactadas, estas deverão ser atualizadas sem ônus adicional à contratante. Caso contrário, se for demandada a redocumentação dessas funcionalidades, deve-se adicionar ao FI um fator de redocumentação de 15%, conforme descrito na seção 4.2.

## 4.9 Apuração Especial

São funcionalidades executadas apenas uma vez para: corrigir problemas de dados incorretos na base de dados das aplicações ou atualizar dados em bases de dados de aplicações, detalhados na subseção 4.9.1; gerar um relatório específico ou arquivo para o usuário por meio de recuperação de informações nas bases da aplicação, detalhados na subseção 4.9.2. A subseção 4.9.3 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.

Para estes casos, deve ser solicitada formalmente à empresa contratada o armazenamento do *script* para permitir posterior reexecução.

#### 4.9.1 Apuração Especial – Base de Dados

Este tipo de 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 de valor “sim” ou “não” no campo “indicador de matriz” referente ao CNPJ.

Normalmente, nesse tipo de atualização são afetados múltiplos registros. Nestes casos, considera-se a contagem de pontos de função das funcionalidades desenvolvidas. Geralmente, estas funcionalidades são classificadas como Entradas Externas. Nesse caso, como artefato de homologação da demanda, deve ser gerado um relatório para validação do usuário.

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 Internos.

Foram identificados três tipos de Apuração Especial – Base de Dados, cujas fórmulas de cálculo são apresentadas a seguir:

a) Atualização de Dados sem Consulta Prévia

$$\text{PF\_APURAÇÃO\_BD} = \text{PF\_INCLUÍDO}$$

b) Consulta Prévia sem Atualização

Em alguns casos de Apuração Especial – Base de Dados, o usuário solicita uma consulta prévia das informações. Deve-se ressaltar que essa consulta deve ser realizada antes da construção

da funcionalidade, não se trata de homologação. A consulta prévia não é definida pela empresa contratada, obrigatoriamente essa deve ser solicitada pelo órgão contratante para a avaliação da viabilidade de implementar a Apuração Especial – Base de Dados. 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 o tamanho da funcionalidade em questão, conforme a fórmula abaixo:

$$\text{PF\_CONSULTA\_PRÉVIA} = \text{PF\_INCLUÍDO}$$

#### c) Atualização de Dados com Consulta Prévia

Caso a Apuração Especial – Base de Dados seja solicitada após uma demanda de consulta prévia, deve-se aplicar um fator de 60% na fórmula de contagem da Apuração Especial – Base de Dados, seguindo a fórmula abaixo.

$$\text{PF\_APURAÇÃO\_BD\_PÓS\_CONSULTA\_PRÉVIA} = \text{PF\_INCLUÍDO} \times 0,60$$

### 4.9.2 Apuração Especial – Geração de Relatórios

Este tipo de 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 essas funções transacionais são Saídas Externas, devido à atualização do Arquivo Lógico Interno.

Deve-se destacar que essas 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.

$$\text{PF\_APURAÇÃO\_RELATÓRIOS} = \text{PF\_INCLUÍDO}$$

### 4.9.3 Apuração Especial – Reexecução

Em alguns casos, a 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 a aplicação de um fator redutor de 10% na contagem de pontos de função da apuração especial em questão, da seguinte maneira:

$$\text{PF\_REEEXECUÇÃO\_APURAÇÃO} = \text{PF\_NÃO\_AJUSTADO} \times 0,10$$

Para ser considerada reexecução de script, não deve haver alteração de nenhum parâmetro no script original.

#### 4.10 Atualização de Dados

Em alguns casos, as demandas de correção de problemas em base de dados estão associadas a atualizações manuais (de forma interativa), diretamente no banco de dados em um único registro, e que não envolvem cálculos ou procedimentos complexos. São exemplos desse tipo de demanda, a atualização do valor de um campo de uma tabela cadastrado erroneamente ou a exclusão de um registro de uma tabela.

Nestes casos, a aferição do tamanho em Pontos de Função deve considerar 10% do PF de uma Entrada Externa e os Tipos de Dados da Entrada Externa são todos os TD considerados na funcionalidade – campos atualizados e campos utilizados para a seleção do registro.

$$\text{PF\_ATUALIZAÇÃO\_BD} = \text{PF\_INCLUÍDO} \times 0,10$$

#### 4.11 Atualização de Dados de Domínio

Em alguns casos, as demandas de atualização em base de dados estão associadas a atualizações em um único registro em uma Tabela de Domínio do tipo Code Data. Por exemplo, suponha a atualização de uma Tabela de Situação de Projetos (Andamento, Concluído, Cancelado) para inclusão da Situação “Suspenso”. Então, esta demanda será classificada como Atualização de Dados de Domínio (PF\_Domínio) e contada como 10% da contagem de Pontos de Função de uma Entrada Externa de complexidade baixa, ou seja, 0,3 PF. No exemplo acima, EE: Atualizar Situação de Projetos (AR:1 Situação do Projeto; TD: 1 Situação do Projeto) – Baixa – 3 PF x 10% = 0,3 PF. Segue a fórmula de cálculo:

$$\text{PF\_DOMÍNIO} = \text{PF\_INCLUÍDO} \times 0,10$$

Estas demandas correspondem as Apurações Especiais de Atualização de Dados.

#### **4.11 Manutenção de menus, telas e listas de seleção estáticas**

As alterações de menus, telas e listas de seleção estáticas, que não fazem parte de um projeto de melhoria ou de desenvolvimento, serão contadas com a aplicação de um fator de redução de modo a considerar 10% da contagem de uma função transacional de mais baixa complexidade (3 PF), ou seja 0,3 PF, para cada menu, tela ou lista de seleção estática, independentemente da complexidade da funcionalidade alterada. Neste tipo de manutenção não são contadas funções de dados.

As alterações de menus, telas e listas de seleção estáticas que fazem parte de um projeto de melhoria ou de desenvolvimento não terão acréscimos na contagem, uma vez que o projeto de melhoria ou de desenvolvimento já abarca todas as alterações nas funcionalidades do seu escopo, incluindo as chamadas das funcionalidades.

#### **4.12 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 foi definido o fator de impacto de 25% dos pontos de função da aplicação em questão, considerando a fase de requisitos e a geração de artefatos associados a requisitos, conforme a fórmula abaixo.

$$\text{PF\_DOCUMENTAÇÃO} = \text{PF\_NÃO\_AJUSTADO} \times 0,25$$

#### **4.13 Verificação de Erros**

As verificações de erro ou análise e solução de problemas são 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 algum erro de sistema, a demanda será atendida como manutenção corretiva (Seção 4.4).

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 que o cliente reportou erro. Caso não exista documentação de testes disponível dessas funcionalidades

verificadas, será considerado 20% do tamanho funcional dessas funcionalidades com solicitação de análise pelo órgão contratante, segundo a fórmula abaixo:

$$\text{PF\_VERIFICAÇÃO} = \text{PF\_Funcionalidade\_Reportada\_Com\_Erro} \times 0,20$$

Caso exista documentação de testes das funcionalidades verificadas, então será considerado 15% (mesmo percentual da fase de Testes, conforme Tabela 7) do tamanho funcional das funcionalidades analisadas, segundo a fórmula abaixo:

$$\text{PF\_VERIFICAÇÃO} = \text{PF\_Funcionalidade\_Reportada\_Com\_Erro} \times 0,15$$

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 de rede ou banco de dados devem ser tratados como serviços de suporte e não serviços de desenvolvimento e manutenção de sistemas. Esses serviços de suporte não fazem parte do escopo desse roteiro de métricas, não se aplicando verificação de erros nestes casos.

#### 4.14 Pontos de Função de Teste

Muitas vezes, em projetos de manutenção, o conjunto de 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 apenas 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 será o somatório dos tamanhos em pontos de função das funções transacionais envolvidas no teste:

$$\text{PFT} = \text{Somatório dos Tamanhos das Funções Transacionais Testadas}$$

A conversão do PFT em ponto de função deve ser feita de acordo com a fórmula abaixo:

$$\text{PF\_TESTES} = \text{PFT} \times 0,15$$

É importante ressaltar que no caso de uma função ser testada várias vezes, com cenários diferentes, a função só pode ser contada uma vez. Outra observação é que as funções testadas, consideradas no PFT, devem ser documentadas pela contratada considerando-se a documentação de testes definida no processo de desenvolvimento da contratante. Observe que estas funções farão parte do escopo de projetos de manutenção e desenvolvimento.

## 4.15 Componente Interno Reusável

Em alguns casos são demandadas manutenções em componentes específicos de uma aplicação e estes são reusados por várias funcionalidades da aplicação.

Este roteiro define que o componente, o qual deverá ser desenvolvido ou mantido, seja considerado como um processo elementar independente e contado como uma funcionalidade. Além disso, as funcionalidades da aplicação que necessitem de teste devem ser requisitadas pela contratante e dimensionadas por meio da métrica Pontos de Função de Teste proposta na seção 4.14.

### **PF\_COMPONENTE = PF\_NÃO\_AJUSTADO**

Seguem alguns exemplos de manutenção de componentes:

Alteração de valores de elementos internos de configuração que afetem o comportamento ou a apresentação do sistema de forma geral, tais como páginas de estilos (arquivos CSS de sistemas Web), arquivos com mensagens de erro, arquivos de configuração de sistema, arquivos de internacionalização.

Mudança em tópico de um menu de um sistema em PHP que aparece em todas as telas da aplicação. A contagem pode ser realizada considerando o componente “Apresentar Menu”;

As demandas de desenvolvimento de componentes específicos, tais como: middleware, webservices, instaladores de software podem ser tratadas como uma fronteira de aplicação a parte, seguindo as orientações do IFPUG, visto que consistem uma demanda específica.

## 4.16. Projeto de Teste de Interoperabilidade

Nesta seção são tratadas as demandas específicas para testes baseados em processos de negócio. Estas demandas consistem nos testes de interoperabilidade entre diferentes sistemas que trabalham em conjunto para garantir que pessoas, organizações e sistemas computacionais interajam entre si para trocar informações de maneira eficaz (teste funcional) e eficiente (teste não funcional). A contagem de PF destas demandas consiste no seguinte:

- Calcular o tamanho em Pontos de Função de todas as funções de dados ou transacionais envolvida no teste.

$$\text{PF\_TESTES\_INTEROPERABILIDADE} = \text{PF\_Funcionalidades\_Apenas\_Testadas} \times 0,25$$

#### 4.17. Pontos de Função de Testes Não Funcionais

Em alguns projetos podem ocorrer demandas por testes não funcionais. Os testes não funcionais tem o propósito de testar os atributos de qualidade de um sistema ou parte dele, por exemplo: confiabilidade, eficiência, usabilidade, manutenibilidade e portabilidade (ISO/IEC 9126). Os principais tipos de testes não funcionais realizados nos sistemas, dentre outros, são os seguintes: Testes de performance – para verificar o tempo de resposta da aplicação; Testes de stress – para verificar a quantidade de usuários simultâneos suportados pela aplicação; Testes de segurança – para verificar o atendimento dos requisitos não funcionais de segurança; Testes de portabilidade – para verificar o comportamento da aplicação em vários ambientes, como por exemplo, testar um PGD em sistemas operacionais diferentes, testar uma aplicação web em diferentes browsers e testar a mesma aplicação em diferentes tipos de dispositivo móvel. Os atributos não funcionais de qualidade de um produto de software podem também ser aferidos em níveis de interoperabilidade entre sistemas.

As atividades de execução de testes não funcionais devem ser realizadas por especialistas em testes e dimensionadas de acordo com a seguinte fórmula:

$$\text{PF\_TESTES\_NÃO\_FUNCIONAIS} = \text{PF\_Funcionalidades\_Testadas} \times 0,25$$

Devem ser consideradas apenas as funções transacionais impactadas pela demanda do teste não funcional. Deverão ser gerados documentos que evidenciem a realização dos testes não funcionais.

#### 4.18. Pontos de Função de Massa de Dados para Testes em Homologação

Em alguns projetos podem ocorrer demandas do cliente de geração de massa de dados para testes em homologação. Estas atividades consistem em carga de dados no Banco de Dados do Ambiente de Homologação ou de Treinamento. A realização destas atividades é similar à Apuração Especial. Desta forma, pode ser contada como Apuração Especial. Segue a fórmula de cálculo:

$$\text{PF\_MASSA\_TESTES} = \text{PF\_INCLUÍDO}$$

Este Roteiro não contempla retrabalho para carga de Massa de Testes para Homologação. Após a realização da carga de dados no Banco de Dados do Ambiente de Homologação ou de Treinamento, caso o cliente solicite mudanças nos requisitos do projeto, tornando necessário implementar e executar novas cargas.

#### **4.19. Roteiro para Homologação**

A elaboração de Roteiro para Homologação consiste na definição de um guia demonstrando os passos para utilizar as funcionalidades disponibilizadas na homologação do sistema. O produto gerado por esta atividade é um Roteiro para Homologação. O dimensionamento desta atividade será realizado considerando 2% da contagem de Pontos de Função das funções transacionais impactadas, de acordo com a seguinte fórmula:

$$\text{PF\_ROTEIRO\_HOMOLOGAÇÃO} = \text{PF\_Funcionalidades\_Impactadas} \times 0,02$$

O produto Roteiro de Homologação pode ser utilizado como artefato para evidenciar a realização da atividade.

#### **4.20. Gerenciamento de Riscos de Segurança**

O desenvolvimento ou manutenção, geralmente em inclusão de novos módulos, de sistemas com complexidade alta pode tornar necessária a avaliação dos riscos de segurança antes da entrada do mesmo em produção. O produto gerado por esta atividade para o cliente é um Relatório de Riscos. O dimensionamento desta atividade será realizado considerando 2% da contagem de Pontos de Função do Projeto, sem considerar Retrabalho, Massa de Dados para Testes em Homologação, testes funcionais e testes não funcionais, de acordo com a seguinte fórmula:

$$\text{PF\_ROTEIRO\_HOMOLOGAÇÃO} = \text{PF\_Demanda} \times 0,02$$

#### **4.21. Desenvolvimento e Manutenção de Help**

Em alguns projetos há demandas específicas de desenvolvimento e/ou manutenção de help de funções (help de campo, help de tela ou relatório e help do sistema). O conteúdo do help deve ser elaborado pelo cliente. O dimensionamento destas demandas deve ser realizado de acordo com a seguinte fórmula:

$$\text{PF\_HELP} = \text{PF\_Funcionalidades\_Impactadas} \times 0,05$$

Caso a demanda seja **apenas** de Help de sistema, consistindo na disponibilização de texto enviado pelo usuário em um link de Help, então a demanda será contada como Manutenção de Componentes. Sendo considerando uma Consulta Externa de complexidade baixa.

Em casos de projetos de manutenção, considerando-se uma alteração apenas no Help de uma funcionalidade, então este help será contado como publicação de página estática. Observe que se a funcionalidade e o help forem alterados a demanda será contada como projeto de melhoria e Ponto de Função de Help. Se apenas a funcionalidade for alterada, sem impacto no Help, não será contado Ponto de Função de Help. Se o Help for um arquivo, a atualização de dados do Arquivo de Help será tratada como um Projeto de Atualização de Dados de Domínio.

## 4.22. Manual do Usuário

As demandas de construção do manual do usuário quando ocorrem no momento da construção do sistema devem ser tratadas como atividade de desenvolvimento de Help. No entanto, se estas ocorrem para sistemas desenvolvidos pela empresa contratada e implantados deve ser considerado 10% da contagem de Pontos de Função das funcionalidades documentadas. Em caso de sistemas não desenvolvidos pela empresa contratada deve ser considerado 25% da contagem de Pontos de Função das funcionalidades documentadas.

$$\text{PF\_MANUAL\_USUÁRIO\_Empresa} = \text{PF\_Funcionalidades\_Impactadas} \times 0,10$$

$$\text{PF\_MANUAL\_USUÁRIO\_Não\_Empresa} = \text{PF\_Funcionalidades\_Impactadas} \times 0,25$$

## 5. Orientações Complementares para Contagem

Este capítulo tem como propósito apresentar as diretrizes gerais de contagem de pontos de função e procedimentos a serem adotados em casos especiais.

## 5.1 Divisão de um mesmo Processo Elementar em Transações Distintas

A divisão de um mesmo processo elementar em transações distintas fere o conceito disposto no item 5.5.2 do Manual de Práticas de Contagem de Pontos de Função (CPM 4.3.1):

*“Para identificar cada processo elementar devem ser executadas as seguintes atividades:*

*a) Compor e/ou decompor os Requisitos Funcionais do Usuário nas menores unidades de atividade que satisfaçam todos os itens abaixo:*

- 1) é significativa para o usuário;*
- 2) constitui uma transação completa;*
- 3) é autocontida;*
- 4) deixa o negócio da aplicação contada em um estado consistente.”*

O item 3.7 do CPM explicita o conceito de autocontido, reforçando a propriedade de constituição de transação completa. O entendimento do exemplo a seguir facilitará o entendimento de um cenário envolvendo a divisão de um processo elementar em transações distintas.

*“Nenhum passo anterior ou subsequente é necessário para iniciar ou concluir o(s) Requisito(s) Funcional(is) do Usuário.*

*EXEMPLO: O Requisito Funcional do Usuário estabelece que um empregado deve ser incluído e atualizado.*

*Poderiam existir várias partes que comporiam o conjunto completo de informações do empregado. Isto pode ser representado por telas físicas, janelas ou abas distintas, tais como:*

- Identificação do empregado,*
- Localização do empregado,*
- Informações de dependentes,*
- Informações de salário e*
- Instrução.*

*Para incluir um empregado, uma ou mais abas devem ser preenchidas, dependendo das regras de negócio. O processo de inclusão não estará autocontido até que todas as informações obrigatórias tenham sido digitadas e recebidas pelo sistema.*

*Para atualizar um empregado, uma ou mais abas podem ser atualizadas a qualquer momento, mas todas elas constituem passos do processo que satisfaz o Requisito Funcional do Usuário, de atualização do empregado.*

*Incluir, alterar ou excluir informações de cada aba individual não constituem processos elementares distintos, mas sim passos de processo envolvidos na atualização de*

um empregado. Embora seja possível entrar com informações adicionais no registro de empregado, o conjunto total de informações é considerado parte do único processo elementar: atualizar empregado.

Incluir Empregado e Atualizar Empregado seriam, cada um, um processo autocontido.”

## 5.2 Contagem de Pontos de Função com Múltiplas Mídias

A contagem de PF de funcionalidades entregues em mais de uma mídia, na 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*.

É importante enfatizar que o IFPUG reconhece ambas abordagens, *single instance* e *multiple instance*, para a aplicação das regras definidas no CPM.

A seguir são descritos os termos comuns definidos pelo IFPUG [IFPUG, 2010]:

**Canal:** também refere-se a mídia. Múltiplos canais é sinônimo de múltiplas mídias.

**Mídia:** descreve a maneira como 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, apenas 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 funcionalidade, 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 do objetivo da contagem, permitindo uma função de negócio ser reconhecida no contexto das mídias que são requisitadas para que a funcionalidade seja 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 com mídias, mas apresentam a aplicação dos casos mais comuns.

### **5.2.1 Cenário 1: Mesmos dados apresentados em tela e exportados**

Neste cenário, uma aplicação apresenta uma consulta em tela para que possa servir como recurso para incluir, editar, consultar, excluir, etc registros selecionados na listagem da consulta e uma funcionalidade para Exportar o resultado da pesquisa.

Nesse caso, deve ser considerado a funcionalidade de pesquisar como um processo elementar, classificado como Saída Externa (em função do padrão do órgão em apresentar o totalizador de registros) e o processo elementar Exportar como uma outra Saída Externa com um DER adicional (Tipo de arquivo).

O processo elementar Exportar, em um projeto de desenvolvimento, independente das opções de exportação (pdf, xls, csv, txt, html, etc) disponíveis para a seleção do usuário, deve ser considerado um processo elementar único, quando não houver diferentes DER, ALR ou lógicas de processamento solicitadas e validadas pelo usuário.

### **5.2.2 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 às gravadas no arquivo.

Nesses casos, utiliza-se a abordagem *single instance*, considerando que os dados impressos e os dados apresentados no arquivo de saída sejam idênticos. Assim, apenas uma funcionalidade será incluída na contagem de pontos de função.

Este caso ocorre com ou sem o apoio de ferramenta de desenvolvimento para a geração dessas múltiplas saídas.

Porém, 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 tiverem que ser customizadas para o cliente, então será utilizada a abordagem *multiple instance*. Caso seja requisitado a geração de mais de um tipo de arquivo, cada tipo de arquivo gerado será considerada uma funcionalidade distinta.

Caso já exista uma aplicação com seu conjunto de mídias (tela/impresso/arquivos/etc.) para uma funcionalidade e a Contratante solicita a criação de uma ou mais mídia(s) inédita(s) para aquela funcionalidade, a Contratada deverá ser remunerada ao preço de uma funcionalidade, mas não

haverá acréscimo no tamanho funcional da aplicação, por se tratar de mídia(s) adicional(is) da mesma funcionalidade. Porém, deverá ser registrado na *baseline* da aplicação o(s) registro(s) da(s) nova(s) mídia(s), com contagem de 0 (zero) pontos de função, pois é(são) processo(s) elementar(es) mas em duplicidade para fins de tamanho funcional.

### **5.2.3 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, da mesma forma que o processamento da entrada on-line também executa validações das informações. Neste caso, utiliza-se 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.

### **5.2.4 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. Neste caso, utiliza-se a abordagem *multiple instance*, que conta duas funcionalidades: consulta de dados na Web e consulta de dados via celular.

Considera-se que a funcionalidade é desenvolvida duas vezes, uma para cada canal de saída. 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 seja claro o suficiente para dizer que o desenvolvimento é o mesmo, poderá ser utilizada a abordagem *single instance*.

Neste Cenário também se enquadram as funcionalidades de Sistemas Mobile desenvolvidas em mais de uma plataforma, por exemplo, iOS e Android. Cabe ressaltar que existem Frameworks de desenvolvimento de sistemas Mobile que geram o código para diferentes plataforma, IOS, Android, Windows, BlackBerry, etc. Nestes casos, utiliza-se a abordagem *single instance*. Caso seja solicitado pelo cliente IBAMA a construção ou manutenção de funcionalidades Mobile em plataformas distintas e a ferramenta de desenvolvimento solicitada pelo cliente não permita geração de código para as plataformas solicitadas. Ou seja, caso a equipe de desenvolvimento precise desenvolver código separadamente para duas ou mais plataformas, então será utilizada a abordagem *multiple instance* para contagem de PF das funcionalidades em questão.

### **5.2.5 Cenário 5: Relatório em múltiplos formatos**

Em demandas que possuem a solicitação do usuário para a geração de relatórios que contenham mesmas informações (DER), acessem mesmos arquivos (ALR) e possuem mesma lógica de processamento (não havendo supressões de registros por filtros, por exemplo), utiliza-se a abordagem *single instance*, ou seja, conta-se apenas a criação de 1 (uma) funcionalidade.

Conforme o Manual de Práticas de Contagem de Pontos de Função 4.3.1, quando dois processos elementares (PEs) são comparados e se determina que eles contêm diferentes DERs, RLRs ou Processamento Lógico, eles são identificados como processos elementares separados se forem especificados como requisitos funcionais distintos pelo usuário. Nesse caso, utiliza-se a abordagem *multiple instance*.

Assim, uma consulta que é exibida em tela constitui uma funcionalidade. Havendo opções de exportação destes mesmos dados da consulta em diversos formatos (pdf, doc, xls, .txt, .csv, html, xml, etc) ou mesmo a impressão direta em impressora, não se contam funcionalidades adicionais (a exibição das opções de exportação em botões separados ou em caixa de diálogo não invalida este conceito, por ser decisão de projeto/implementação).

Cabe ressaltar que quando houver diferença entre o relatório apresentado em tela e o relatório de exportação (DER, ALR ou lógica de processamento), deve ser especificado no documento de caso de uso o requisito funcional que ateste a diferença entre os relatórios.

## **5.2.6 Cenário 6: Funcionalidades Fornecidas via Aplicação e Webservice**

Em algumas demandas são requisitadas funcionalidades fornecidas pela aplicação via interface com o usuário final e também via Web Service. Estas funcionalidades devem ser contadas duas vezes, considerando a abordagem *multiple instance*, visto que há esforço para o desenvolvimento das duas funcionalidades, assim como são entregues duas funcionalidades para o usuário. Em geral, a lógica de processamento destas funcionalidades costuma ser distinta, considerando as validações.

## **5.3 Log, Trilha de Auditoria e Histórico**

O objetivo dessa sessão é descrever orientações a respeito de contagem de log, trilha de auditoria e histórico.

### **5.3.1 Log**

Conceituamos o termo “Log” como o registro de procedimentos ou ações realizadas pela aplicação, em determinado período de tempo, com o objetivo de apoiar a auditoria do ambiente tecnológico e a identificação das causas raízes de falhas em sistemas. Diante desse conceito, definimos que o Log não deve ser mensurado com Pontos de Função, já que ele não armazena informações negociais reconhecidas pelo usuário da aplicação.

### **5.3.2 Trilha de Auditoria**

Conceituamos “Trilha de Auditoria” como a funcionalidade que tem o objetivo de armazenar informações referentes às ações realizadas pelos usuários da aplicação no passado, de modo que seja possível apurar quais foram as ações executadas quando da utilização do sistema.

Para isso, devem existir no mínimo as informações para identificar quem realizou a ação, quando e o que foi realizado, além de outras informações que o usuário da aplicação defina como necessárias.

A trilha de auditoria deve ser solicitada pelo usuário da aplicação e, para a contagem de Pontos de Função, esta necessidade de negócio deve estar documentada na especificação de requisitos funcionais do sistema.

Caso a trilha de auditoria faça parte da política corporativa de segurança da informação adotada pelo contratante para todos os sistemas do órgão, ela deve ser considerada como um requisito não funcional e, portanto, não será mensurável em ponto de função.

Diante do exposto, a principal diferença entre o Log e a Trilha de Auditoria é:

Log: apoia a coleta de informações no âmbito tecnológico, ou seja, em problemas decorrentes da arquitetura tecnológica que precisam ser investigados, por meio da análise do conjunto de procedimentos executadas pela aplicação, como exemplo a baixa performance no sistema, travamentos e outros comportamentos inesperados.

Trilha de Auditoria: apoia a auditoria para os dados de negócio, armazenando informações das ações realizadas pelo usuário na aplicação.

### **5.3.3 Histórico**

Conceituamos “Histórico” como um registro de estados com informações anteriores de um registro em determinado momento. O usuário poderá consultar a evolução dessas informações em uma linha do tempo e sua existência é justificada pelo negócio. Assim, para fazer parte do tamanho funcional, deve ser solicitado pelo gestor ou requisitante e deverá existir funcionalidade de consulta a tais dados.

A função de consulta aos dados de um histórico deverá ser contada de acordo com as regras de contagem das funções transacionais do CPM.

Não devem ser consideradas na contagem funções de transação separadas para incluir, alterar e excluir as informações históricas, pois o armazenamento dessas informações é parte integrante das mesmas funcionalidades que processam os dados de negócio. Apenas quando o histórico for mantido de forma independente do registro principal, por exemplo no caso do ALI principal ter sido excluído, o histórico se torna um ALI independente e não um registro lógico do ALI relacionado.

Para contagem de Pontos de Função de dados de Histórico, a manutenção de dados de Histórico deve ser requisitada pelo usuário para atendimento de necessidades de negócio. Em caso de requisitos técnicos para registros de eventos de histórico em tabelas próprias, não deve ser computado Pontos de Função. Considerando que Pontos de Função é uma métrica de tamanho funcional.

## **5.4 Documentação de Contagem de Pontos de Função**

As funcionalidades identificadas devem ser descritas, incluindo a rastreabilidade para o requisito de origem. Também podem ser descritas: motivo da contagem, justificativa de contagem e observações no campo Parecer do Estimados caso necessário. Para os Arquivos de Interface Externa deve ser identificada a fronteira da aplicação de origem. Por exemplo, AIE: Municípios (TOM). Os Arquivos Referenciados e Registros Lógicos devem ser descritos na memória de cálculo. Os Tipos de Dados devem ser descritos apenas para justificar a complexidade. Por exemplo, para um Relatório com 4 Arquivos Referenciados, só precisarão ser descritos 6 tipos de dados para justificar a complexidade alta. Caso haja disponibilidade do estimador, podem ser descritos todos os Tipo de Dados da funcionalidade.

## **5.5 Contagem de Pontos de Função de Demandas de Manutenção**

Em algumas demandas podem ocorrer dois ou mais tipos de manutenção em uma mesma funcionalidade. Por exemplo, acrescentar um campo em uma consulta e alterar o texto estático do cabeçalho da consulta e melhorar o tempo de resposta da consulta em questão. Observe que foram solicitadas três tipos de manutenção em uma mesma funcionalidade: Melhoria – acrescentar campo; Manutenção em Interface – atualização texto estático da tela; Manutenção adaptativa em requisitos não funcionais – melhoria de performance. Como estas manutenções foram pedidas em uma mesma demanda, a funcionalidade poderá ser contada APENAS UMA

VEZ em uma demanda. Desta forma, o tipo de demanda considerado será aquele que possui a maior contagem de Pontos de Função. Neste caso será classificado como Melhoria. Caso o IBAMA realize os pedidos de manutenção em uma mesma funcionalidade em momentos distintos, ou sejam demandas distintas, então serão realizadas contagens de PF separadamente. Outro ponto, se o IBAMA solicitar várias mudanças de requisitos em uma mesma funcionalidade em momentos distintos, serão realizadas contagem de PF retrabalho separadamente, considerando a fase em questão.

## **5.6 Requisitos Funcionais x Requisitos Não Funcionais**

A métrica Pontos de Função é uma métrica de tamanho funcional de projetos de software. Desta forma, todos os requisitos funcionais descrito nos documentos de requisitos do sistema devem ser considerados em uma contagem de Pontos de Função. Cabe ressaltar, que em aderência a literatura de Engenharia de Requisitos, os Casos de Uso documentam os requisitos funcionais do sistema. Alguns outros tipos de documentos de requisitos podem ser utilizados para documentar os requisitos funcionais dos sistemas, tais como: regras de negócio, cenários de uso, itens de backlog, histórias de usuários e Formalização Simples de Requisitos (FSR). O documento de requisitos funcionais aprovado pelo cliente deve ser usado como base para contagem de Pontos de Função da demanda.

## **5.7 Unicidade do Processo Elementar**

A contagem de Pontos de Função de um processo elementar depende de sua unicidade, ou seja, um processo elementar não deve ser contado duas vezes na mesma fronteira de aplicação, mesmo que existam variações na implementação deste processo elementar, por exemplo uma lista com nome e CPF de aluno implementada em uma tabela e em outra tela implementada em uma combobox deve ser contada apenas uma vez. Um processo elementar é único quando ele possui Arquivos Referenciados distintos, ou Tipo de Dados distintos ou Lógica de Processamento distinta. Cabe ressaltar que a lógica de processamento de ordenação não fornece unicidade para o processo elementar. Uma vez que o processo elementar seja identificado e considerado único, este deve ter contagem de Pontos de Função.

## **5.8 Entidades Lógicas x Entidades Físicas**

Uma entidade física pode corresponder a várias entidades lógicas, especialmente em sistemas que usam o Banco de Dados hierárquicos como ADABAS. A contagem de Arquivos Lógicos

leva em consideração as entidades lógicas. Desta forma, uma tabela física pode ser contada como vários Arquivos Lógicos Internos. Por outro lado, um Arquivo Físico pode ser contado como parte de um Arquivo Lógico Interno – Registro Lógico (implementação de entidades fracas) ou até mesmo não ser contado (implementação de Code Data).

## 5.9 Perfis de Acesso e Funcionalidades

Alguns sistemas possuem vários grupos de usuários com perfis de acesso distintos. Para estes perfis de acesso ou grupos de usuários são disponibilizadas funcionalidades distintas ou não. Na contagem de Pontos de Função deve-se contar todas as funcionalidades que a aplicação disponibiliza para os grupos de usuários. No entanto, deve-se atentar para não contar duas vezes funcionalidades iguais com base nos critérios de unicidade do manual de práticas de contagem (CPM), a saber: A funcionalidade é única se ela tiver arquivos referenciados distintos, ou tipos de dados distintos ou lógica de processamento distinta. Por exemplo: uma função alterar senha disponibilizada para vários perfis deve ser contada apenas uma vez, considerando que esta seja idêntica para todos os perfis. Já a funcionalidade alterar aluno de um sistema de treinamento não é igual para o perfil aluno e o perfil professor. O perfil aluno pode alterar seus dados cadastrais e o perfil professor pode alterar a informação de notas das provas. Desta forma, alterar aluno – perfil aluno e alterar aluno – perfil professor, devem ser contados como dois processos elementares distintos porque possuem Tipo de Dados distintos.

Observação: Existem telas de consulta onde o resultado apresentado é diferente de acordo com o perfil. Digamos por exemplo que exista um sistema de gerenciar Ordens de Serviço. Ao acessar a tela com o meu usuário (comum) o sistema lista todas as Ordens de Serviço atribuídas a mim. Ao acessar a mesma tela com o perfil de gerente, o sistema listas todas as Ordens de Serviço, independentemente para quem esteja atribuída a Ordem de Serviço. Trata-se da mesma funcionalidade, que apresenta os mesmos campos, só que com uma visualização distinta por conta de uma regra de negócio, onde o que muda não é a funcionalidade em si, mas o conteúdo apresentado. Neste caso, a diferença entre as funcionalidades para os grupos de usuários é o de um valor do filtro, ou seja, não atende os critérios de unicidade do CPM para ser contada em duplicidade.

## 5.10 Contagem de PF de Telas de Sistemas com Abas

A contagem de Pontos de Função das funcionalidades dos sistemas leva em consideração os requisitos funcionais e especialmente a identificação dos processos elementares. Cabe ressaltar que um processo elementar pode ser implementado em várias telas da aplicação e ainda uma tela pode implementar vários processos elementares da aplicação. Um processo elementar é a menor unidade de atividade significativa para o usuário. O processo elementar deve ser

autocontido e deixar a aplicação em um estado consistente. Ou seja, o processo elementar é uma transação independente. Desta forma, em uma implementação de uma tela com abas para cadastro de informações deve-se observar o seguinte:

Caso os requisitos do negócio identifiquem a necessidade de salvar as informações de todas as abas no mesmo momento (por exemplo com um único botão de salvar), então todas as informações das abas fazem parte do mesmo processo elementar. Em alguns casos, onde o usuário realiza o preenchimento por abas e salva parcialmente, os dados ficam armazenados em tabelas temporárias e o usuário de negócio nem tem acesso ao conteúdo que foi salvo.

Caso os requisitos do negócio identifiquem a necessidade de salvar as informações de cada aba de forma independente das demais, seja automaticamente quando o usuário mudar de aba ou através de um acionamento de botão, serão identificados processos elementares distintos para cada aba (incluir, alterar, excluir, consultar, etc. ). Observe que se a necessidade de salvar os dados de cada aba de forma isolada seja para atender um requisito técnico, por exemplo evitar a perda de informações em caso de time out de sessão de usuário, dentre outros. Então não se deve contar separadamente. Porque não se trata de um requisito de negócio/funcional.

Por exemplo, existem cadastros a serem realizados, onde a quantidade de dados para preenchimento é bastante extensa e para facilitar o preenchimento e deixar claro para o usuário que trata-se de conteúdo distinto, os dados estão disponibilizados em abas distintas por conta de um requisito não funcional. Por exemplo, Cadastrar Currículo, ao invés de ter uma tela extensa com vários campos, divide-se por conteúdo e em abas: Dados Básico, Formação Acadêmica, Experiência Profissional, Atividades Sociais, etc. Sendo que em cada aba desta possa existir o botão de salvar, mas isto feito por uma necessidade técnica, por exemplo, para o usuário não perder os dados no caso de uma queda de energia ou internet, mas para o usuário de negócio somente importa receber o currículo completo, com todas as abas preenchidas. Neste exemplo, a funcionalidade Cadastrar Currículo é apenas um processo elementar.

Portanto, é importante analisar a necessidade negocial e técnica para identificação dos processos elementares. Se ao salvar parcialmente cada aba, o usuário de negócio pode ter acesso àquele dado, pode trabalhar no mesmo e é irrelevante para o negócio as demais abas não terem sido preenchidas, então deve-se mensurar como um Processo Elementar a parte. No entanto, se ficar evidenciado que o salvar parcial apenas atende um requisito técnico e que para o usuário de negócio não faz sentido ter aqueles dados até tudo estar preenchido, cada aba não pode ser considerada um Processo Elementar.

## 5.11 Dados de Código

Os Dados de Código ou Code Data são implementações de requisitos técnicos, requisitos não funcionais. para melhorar a manutenibilidade da aplicação, não sendo contemplados em uma Contagem de Pontos de Função, em aderência ao CPM 4.3.1. São tabelas do tipo Código – Descrição ou qualquer outra entidade com dados estáticos, sem funcionalidades de manutenção requisitadas pelo cliente, considerando os requisitos de negócio ou funcionais da aplicação. Geralmente, são mantidos por meio de Apuração Especial.

É importante observar que nem sempre as Tabelas do tipo Code Data possuem atributos de Código e Descrição. O CPM menciona a existência de três tipos de Dado de Código:

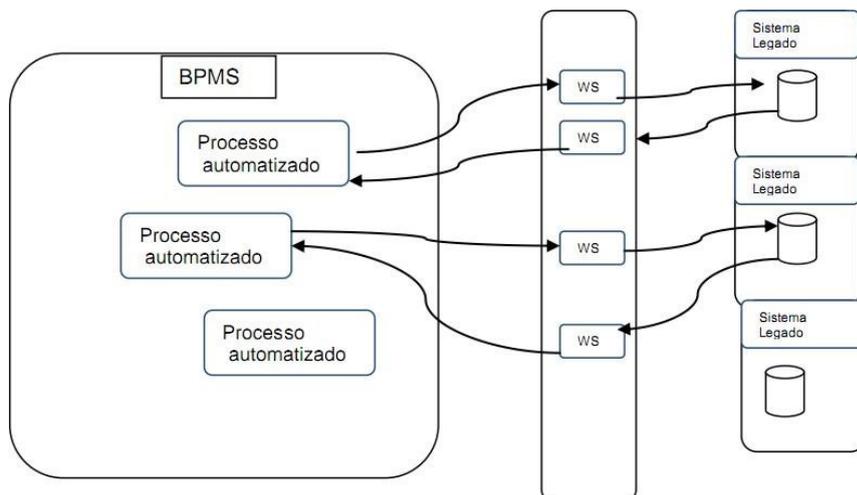
- Substituição
- Estáticos ou Constantes
- Valores Válidos

Os Dados de Negócio podem ser tabelas contendo atributos de Código e Descrição com dados dinâmicos, com funcionalidades de manutenção requisitadas pelo cliente, ou seja, contidas na especificação de requisitos de negócio ou funcionais do usuário. Desta forma, as funcionalidades de manutenção de dados devem estar contidas em um documento de requisitos funcionais aprovado pelo IBAMA. Estas Tabelas são contadas como Arquivo Lógico Interno ou Registro Lógico, observando as regras de contagem de Funções de dados do CPM. As funcionalidades de manutenção e de consulta a estes dados, descritas no Documento de Requisitos aprovado, também serão contadas de acordo com as regras de contagem de Funções transacionais do CPM.

Portanto, na identificação de Dados de Negócio e de Dados de Código deve-se ser analisado a essência do dado e verificar qual a relevância deste para o negócio. E ainda se este atende as características do CPM para ser considerado como um dos três tipos de Dado de Código.

## 5.12 Aferição de Pontos de Função em ambiente BPMS

Um BPMS é um Sistema no qual se pode desenvolver aplicativos integrados para gerenciamento de processos de negócios contemplando recursos como: documentação de processos, execução automatizada de processos, possibilidade de criação de indicadores gerenciais de processos em painéis de controle, *upload* e trâmite de documentos eletrônicos, com possibilidade de certificação digital e integração com sistemas legados através da filosofia SOA (*Service Oriented Architecture*).



O ambiente BPMS mantém um conjunto de processos automatizado que se comunicam com os sistemas legados por meio de uma Interface de integração composta por tecnologias do tipo *webservice*, *view*, etc.

Os processos automatizados são executados dentro do ambiente BPMS. No Ibama este ambiente é proporcionado pelas ferramentas integradas Oracle.

### 5.12.1 Fronteira de Aplicação

Para a contagem de pontos de função de automação de processos, deve-se considerar como fronteira da aplicação apenas a interface conceitual do processo de negócio que será automatizado.

Não se deve considerar na contagem as funcionalidades que já integram e fazem parte do ambiente BPMS, a exemplo: funcionalidade de login, impressão nativa, Minhas Atividades, Meus Processos, Monitoramento de Processos, iniciação (*Start Point*) e finalização (*End Point*) dos processos.

Os grupos de dados mantidos nativamente pela ferramenta devem ser considerados como AIE – Arquivo de Interface Externa, a exemplo: cadastro de usuários e grupos organizacionais.

Não será considerada na contagem a interface de integração utilizada baseada apenas na tecnologia para acessar os dados mantidos em outras aplicações.

No caso do desenvolvimento de *webservices*, deve-se considerar:

Como Entrada Externa aqueles consumidores de informação a ser mantida no escopo da aplicação.

Como Consulta Externa ou Saída Externa aqueles consumidores de informação que não será mantida no escopo da aplicação.

Como Consulta Externa ou Saída Externa aqueles provedores de informação para sistemas externos.

Na primeira ocorrência de acesso a dados mantidos por aplicação externa, deve-se contabilizar os arquivos de interface externa associados.

Não serão consideradas na contagem funcionalidades pertencentes a outras aplicações que estão apenas sendo apresentadas na aplicação BPMS, uma vez que as regras de negócio e o processamento da informação são realizados nas aplicações de origem.

### 5.12.2 Critérios de Contagem

Os recursos de usabilidade não contabilizam pontos de função, pois constituem requisitos não funcionais, tais como:

Mecanismos de navegação;

Menus;

Telas de Ajuda (*Help*), *tooltips*, links com manuais e roteiros;

As atividades e subprocessos que são executados fisicamente ou de modo externo à fronteira da Aplicação não contabilizam pontos de função, a exemplo: encaminhar memorando, assinar contrato.

As funcionalidades que existem exclusivamente para a manutenção de dados de código não devem ser consideradas processos elementares, assim como os dados de código não devem ser considerados como arquivos referenciados nos processos elementares que os leiam e/ou atualizem.

Deve-se considerar na contagem como um único processo elementar as Atividades modeladas em BPMN, que apesar de se apresentarem distintas no modelo, constituem uma transação completa para o negócio, com sentido de completude de determinado requisito funcional para o usuário.

Não são contabilizadas na contagem de pontos de função as atividades que foram modeladas apenas para melhor entendimento do negócio e, que não constitui um processo elementar para o sistema, a exemplo das atividades “Selecionar Licenças” e “Renovar Licenças”. Apesar de serem duas atividades no modelo, para o negócio elas constituem apenas um processo elementar, individualmente elas não constituem uma transação completa.

Uma mesma atividade que pode ser realizada por pessoas/entidades diferentes deve ser contada apenas uma vez, a exemplo da atividade “Resolver Pendência” executada pela área jurídica, área de licitações e área técnica.

Não se deve contabilizar na contagem de pontos de função a notificação/lembrete (e-mail, sms ou outro meio) utilizadas em atividades que tenham um prazo para serem realizadas, onde após a expiração do prazo dessa notificação/lembrete é disparada.

Um exemplo consiste na atividade “Assinar Contrato” que tem um prazo de 01 (um) dia para ser realizada. Se o prazo expirar primeiro do que a conclusão da atividade, então será disparada uma atividade para notificar o responsável avisando do ocorrido e solicitando a sua realização. A notificação em si seja por email, sms ou algum outro meio, não tem sentido completo de negócio para o usuário, é apenas um procedimento integrante deste.

## **5.13 Aferição de Pontos de Função em ambiente de Apoio à Decisão (*Business Intelligence* e *Data Warehouse*)**

Esta seção visa apresentar as diretrizes de Contagem de Pontos de Função utilizadas no Ibama quanto a aferição do tamanho funcional de ambientes de *Data Warehouses* e *Business Intelligence*. Esta abordagem é reconhecida pelo IFPUG. As definições apresentadas têm como base o artigo “*Function Points & Counting Enterprise Data Warehouses*” Release 1.0 publicado pelo IFPUG [IFPUG, 2007] e Guia de Contagem de Pontos de Função do SISP para projetos DataWarehouse versão 1.0.

### **5.13.1 Introdução**

Em projetos de DW (*Data Warehouse*), a modelagem da base de dados (modelo multidimensional) tem um papel fundamental e exerce grande influência no desenvolvimento das funcionalidades que buscam atender aos requisitos do usuário com relação às consultas e relatórios, à análise de grandes volumes de dados e à obtenção de informações estratégicas para a tomada de decisão. Além disso, o processo de contagem do tamanho funcional de projetos de DW, usando a métrica Análise de Pontos de Função, baseia-se fortemente no modelo de dados multidimensional do DW, considerando as tabelas Fato e Dimensão definidas nesse modelo.

Em um projeto de DW observa-se que o esforço de desenvolvimento para as cargas de dados através do processo de ETL (Extração, Transformação e Carga) é maior (e mais complexo) do que o esforço para realização de outros serviços de desenvolvimento em um projeto de DW, como por exemplo, destaca-se a geração de consultas e relatórios no OLAP (*On-line Analytical Processing*), além da especificidade de profissionais de mercado que atendem essas demandas.

### **5.13.2 Modelo de Arquitetura de Referência de Data Warehouse / Data Mart**

Existem duas abordagens de desenvolvimento de um DW: Kimball e Inmon. Segundo NESMA (2014), as diretrizes de contagem são aplicáveis tanto no modelo Kimball quanto no Inmon. Inclusive, observa-se muito mais discussão e trabalhos sobre a medição funcional de Fatos e Dimensões em um Esquema Estrela (abordagem Kimball) do que sobre a medição de data

warehouses relacionais (abordagem Inmon). Além disso, no desenvolvimento de um DW usando a filosofia de Inmon, as orientações sobre a medição de Dimensões e Fatos são igualmente relevantes e aplicadas quando ocorre a criação de data marts (numa fase posterior ao DW).

Este guia adota a filosofia de Kimball, pois segundo NESMA (2014) observa-se que o Esquema Estrela (com Fatos e Dimensões) é mais utilizado do que um DW Relacional.

Na figura a seguir são apresentados os elementos que compõem o modelo de arquitetura de referência de projeto de DW.

A seguir, descreve-se os componentes da arquitetura de referência de projetos DW/DM apresentada na figura anterior.

#### Camada de Origem de Dados

Esta camada representa as fontes de dados que alimentam a base de dados do DW. Seu principal componente são os Sistemas Transacionais de Origem que representam todas as aplicações que fornecem dados para o DW, além de fontes como arquivo de dados nos formatos XML, XLS, TXT, ODS entre outros. Esta camada é usada, ainda, para a leitura de dados de sistemas de origem para a validação de dados no DW e para a geração de cubos.

#### Camada de Dados

Esta camada é subdividida em três subcamadas: Data Staging Area, Data Warehouse e Data Marts, responsáveis por armazenar os dados dos sistemas de origem e disponibilizá-los para consultas e relatórios.

O processo de disponibilização dos dados para consultas e relatórios envolve, principalmente, alguns componentes adicionais como o ETL (Extração, Transformação e Carga) e ODS (Operational Database Storage).

#### Data Staging Area (DSA)

A DSA representa uma base de dados temporária que armazena os dados extraídos dos sistemas transacionais de origem para realizar as transformações necessárias antes de carregar esses dados nas tabelas do DW. Em outras palavras, os dados são importados da aplicação de origem para a DSA e então, em outro processo de integração, os dados da DSA são importados para as tabelas Fato e Dimensão do DW.

#### Operational Database Storage (ODS)

Similar ao componente DSA descrito acima, o ODS é, também, uma base de dados que armazena os dados extraídos dos sistemas transacionais de origem antes de carregar esses dados nas tabelas do DW.

Entretanto, os dados da ODS podem sofrer atualização e, eventualmente, podem ser acessados pelo usuário final por meio de consultas *ad hoc*.

#### Extração, Transformação e Carga (ETL)

É o processo que compreende as três etapas (extração, transformação e carga de dados) para a disponibilização dos dados no DW a partir dos sistemas transacionais de origem.

#### Data Warehouse (DW)

O DW representa a base de dados que armazena várias versões integradas dos dados recebidos de diferentes fontes de origem, após o processo de ETL. Em um modelo de dados multidimensional do tipo Esquema Estrela, são reconhecidos dois tipos de entidades: Tabelas Fato e Tabelas Dimensão, sendo que as tabelas Dimensão podem ser classificadas em três tipos: Dimensão, Dimensão Estática e Dimensão Hierarquia.

**Tabela Fato:** Principal tabela em um modelo multidimensional. Sua finalidade é armazenar medidas do negócio. Cada tabela Fato pode ou não se conectar com outras tabelas Dimensão ou outras tabelas Fato.

**Tabela Dimensão:** As tabelas Dimensão detalham as informações contidas numa tabela Fato. Elas descrevem os fatos. Uma tabela Dimensão pode se conectar a mais de uma tabela Fato.

**Tabela Dimensão Estática:** As tabelas Dimensão Estática são tabelas criadas dentro do contexto do DW que detalham as informações contidas em uma tabela Fato mas que não são extraídas a partir de fonte de dados original. As tabelas Dimensão Estática podem ser carregadas manualmente ou por meio de um processo computacional (*procedure*). São exemplos de Dimensão Estática: códigos de estado e dimensão data e hora. A tabela Dimensão Estática implementa o conceito de Dados de Código (IFPUG, 2010).

**Tabela Dimensão Hierárquica:** É uma tabela Dimensão que não se relaciona diretamente com uma Tabela Fato. A tabela Dimensão Hierárquica é ligada a outra tabela Dimensão e representa um conjunto de dados que descrevem essa Dimensão.

**Tabela Dimensão Compartilhada:** É uma tabela Dimensão que atende a várias tabelas Fatos ou vários DW/DM e é usada para a validação de dados no processo ETL e para a geração de Cubos. As dimensões compartilhadas podem ser mantidas por outro DW ou por uma equipe específica. Devem ser atualizadas constantemente para atender as necessidades de todos os temas que as utilizam. A Dimensão Compartilhada é conhecida, também, como Dimensão Global ou Corporativa.

**Tabela de Agregação:** A característica dos DW é possuir uma grande quantidade de dados em grande nível de detalhe. Quando o tempo de resposta na recuperação de dados no DW trazem impacto ao negócio, aumentando o tempo de resposta das consultas, são utilizadas tabelas

sumarizadas. Elas agregam dados armazenados no DW em níveis mais elevados gerando melhor performance às consultas.

#### Data Marts (DM)

O DM representa um subconjunto de dados armazenados e localizados por assunto. Usado para agrupar os dados por departamento ou área de negócio de uma organização. Em um DM podem ocorrer agregações entre tabelas Fato e/ou Dimensão.

#### Cubos

A geração do Cubo, também, conhecido como Universo ou Contexto de Análise, representa a disponibilização das possíveis combinações, totalizações e medidas estatísticas básicas dos dados do DW em uma nova tabela, para acesso pela ferramenta OLAP, atendendo aos requisitos de negócio.

De uma forma geral, Kimball e Ross (2002) definem como sendo o nome de uma estrutura dimensional em uma plataforma de banco de dados de processamento analítico on-line (OLAP) ou multidimensional, originalmente referindo-se ao caso simples de três dimensões: produto, mercado e hora. Para complementar o entendimento, ainda em Kimball e Ross (2002), é dito que a modelagem dimensional se aplica a bancos de dados relacionais e multidimensionais. Embora ambos possuam um design lógico comum com dimensões que podem ser reconhecidas, a implementação física é diferente.

#### Camada de Aplicação

A camada de aplicação contém os relatórios pré-definidos pelo usuário e disponíveis para consulta dos dados armazenados no DW/DM.

#### Camada de Apresentação

A camada de apresentação é a interface do usuário final com o projeto de DW e, onde estão disponíveis várias funcionalidades como relatórios pré-definidos, relatórios *ad hoc*, relatórios operacionais (ODS) e mineração de dados.

### 5.13.3 Estimativa de Projeto de Data Warehouse

A contagem estimada de um projeto de DW consiste em identificar as funções de dados e funções transacionais a partir dos requisitos iniciais do sistema. Utiliza-se o método de Contagem Estimada de Pontos de Função da NESMA (2004), aplicando-se a complexidade baixa para as funções de dados e a complexidade média para as funções transacionais.

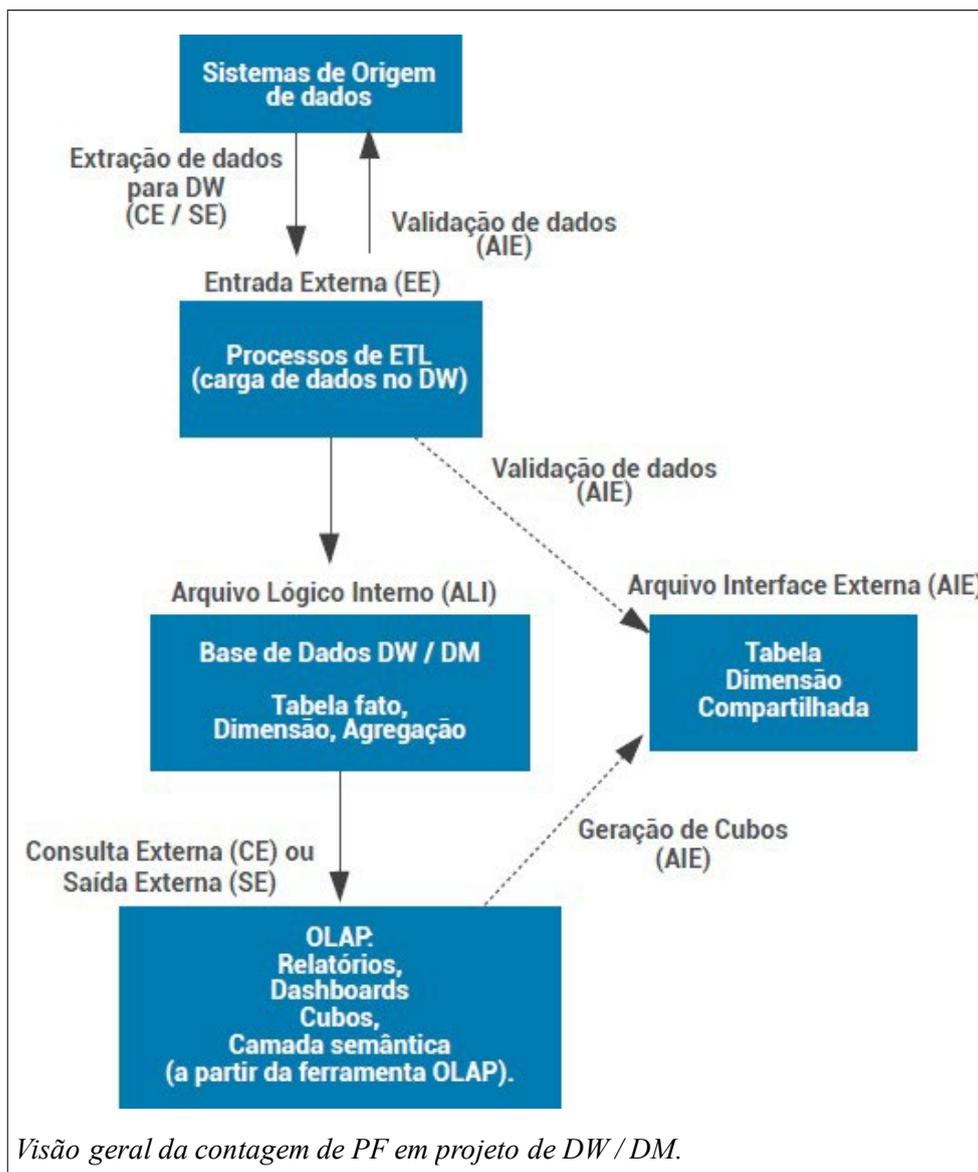
A Tabela seguinte apresenta, para a estimativa de projetos de DW, um resumo do que deve ser medido como função de dados e função transacional na contagem em PF.

Tipos de Funções	Complexidade	Tamanho PF	Contagem no DW
ALI	Baixa	7 PF	1 ALI para cada tabela Dimensão. 1 ALI para cada tabela Fato.
AIE	Baixa	5 PF	1 AIE para cada tabela do sistema de origem usada na validação de dados do processo de ETL. 1 AIE para cada tabela Dimensão Compartilhada usada na validação de dados do processo ETL ou geração do Cubo.
EE	Média	4 PF	1 EE para a carga de dados em cada ALI de tabela Dimensão. 1 EE para a carga de dados em cada ALI de tabela Fato.
SE	Média	5 PF	1 SE para cada Cubo (ou Universo ou Contexto de Análise). 1 SE para cada gráfico do <i>dashboard</i> solicitado.
CE / SE	Média	4 PF / 5 PF	1 CE ou SE para cada relatório solicitado (não <i>ad hoc</i> ). 1 CE ou SE para extração de dados de cada sistema de origem (quando houver requisito de desenvolvimento). Senão, não conta PF.

Tabela 2: Resumo da contagem estimada em PF de projeto de DW.

#### 5.13.4 Contagem de Projeto de Data Warehouse

A Figura seguinte apresenta uma visão geral da contagem de PF (funções de dados e funções transacionais) em um projeto de desenvolvimento de Data Warehouse / Data Mart (software analítico).



No contexto de contagem de PF de Data Warehouse, o escopo da contagem abrange, normalmente, o Projeto de Desenvolvimento ou Melhoria da fronteira do Data Warehouse e o Projeto de Melhoria realizado nas aplicações de origem para que estas forneçam informações para o Data Warehouse. Assim sendo, o Data Warehouse é definido como uma fronteira de aplicação distinta dos Sistemas de Origem, sendo necessária a contagem separadamente destas fronteiras.

#### 5.13.4.1 Contagem de Arquivos Lógicos Internos (ALI)

Em um modelo de dados multidimensional, são reconhecidos dois tipos de entidades: tabelas Fato e tabelas Dimensão. As tabelas Fato e Dimensão, mantidas por um ou mais processos de ETL, devem ser contadas como um ALI.

As tabelas Dimensão podem ser classificadas em Dimensão Estática e Dimensão Hierárquica. A tabela Dimensão Estática, se assemelha ao conceito de tabelas *Code Data* do CPM, armazenando dados estáticos e, portanto, não contribui para o tamanho funcional. As tabelas Dimensão Hierárquica ocorrem quando, no modelo de dados multidimensional, algumas tabelas Dimensão não estão ligadas diretamente na tabela Fato, e são consideradas níveis hierárquicos de uma outra tabela Dimensão. Deve observar a quantidade de Dimensões Hierárquicas ligadas a determinada Dimensão e contar um Registro Lógico para cada Dimensão Hierárquica ligada.

As hierarquias devem ser consideradas Registros Lógicos da Dimensão que se relaciona diretamente com a tabela Fato. Caso não existam níveis hierárquicos ou subgrupos de dados dentro da Dimensão, considere apenas um Registro Lógico.

As tabelas de Agregação geradas a partir da combinação de tabelas Fato e Dimensão e com o objetivo de apoiar a geração de consultas aos dados do DW, segundo NESMA (2014) podem ser contadas como ALI e sua carga de dados como uma EE, nos seguintes casos:

A) Quando existir cruzamento de dados de dois ou mais ALI do DW para gerar um novo Grupo de Informação, reconhecido pelo usuário.

B) Quando existir necessidade de se guardar os dados no DM cuja data de armazenamento no DW já está expirada.

Entretanto, uma tabela de Agregação ou um novo arquivo gravado para armazenar resultado de sumarizações, médias, máximas e percentagens de tabelas do DW, para atender requisitos não funcionais de desempenho, por exemplo, não é considerado um ALI.

Além disso, caso seja requisitado pelo usuário que os dados dos sistemas de origem do DW sejam disponibilizados na área de armazenamento temporária do DW chamada ODS (Operational Data Store) e esses dados não representam uma cópia dos dados do sistema de origem, mas são dados transformados e disponibilizados para consulta pelo usuário, então esses dados serão contados como ALI e as cargas desses dados serão contadas como EE.

#### **5.13.4.2 Contagem de Arquivos de Interface Externa (AIE)**

No processo de ETL, a validação de dados para a carga na tabela do DW pode requerer uma consulta à base de dados das aplicações de origem ou às tabelas Dimensão Compartilhada.

Observe que, nesses casos não há transferência de dados, e conta-se apenas a função EE referente ao processo de ETL, onde a validação de dados é parte desse processo.

Assim sendo, conta-se um AIE para cada tabela de dados consultada do sistema de origem ou da tabela Dimensão Compartilhada para a validação de dados do processo de ETL.

As tabelas Dimensões Compartilhadas são consideradas ALI de outro DW que as mantém e, são contadas como AIE no DW que as utiliza para validação de dados no processo ETL ou na geração de um Cubo.

### **5.13.4.3 Contagem de Funções Transacionais (EE, CE e SE)**

Em projetos de Data Warehouse, o processo de ETL é considerado um processo elementar único e representa a extração, transformação e carga de dados (IFPUG, 2007) nas tabelas Fato e Dimensão de um modelo multidimensional. O processo de ETL será denominado, neste Guia, de funcionalidade de carga de dados de uma tabela do DW e deve ser contada como uma Entrada Externa (EE).

A carga de dados periódica em uma tabela do DW pode ser do tipo carga completa (ou carga full) ou carga incremental (ou carga delta). A carga completa representa a exclusão dos dados da tabela para a inserção dos dados da extração. A carga incremental representa a atualização dos dados da tabela a partir dos dados da extração.

Frequentemente, tem-se apenas uma funcionalidade de carga de dados requisitada para cada tabela Fato e Dimensão do DW, podendo ser do tipo carga completa ou carga incremental. Entretanto, se houver requisito de negócio para o desenvolvimento das duas cargas (completa e incremental) para uma mesma tabela do DW, então, neste caso, conta-se duas EE (carga completa e carga incremental).

Os dados de uma mesma tabela Fato ou Dimensão podem ser carregados de vários sistemas transacionais de origem. Conta-se uma EE para a carga de dados de cada sistema transacional de origem distinto, observando o critério de unicidade de um processo elementar (IFPUG, 2010).

Considere a funcionalidade de carga de dados, também, para cada Registro Lógico da tabela Dimensão, onde os processos de manter os dados do Registro Lógico e da respectiva tabela Dimensão, são distintos e independentes.

Segundo NESMA (2014), geralmente, não existe funcionalidade de alteração de dados em projetos de DW e a exclusão de dados como requisito funcional, raramente, ocorre. Entretanto, quando ocorrer o requisito de usuário para a exclusão de dados pode ser tratado como uma funcionalidade de limpeza de dados (Exclusão de Dados Antigos do DW).

As cargas de dados em tabelas de Agregação do DW definidas como ALI, conforme descrito

no item “Contagem de Arquivos Lógicos Internos (ALI)”, também são contadas como uma EE.

Em aplicações de Data Warehouse existem funcionalidades de geração de relatórios (ou consulta de relatórios pré-definidos) ou gráficos (painéis, *dashboards*) usando as ferramentas OLAP. Cada relatório ou gráfico requisitado pelo usuário e implementado pela equipe de desenvolvimento são contados como SE ou CE. São CE quando a geração do relatório não possuir lógicas de processamento de cálculos ou criação de dados derivados, devendo-se, ainda, verificar se atendem os critérios de determinação da unicidade de processo elementar do CPM. Entretanto, os relatórios gerados pelo usuário por meio da funcionalidade de consultas *ad-hoc* ou personalizadas, disponível na ferramenta OLAP, não são contados, porque não constituem um requisito do usuário para a equipe de desenvolvimento.

A geração do Cubo (também denominado Universo ou Contexto de Análise) deve ser contada como uma SE. Esse tipo de tabela, normalmente, é utilizada para consumo por outras aplicações ou pelo próprio Datamart.

Conta-se uma SE para a geração do Cubo por cada Estrela modelada. Além dos Cubos por Estrela (uma tabela Fato e suas respectivas Dimensões), podem existir Cubos com mais de uma Estrela. Para estes casos, as tabelas Fatos e Dimensões envolvidas em cada Estrela devem ser consideradas como arquivo lógico referenciado, bem como as tabelas Dimensões Compartilhadas, sendo respeitadas as regras descritas nos itens referentes a contagem de funções de dados ALI e AIE.

A utilização da DSA é uma solução técnica, portanto não tem contagem de Pontos de Função.

Em alguns casos, o usuário com receio de perder dados das aplicações de origem, requisita que os dados dos sistemas de origem sejam copiados para uma área de armazenamento de dados operacional (*Operational Data Store – ODS*) do DW. Nestes casos os dados são copiados do sistema transacional de origem para o ODS e esses dados serão contados como DERs na função EE de carga no ODS. Caso os dados copiados sejam transformados, os dados do ODS serão contados como ALI e as cargas desses dados serão contadas como EE. Entretanto, as consultas e relatórios sobre esses dados do ODS são *ad hoc* e não serão contados como CE ou SE. Posteriormente, esses dados são armazenados em um ALI do DW (tabela Fato ou tabela Dimensão).

#### **5.13.4.4 Regras Resumo sobre Contagem em DW**

- Contar 1 ALI para cada tabela Fato;
- Contar 1 ALI para cada tabela Dimensão mantida por 1 ou mais processos de ETL;
- Contar 1 RL (registro lógico) para cada nível hierárquico de uma Dimensão;

- Contar 1 ALI para uma tabela de Agregação (entre Dimensões e/ou Fato) sem objetivo de desempenho;
- Contar 1 ALI para cada grupo de dados disponibilizado na área temporária ODS e que não representam apenas uma cópia dos dados dos sistemas de origem;
- Dimensão Estática não contribui para o tamanho funcional;
- Contar 1 AIE para cada ALI do sistema de origem usado na validação de dados do processo de ETL;
- Contar 1 AIE para cada tabela Dimensão Compartilhada usada na validação de dados do processo de ETL ou na geração de Cubos;
- Contar 1 EE para a carga de dados do processo de ETL, nas seguintes situações:
  - Para cada tabela Fato;
  - Para cada tabela Dimensão identificada como ALI (exceto Dimensão Estática);
  - Para cada Agregação identificada como ALI;
  - Para cada sistema transacional de origem de dados quando identificado o critério de unicidade de processo elementar (IFPUG, 2010) para a carga em cada tabela Fato ou Dimensão;
  - Para cada Registro Lógico (ou Dimensão Hierárquica) da tabela Dimensão (mantido independentemente da Dimensão), desde que não sejam Dimensões Estáticas;
- Contar 1 CE/SE para a extração de dados na fronteira do sistema de origem, se houver desenvolvimento da funcionalidade;
- Contar 1 CE/SE para cada relatório ou gráfico (painel, *dashboard*) requisitado pelo usuário e implementado pela equipe de desenvolvimento;
- Contar 1 SE para a geração de um Cubo;
- Não conte o mesmo grupo de dados como ALI e AIE;
- Não conte duas vezes o mesmo grupo de dados, por exemplo uma tabela Dimensão que esteja modelada em duas Estrelas do DW;
- As hierarquias das tabelas Dimensão devem ser contadas como Registros Lógicos e não como ALI. Por exemplo: Departamento e Setor.

#### **5.13.4.5 Considerações sobre Contagem em DW**

O projeto de melhoria de DW deve ser medido em PF conforme o item 4.2 Projeto de Melhoria.

### Alteração de Dados de Dimensões Estáticas

Em projetos de manutenção, a inclusão ou alteração de dados nas tabelas do tipo Dimensão Estática representam, normalmente, demandas de alteração da descrição associada ao código na tabela Dimensão Estática. Por exemplo: 1-homem, 2-mulher. Alterar a descrição dos códigos 1 e 2 para 1-masculino, 2-feminino.

Em analogia ao item Atualização de Dados (SLTI, 2012), conta-se 10% de uma EE de baixa complexidade, ou seja, 0,3 PF (0,10 x 3PF). Nesse caso, a fórmula é:

$$\text{PF\_Dimensão\_Estática} = 0,3 \text{ PF por Dimensão Alterada}$$

### Criação/Alteração de Campos em Tabelas Fato e Dimensão

É importante ressaltar que caso seja solicitada a criação de novos campos em tabelas Fato e Dimensão ou a alteração em campos (tipo, tamanho) existentes, deve-se considerar a contagem como PF\_ALTERADO e aplicar o Fator de Impacto (FI) conforme definido no Projeto de Melhoria de DW.

### Alteração de Dados em Tabelas Fato e Dimensão

A demanda de alteração de dados em tabelas Fato e Dimensão não é comum em projetos de DW.

Além disso, em simetria à Seção 4.10 Atualização de Dados, não será permitida tal ação, pois permitiria contar as alterações diretas em registros em banco de dados sem a utilização de *scripts*, ou seja, atualizações manuais, de forma interativa.

Assim, devido a questões de segurança, rastreabilidade e auditabilidade, não será permitida a utilização deste caso de contagem.

Para alterar registros de banco de dados (registro único ou registros que atendem a um critério de busca), tabelas Fato ou Dimensão, é necessário o desenvolvimento de *script*, de acordo com a Seção 4.9 – Apuração Especial.

### Extração de Dados em Tabelas Fato e Dimensão

A extração ou recuperação de dados de tabelas Fato e Dimensão não é uma demanda comum de ocorrer em projetos de DW, sendo requisitada pelo usuário, pois a ferramenta OLAP já fornece diversas formas e possibilidades de consulta aos dados do DW. Entretanto, se apesar disso, ainda houver requisito para o desenvolvimento de extração de dados, deve-se seguir o disposto na Seção

4.9 Apuração Especial. Um exemplo desse tipo de demanda é a extração de dados de um DW para a carga em outro DW.

#### Criação, Configuração e Disponibilização de um Filtro de Relatório

A criação de um filtro de relatório para o DW é interpretada de forma semelhante ao disposto na Seção 4.15 Componente Interno Reusável, de forma a não considerar a contagem de todos os relatórios/consultas impactados pelo filtro. Um filtro de relatório pode ser gerado pela combinação de uma ou mais tabelas Dimensão.

Deve-se utilizar a fórmula abaixo para a contagem desse tipo de demanda:

$$\text{PF\_Filtro\_Relatório} = 0,3 \text{ PF por Filtro}$$

#### Criação de Métricas

As métricas (fórmulas), criadas a partir de um requisito de usuário, são atributos lógicos associados a tabela Fato e são implementadas como DERs (Dado Elementar Referenciado) nas tabelas Fato. O processo de carga das tabelas Fato populam essas métricas que são totalizações ou cálculos a serem apresentados em relatórios gerados pelo usuário.

Caso o usuário solicite a criação ou alteração de uma métrica (por exemplo, sumarizações, médias, máximas e percentagens), deve-se avaliar o impacto nas funções transacionais que referenciam essa métrica. Essa solicitação pode envolver a criação de novo arquivo para armazenar resultado de sumarizações, médias, máximas e percentagens de tabelas do DW. A contagem de PF deve considerar, além da função de dados alterada (no caso, a tabela Fato), todas as funções transacionais impactadas pela nova métrica, aplicando-se o Projeto de Melhoria de DW.

#### Reorganização de Bancada

A reorganização de bancada ou reposicionamento de itens é uma demanda similar a manutenção em interface (bancada é uma interface para o usuário). Os itens da bancada são objetos da interface como, por exemplo: métricas, atributos, filtros, gráficos, tabelas e etc. Uma demanda deste tipo implica na necessidade de trocar a ordem dos itens/campos em relatórios e *dashboard* da bancada por solicitação do usuário.

Para este item, conta-se como Manutenção em Interface (4.7 Manutenção em Interface).

#### Exclusão de Dados Antigos do DW

As funcionalidades de limpeza de dados do DW estão, geralmente, associadas à administração do ambiente de DW, por exemplo, dados históricos com mais de 60 (sessenta) meses devem ser excluídos da base do DW.

Caso exista o requisito de exclusão de dados antigos (ou históricos) do DW, considerando, principalmente, o escopo de tabelas Fato, deve ser contado uma EE. Importante observar que, não necessariamente, identifica-se uma EE para cada ALI do DW.

#### Metadados Relacionados ao Controle do Processo de ETL

Os metadados relacionados ao controle do processo de carga de dados no DW correspondem aos dados usados para gerenciar o DW e podem ser, por exemplo: dados para controlar a adição de novos dados às tabelas do DW; o número de registros adicionados, alterados ou rejeitados; ou parâmetros usados para o processamento do ETL no DW.

Normalmente, a ferramenta de ETL já possui e disponibiliza esses controles (metadados) do processo de carga de dados no DW. Nesse caso, não há contribuição ao tamanho funcional. Senão, e caso essas funcionalidades sejam requeridas e desenvolvidas para apresentar ou editar esses dados de controle de forma diferente do disponível na ferramenta de ETL, então conta-se conforme o CPM.

De forma geral, sempre que as ferramentas e o ambiente de DW já disponibilizarem as funcionalidades e dados requisitados pelo usuário ou administrador do DW, então não há contagem de PF.

#### Metadados Relacionados ao Significado dos Dados – Metadados do Negócio

Os metadados do negócio descrevem o significado dos dados de um DW. Normalmente, tem-se o “rótulo” e a “descrição” do campo, representam as demandas para descrever atributos, métricas e pastas relacionadas a uma tabela Fato ou atributos e pastas associados a uma tabela Dimensão.

No projeto de desenvolvimento, esses metadados são parte do projeto de DW e, portanto, não conta PF. No projeto de manutenção, se houver demanda para alterar a “descrição” associada ao nome do campo criado, então, conta-se PF com as seguintes fórmulas:

**$PF\_Metadados\_Negocio = 0,2 \times Tamanho\_PF\_ALI$  (para tabelas Fato e Dimensão)**

**$PF\_Metadados\_Negocio = 0,3 PF$  por Dimensão Estática**

Se uma tabela Dimensão tiver mais de um registro lógico, então conta-se apenas uma vez esse ALI em  $PF\_Metadados\_Negocio$ .

### **Migração de Banco de Dados**

Essa demanda representa a necessidade de alteração do banco de dados (ou sua versão) que atende ao projeto de DW. Por exemplo, a migração do banco Oracle para o banco SQL Server.

Neste caso, conta-se esse serviço conforme descrito na Seção 4.5 Mudança de Plataforma.

### **Migração de ferramenta ETL**

A demanda de migração de ferramenta ETL implica em refazer um processo de ETL e, nesse caso, deve ser contado conforme descrito na Seção 4.5.1 Mudança de Plataforma – Linguagem de Programação.

## **5.13.4.6 Atividades Sem Contagem de Pontos de Função**

Nesta Seção, destacamos algumas atividades sem contagem de Pontos de Função que estão relacionadas ao ambiente produtivo do DW, como administração de usuários, segurança, customização de ferramentas OLAP e ETL para acesso de usuários, etc.

### **Tabelas Sem Contagem de Ponto de Função**

Algumas tabelas do DW não devem ser contadas como funções de dados em projetos de DW, a saber:

- Tabelas de repositório das ferramentas OLAP ou ETL não são contadas, mesmo que o usuário tenha necessidade de visualizá-las, pois, são desenvolvidas e mantidas pelas ferramentas, portanto, não são desenvolvidas pela equipe de Data Warehouse;
- Tabelas criadas para auxiliar o processo de transformação dos dados (tabelas da DSA ou temporárias), não são contadas, pois tratam-se de soluções técnicas;
- Tabelas Dimensão Estática;
- Tabelas de Agregação criadas para atender requisitos não funcionais, como desempenho.

### **Customização da Ferramenta OLAP**

Corresponde a customizações feitas em ferramentas OLAP para personalizações de forma a atender as necessidades do cliente. Por exemplo, alterações de uma página web

da ferramenta OLAP para a substituição do logo do fornecedor pelo logo do cliente; disponibilização de mensagens, dentre outras. Geralmente, essas customizações são semelhantes as customizações realizadas em desenvolvimento de software.

### **Integração da Ferramenta OLAP ao AD/LDAP**

A maioria das ferramentas OLAP permitem autenticação de usuário integrada com o servidor AD/LDAP da organização. Para isso é necessário realizar algumas configurações na ferramenta OLAP para obter este tipo de integração, onde o DW torna-se cliente do servidor AD/LDAP para validar o acesso de seus usuários.

Essa atividade está associada a ações administrativas do DW e permissão de uso na ferramenta OLAP, independente do tipo de acesso. Não conta Ponto de Função, pois está associado a infraestrutura de segurança do órgão.

### **Customização da Ferramenta ETL**

De forma similar ao item Customização da Ferramenta OLAP, a customização da ferramenta de ETL corresponde a customizações/personalizações feitas em ferramentas ETL de forma a atender as necessidades do cliente. Por exemplo, alterações de uma página web da ferramenta ETL para a substituição do logo do fornecedor pelo logo do cliente; disponibilização de mensagens, dentre outras necessidades.

### **Criação de usuários e perfis de acesso às ferramentas OLAP e ETL**

A criação de usuários e perfis de acesso às ferramentas OLAP e ETL são atividades que fazem parte da administração do ambiente de DW e, normalmente, estão sob a responsabilidade da área de infraestrutura e produção dos sistemas do órgão.

Portanto, esses serviços de criação de usuários e perfis de acesso às ferramentas OLAP e ETL não devem ser contados em PF.

### **Automação do Processo de ETL**

Essa atividade representa a implantação de um processo de ETL no DW. Constitui a criação dos *jobs* ou processos/*scripts* que fazem o encadeamento/automatização de todos os processos de carga das tabelas Fato e Dimensão.

A automação do processo de ETL é uma atividade do ambiente produtivo do DW e, portanto, sem contagem de PF.

## **5.13.1 Fronteira de Aplicação**

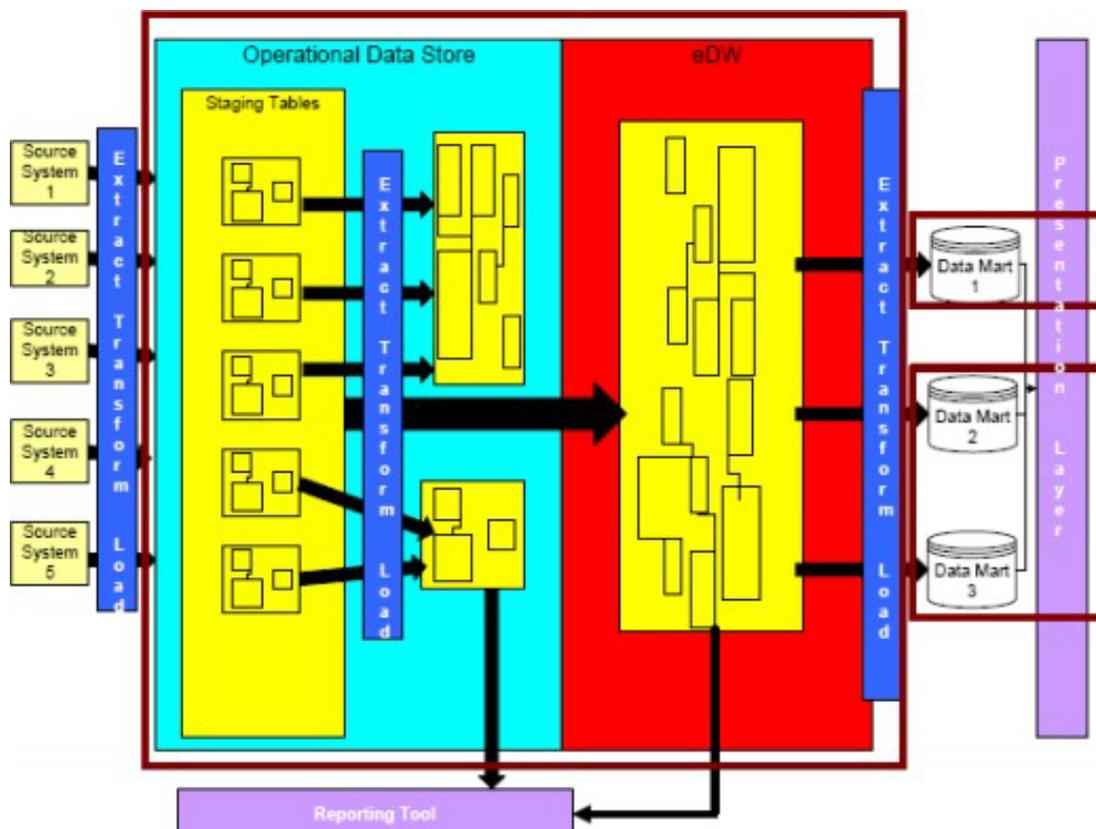
Para a contagem de pontos de função de *Data Warehouse (DW)* e *Business Intelligence (BI)* deve-se considerar os limites:

- A fronteira é determinada com base na visão do usuário. O enfoque é no que o usuário pode entender e descrever.
- A fronteira entre as aplicações relacionadas baseia-se em áreas funcionais separadas como visto pelo usuário, não em considerações técnicas.
- A fronteira inicial já estabelecida para a aplicação ou aplicações sendo modificadas não são influenciadas pelo escopo de contagem.

Não se deve considerar na contagem de pontos de função os seguintes elementos da aplicação:

- Arquivos lógicos mantidos pelo(s) sistema(s) de origem, exceto aqueles referenciados nas funções transacionais;
- Tabelas Temporárias;
- Staging Areas;
- Tabelas de códigos;

Os *Data Marts* podem ser contados como fronteiras de aplicações separadas.



A *Staging Area*, usada para armazenar uma versão atual do *Data Warehouse* que existe no sistema original, não deve ser contabilizada na contagem pois não é reconhecida pelos usuários.

O ODS (Depósito de Dados Operacionais), área que contém dados transacionais detalhados que são tipicamente modificados, não deve ser contabilizado na contagem.

## 5.13.2 Critérios de Contagem

### FUNÇÕES DE DADOS

Conte um Arquivo referenciado (ALR) para cada ALI mantido.

Conte um Arquivo referenciado (ALR) para cada ALI ou AIE referenciados durante o processamento da entrada externa.

Conte um AIE para cada grupo lógico de dados que é copiado de um sistema de origem para o *Data Warehouse* sem nenhuma lógica especial de processamento.

Conte um ALI para cada tabela fato e dimensão mantidas pelo DW. Sendo que as tabelas dimensão essencialmente estáticas não devem ser contadas, pois configuram *CODE DATA*.

Não conte o mesmo grupo de dados com ALI e AIE.

Não conte duas vezes o mesmo grupo de dados como uma tabela dimensão que esteja modelado em duas estrelas do DW.

As hierarquias das tabelas dimensão devem ser consideradas como Tipo de Registro (TR) e não como ALI, a exemplo: departamento e setor.

### FUNÇÕES TRANSACIONAIS

Cada processo completo de carga de dados em tabela fatos e dimensão é contado como apenas uma Entrada Externa.

Não conte três EE separadas para cada passo do processo (ex.: uma EE de Extração, uma EE de Transformação, e uma EE de Carregamento), uma vez que todos os três são requeridos para completar o processo elementar.

Conte pelo menos uma SE ou CE para cada relatório ou consulta desenvolvida e/ou suportada para satisfazer as necessidades do usuário.

As ferramentas de relatórios inerentes ao produto de BI ou DW não são contabilizadas, ou seja, as consultas e recursos gerados pelo usuário não devem ser contabilizados.

As funções de conversão de dados (carga inicial de dados) são contadas como EE.

As cargas de dados em tabelas dimensão são contadas como EE, mesmo que tais tabelas sejam consideradas Tipos de Registro (TR).

A geração de contexto de análise (Cubos) deve ser contada como SE.

## 5.14 Aferição de Pontos de Função em Ambiente Baseado em Geotecnologias

Sistemas baseados em geotecnologias são ferramentas utilizadas para análise de informações geográficas que usam funções de dados geométricos ligados a tabelas de atributos alfanuméricos. O objetivo deste item é esclarecer e definir diretrizes para medição de Pontos de Função a sistemas georreferenciados do IBAMA, a diversos cenários não devidamente tratados pelo CPM (manual oficial de pontos de função do IFPUG), tais como:

1. Cenários de georreferenciamento cuja interpretação em pontos de função possa acarretar dúvidas e potenciais divergências entre os analistas;
2. Cenários em que a identificação de processos elementares georreferenciados precisa refletir corretamente o real esforço de desenvolvimento, a despeito de considerar apenas as diretrizes existentes no CPM;

A medição de pontos de função para sistemas georreferenciados devem ser feitas conforme as diretrizes abaixo. Para demais situações não previstas nos cenários abaixo, estas devem ser medidas segundo as diretrizes dos manuais CPM 4.3 e SISP 2.2.

### 5.14.1 Medição de camadas georreferenciadas

Este tópico se refere a Camadas de georreferenciamento, tais como “camadas overlayer”, “baselayer”, entre outras.

Existem características comumente apresentadas em sistemas de georreferenciamento, a saber:

- A. Cada camada é geralmente obtida de outra aplicação, distinta das demais camadas;
- B. Cada camada é individualmente reconhecida pelo usuário da aplicação, fazendo parte da área negocial da aplicação;
- C. cada camada constitui uma estrutura de dados (Tipos de Dados, segundo o CPM) bem heterogênea e distinta das demais;
- D. Estas camadas são grupos de dados independentes entre si, não constituindo entidades dependentes de qualquer tipo.

Interpretando essas características sob a ótica de pontos de função, cada camada de georreferenciamento será 1 arquivo lógico independente.

Pelo item “C” acima, nota-se que os Tds de cada camada, via de regra, são distintos e incompatíveis, não fazendo sentido por exemplo agrupar “todas as camadas de overlayer do sistema” em um único arquivo lógico “Camada Overlayer”, nem agrupar “todas as camadas baselayer” em um arquivo lógico “Camada Baselayer”.

Ademais, em estudo técnico preliminar, constatou-se que camadas georreferenciadas apresentam um elevado custo de desenvolvimento.

Desta forma, com intuito de promover a exequibilidade de contratos georreferenciados por PF, a diretriz a seguir deverá ser aplicada:

Cada camada georreferenciada deverá constituir 1 arquivo lógico (ALI/AIE) distinto e a complexidade deve ser analisada em cada caso, segundo a complexidade do CPM.

### 5.14.2. Apresentação de dados em formatos diferentes (multiple instance)

Este tópico trata o cenário em que um mesmo processo elementar é apresentado em mídias/formatos diferentes, conforme o conceito “multiple instance” do SISP.

Considere uma consulta ‘C’ feita em um grupo de dados georreferenciado E considere que o sistema possua as funcionalidades:

- Apresentação destes dados da consulta ‘C’ em forma de gráfico;
- Apresentação desta mesma consulta ‘C’ em forma de tabela;
- Apresentação desta mesma consulta ‘C’ em forma de mapa;

*Nota: “mapas, tabelas e gráficos” se enquadram no conceito de “multiple instance” do SISP. (Vide maiores detalhes no item “5.1 Contagem de Pontos de Função com Múltiplas Mídias” do “Guia de Métricas SISP v2.2”).*

Tais funcionalidades contém os mesmos TDs, mesmos ARs e mesmas lógicas de processamento, constituindo contudo formas de apresentação distintas (gráfico, tabela e mapa) do tipo “multiple instance”.

Segundo o CPM, como os requisitos funcionais das três funcionalidades são idênticos, deve-se contar um único processo elementar para elas.

A apresentação de uma mesma consulta de dados em dois formatos rigorosamente distintos (i.e. “formato *multiple instance*” - gráfico, tabela e mapa) acarreta efetivamente o dobro do esforço de desenvolvimento.

Logo, para o cenário de representação de grupos georreferenciados idênticos em 2 formatos “multiple instance”, segundo este guia, deverão ser contados 2 processos elementares distintos.

*Nota: esta solução é respaldada pelo “Guia de Métricas SISP”, que defende a contagem de 2 transações para um mesmo processo elementar exibido em 2 mídias distintas, do tipo “multiple instance”. (Vide maiores detalhes no item “5.1 Contagem de Pontos de Função com Múltiplas Mídias” do “Guia de Métricas SISP v2.2”).*

### 5.14.3. Apresentação simultânea de vários elementos visuais

Este tópico trata cenários em que vários “elementos visuais georreferenciados” são exibidos simultaneamente (e.g. na mesma tela) representando um processo elementar único na visão do usuário, e não N funcionalidades autocontidas para os N elementos visuais.

**Segundo este guia, deve-se contar 1 processo elementar para cada “elemento visual georreferenciado”, mesmo quando eles forem exibidos simultaneamente em uma mesma funcionalidade.**

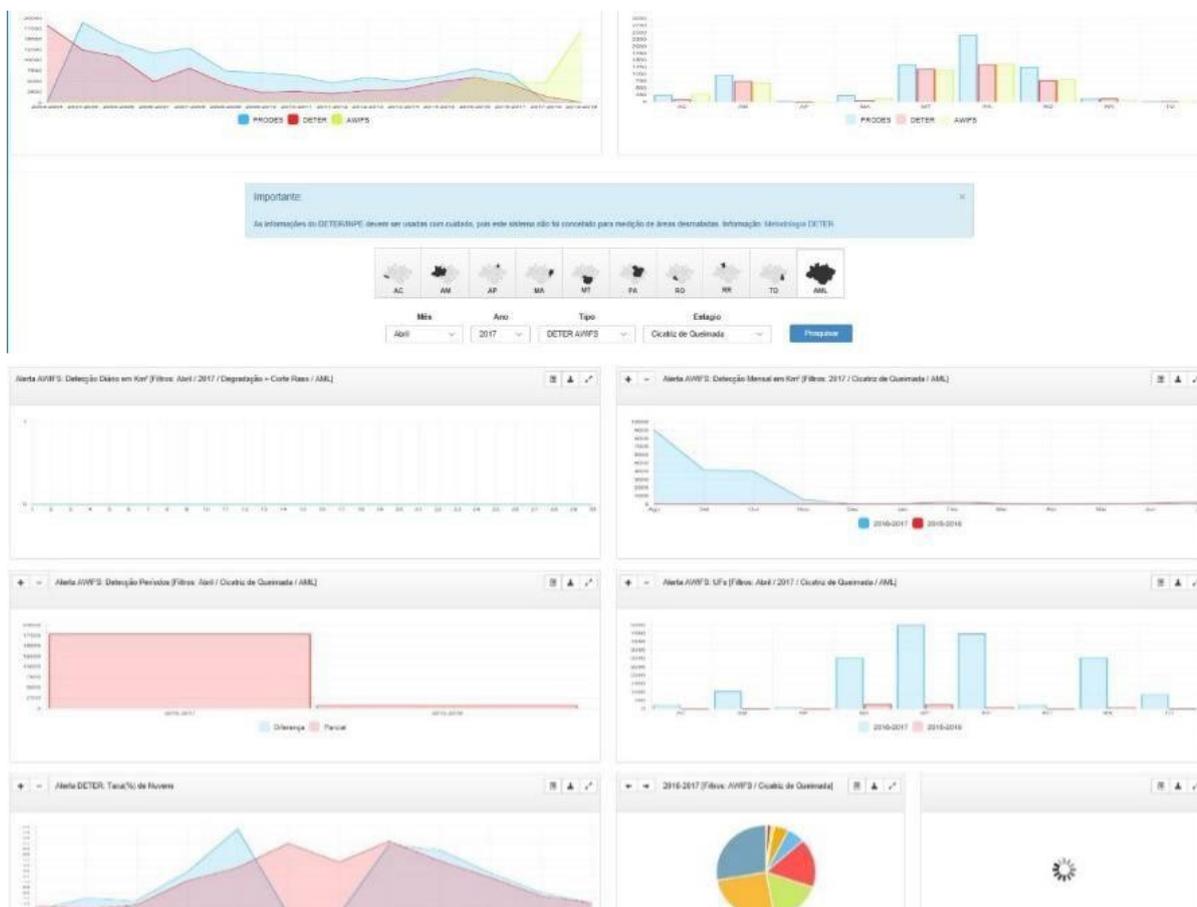
Nota: Entenda por “elemento visual georreferenciado” uma consulta feita e exibida em um formato visual (e.g. em gráfico, em tabela ou em mapa). Por exemplo, o **gráfico** da figura a seguir representa a consulta de dados em 1 (um) “elemento visual georreferenciado” (note que esta mesma consulta poderia ser representada em forma de **tabela**, **mapa**, etc).

## Exemplo

Este cenário será melhor compreendido no exemplo a seguir:

A imagem abaixo apresenta 8 “elementos visuais georreferenciados” (i.e. 8 consultas exibidas em forma de gráficos) exibidos como uma consulta única e autocontida (na qual o usuário aciona um comando e o sistema exibe os 8 gráficos de uma vez). Desta forma, o CPM reconhece a existência de um único processo elementar.

O desenvolvimento de um único “elemento visual georreferenciado” representa um esforço elevado. Portanto, idealmente deve-se considerar 1 processo elementar a cada “elemento visual georreferenciado” (e não 1 processo elementar para toda a tela).



## 5.14.4. Funcionalidades complexas de georreferenciamento

Sejam conceitualmente as seguintes funcionalidades georreferenciadas:

- I. Funcionalidades que disponibilizam a “edição de desenhos de elementos em mapas dentro do próprio sistema”, tais como:
  - a) Inclusão de elementos (polígono, ponto, reta...) no mapa – efetuada via desenho em mapa
  - b) Inclusão de elementos (polígono, ponto, reta...) no mapa – efetuada via coordenadas do elemento
  - c) Alteração de elementos (polígono, ponto, reta...) no mapa – efetuada via desenho em mapa
  - d) Alteração de elementos (polígono, ponto, reta...) no mapa – efetuada via coordenadas do elemento
  - e) Exclusão de elementos (polígono, ponto, reta...) no mapa – efetuada via desenho em mapa
  - f) Exclusão de elementos (polígono, ponto, reta...) no mapa – efetuada via coordenadas do elemento
  
- II. Funcionalidades que disponibilizam a manipulação de dados geo por arquivos shapefile, tais como:
  - a) Funcionalidade para gerar, armazenar e disponibilizar em um “arquivo shapefile” as informações geoespaciais obtidas de uma fonte de dados (e.g. uma “camada geo”, uma representação visual de mapa contendo vários elementos e/ou camadas geo, um conjunto de dados geoespaciais armazenados em banco, etc.);
  - b) Funcionalidade para efetuar download de arquivo shapefile;
  - c) Funcionalidade para efetuar upload de arquivo shapefile (incluindo validações negociais dos atributos do arquivo);
  
- III. Funcionalidade para visualização mapas na tela (com ou sem filtros de pesquisa), junto aos seus Arquivos Lógicos Referenciados (camadas, elementos, etc).
  
- IV. Funcionalidade onde o usuário solicita um zoom no mapa para uma localização / área específica.

Desta forma, com intuito de promover a exequibilidade de contratos georreferenciados por PF, a diretriz a seguir deverá ser aplicada:

A ocorrência de cada funcionalidade individual acima (e outras funcionalidades similares) deverá ser contada como um processo elementar e a complexidade deve ser analisada em cada caso, segundo a complexidade do CPM.

Mesmo nos casos em que estas não sejam autocontidas, ou constituam apenas lógicas de processamento de uma função transacional.

Isto é, considere as funcionalidades supracitadas sempre como funcionalidades *autocontidas* e que deixem o sistema em *estado consistente* (ou seja, como “processos elementares”).

Nota: Caso, segundo a abordagem clássica do CPF, as funcionalidades supracitadas constituam lógicas de processamento de uma função transacional F, esta última (F) deverá ser identificada e classificada segundo as regras do CPM.

Exemplo:

Em um determinado sistema de georreferenciamento, dadas as funcionalidades:

“inclusão de download de shapefile”,  
“inclusão de upload de shapefile”,  
“incluir mapa via desenho”  
“incluir mapa via coordenadas”  
“exibir mapa com elemento existente no shapefile”

Considere que, segundo as regras preliminares do CPM, as 5 funcionalidades acima não se classifiquem em processos elementares (i.e. não sejam autocontidas, etc.), mas constituam lógicas de processamento contidas em uma função transacional “incluir projeto de mapas”.

Neste cenário, segundo este guia, deverão ser medidas 6 funções transacionais:

EE: inclusão de download de shapefile  
EE: inclusão de upload de shapefile  
EE: incluir mapa via desenho  
EE: incluir mapa via coordenadas  
EE: exibir mapa com elemento shapefile  
EE: incluir projeto de mapas (caso esta seja corretamente identificada como função transacional, segundo as regras do CPM).

## 6. Contagem de Pontos de Função de Portais

Seguem as orientações para contagem de Pontos de Função de Portais utilizando ferramenta CMS.

O relatório RAG 5 - “Relatório de Avaliação por Área de Gestão Nº 5” elaborado pela Controladoria Geral da União (CGU) identifica quatro atividades necessárias à disponibilização de portais com utilização de CMS, a saber:

- a. Implantação e configuração: realização da instalação do sistema no ambiente tecnológico do órgão/entidade, bem como da configuração necessária ao pleno funcionamento do CMS. Destaca-se que grande parte da configuração é realizada a partir da edição de arquivos XML;
- b. Adaptação (customização) do sistema: implementação de novas funcionalidades no sistema, bem como de adaptações nas funcionalidades já existentes, de forma a atender às necessidades específicas do órgão/entidade;
- c. Apresentação visual: definição de identidade visual dos portais;
- d. Gestão de conteúdo: Edição e manutenção do conteúdo a ser disponibilizado nos portais pelos próprios usuários do órgão/entidade.

Além disso o Relatório menciona o seguinte:

Os serviços de Implantação e Configuração (a), bem como aqueles relacionados à Apresentação visual (c) e à Gestão de conteúdo (d), claramente não se enquadram como

desenvolvimento ou manutenção de funcionalidades de um sistema. Estas atividades devem ser mensuradas em hora-análise.

A Adaptação (customização) do sistema (b) será mensurada em Pontos de Função de acordo com as orientações deste guia.

As funcionalidades da ferramenta CMS **APENAS** disponibilizadas para o usuário não serão mensuradas em Pontos de Função. Estas funcionalidades se enquadram no item (a) Implantação e configuração e serão mensuradas em hora-análise. O Documento de Visão do Portal deverá registrar que estas funcionalidades do Zope/Plone serão apenas disponibilizadas. Não serão construídos casos de uso para estas funcionalidades. Elas podem ser registradas no documento de requisitos não funcionais.

As funcionalidades desenvolvidas ou mantidas pela Contratada serão contadas de acordo com as diretrizes do CPM 4.3.1 e do Roteiro de Métricas IBAMA. As funcionalidades da ferramenta Zope/Plone customizadas para o usuário são mensuradas em Pontos de Função. Observe que estas funcionalidades são requisitadas e recebidas pelo usuário. Estas funcionalidades devem ser documentadas na Especificação de Casos de Uso ou outra documentação de requisitos funcionais do sistema.

A implementação de funcionalidades de portais Zope/Plone geralmente acessam o arquivo de usuários. No entanto, na visão do usuário, os dados de usuário já foram passados para a aplicação no momento do Controle de Acesso. A referência aos dados de usuário, por exemplo, pela funcionalidade Incluir Banner é uma implementação de um requisito técnico de segurança. Desta forma, o ALI usuário não pode ser contado como Arquivo Referenciado na funcionalidade incluir Banner. Os tipos de dados de usuário que não atravessam a fronteira da aplicação também não devem ser contados.

Em requisitos de manutenção de cadastros, por exemplo manter notícias, deve-se observar as funcionalidades entregues para todos os perfis de acesso. Por exemplo, a consulta lista de documentos para download é idêntica para o perfil de acesso gestor e usuário comum. Então, contar apenas uma vez. No entanto, a consulta de notícias do usuário gestor, apresenta dados referentes ao histórico de publicação de notícias, tais como: autor, revisor, publicador, data da revisão, e a consulta de notícia do usuário não apresenta tais informações. Assim, devem ser contadas duas funcionalidades distintas: CE/SE: Consulta de Notícias – Usuário; CE/SE: Consulta de Notícia – Gestor.

As demandas de migração de conteúdo de Portais devem ser contadas como Apurações Especiais geração de relatório e base de dados, da seguinte forma: No portal antigo, a necessidade de exportação de conteúdo do portal antigo é contada como CE ou SE. No

portal novo, a necessidade de importação de conteúdo do portal novo é contada como EE. Em alguns casos, há necessidade de criação de ALI no portal novo, então este deverá ser contado.

As demandas de instalação de plug ins ou componentes na ferramenta CMS devem ser mensuradas em hora-análise.

As demandas de alteração do Painel de Controle podem ser contadas como PF Componente ou mensuradas em hora-análise.

As atualizações de arquivo de configuração do cliente podem ser contadas como PF Componente ou mensuradas em hora-análise.

As demandas de Análise de Solução devem ser mensuradas em hora-análise.

As necessidades de testes funcionais em funcionalidade por conta de instalação de plug in devem ser dimensionadas com PF Testes.

A customização do Plug in pode ser mensurada como PF Componente ou mensurada em hora-análise.

A configuração do Plug in pode ser mensurada como PF Componente ou mensurada em hora-análise.

A instalação de plug in do CMS disponível pela Comunidade com necessidade de customização pode ser mensurada em: Instalação – hora-análise; Customização do Plug in: PF Componentes; Adaptação das funcionalidades do Portal: Manutenção Adaptativa em Requisitos Não Funcionais; Teste de funcionalidades de o Portal: PF Testes.

As demandas de atualização do CMS podem ser mensuradas como PF Componente ou mensuradas em hora-análise.

## **7. Considerações Especiais para Planejamento e Acompanhamento de Projetos**

Este capítulo tem como propósito apresentar diretrizes para o planejamento e acompanhamento de projetos com o auxílio da métrica Ponto de Função e de técnicas relacionadas. Com base nesta finalidade é descrito um processo de estimativas de projetos de software aderente à área de processo de Planejamento de Projeto do CMMI (*Capability Maturity Model Integration*). Nesse contexto, é apresentado o método Contagem Estimativa de Pontos de Função (CEPF) para estimar o tamanho dos projetos de software em PF. Também são apresentadas recomendações

para o gerenciamento de: mudanças de requisitos, projetos cancelados e progresso de projetos, e considerações sobre redução de cronograma e fator de criticidade de solicitação de serviços.

## 7.1 Diretrizes para Planejamento: Estimativas de Projetos de Software

Esta seção define métodos para estimativas de projetos de software.

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 a ser utilizado é um documento inicial de requisitos, por exemplo, o 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 prazo (cronograma) e custo (orçamento) com base na estimativa de tamanho e nos dados históricos de projetos concluídos da organização. Neste ponto, as principais estimativas foram geradas e precisam ser documentadas. As premissas e suposições utilizadas na geração das estimativas, dentre as quais: 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, o projeto deve ser reestimado sempre que ocorrerem mudanças significativas nos requisitos funcionais ou não funcionais. Quando o projeto é concluído, deve-se aferir e documentar o tamanho, prazo e custo, 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].

As estimativas devem ser realizadas em no mínimo dois 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 como, por exemplo, o 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 às 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 Final:** realizada após a homologação da aplicação. Esta contagem considera as funcionalidades efetivamente entregues para o usuário pela aplicação. Neste caso, deve ser aplicada a contagem detalhada.

Para fins de faturamento, deve-se considerar a Contagem Final.

É importante ressaltar que as evoluções de requisitos (*scope creep*) devem ser consideradas na oficialização de abertura das demandas. Contudo, durante a execução do projeto, se estas mudanças forem significativas, maiores que a estimativa inicial, deverão ser tratadas como novo projeto.

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

### 7.1.1 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.

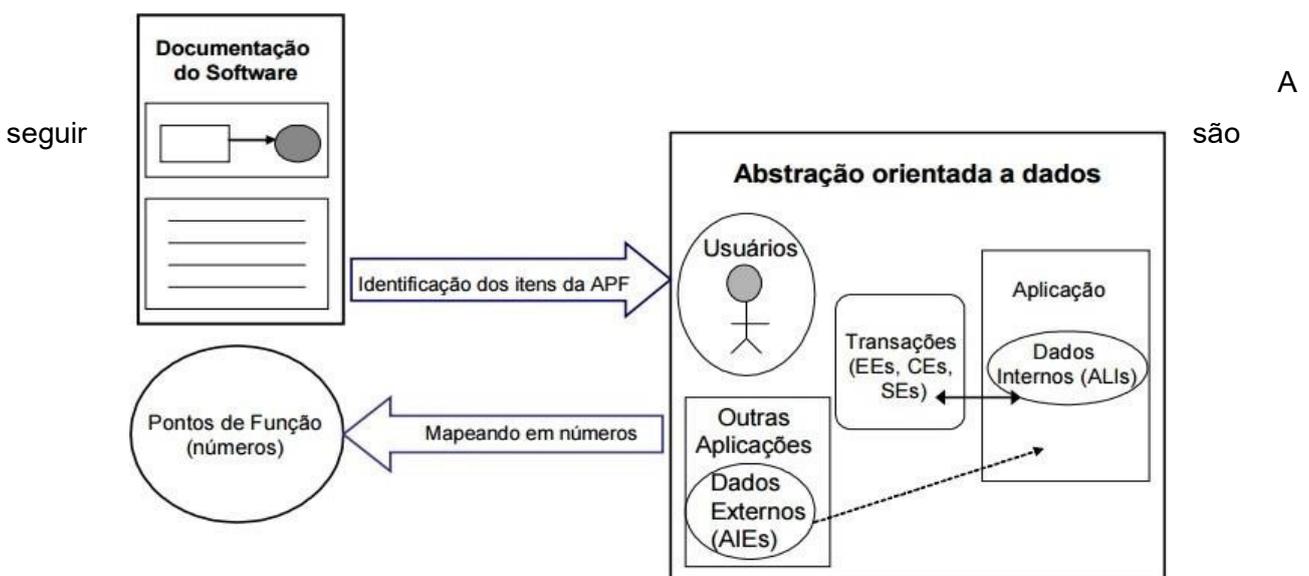
O método CEPF visa aferir o tamanho em PF de maneira simplificada, com base no conhecimento dos requisitos iniciais do projeto [Hazan, 2005]. A CEPF foi definida com base nas diretrizes adotadas no método Contagem Estimada de Pontos de Função da NESMA [NESMA, 2005]. A diferença é que o método da NESMA não recomenda a análise das funções identificadas, considerando todas as funções de dados identificadas com complexidade Baixa e as funções transacionais com complexidade Média. A CEPF propõe a análise das funcionalidades identificadas, e caso não seja possível determinar a complexidade, então são adotadas as diretrizes do método Contagem Estimada da NESMA. A CEPF também apresenta algumas dicas para ajudar um

estimador no mapeamento dos requisitos iniciais nos tipos funcionais da Análise de Pontos de Função. Segue a descrição da CEPF [Hazan, 2005].

Primeiramente, conforme Figura 2, 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 do 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.

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 das funções transacionais ou de dados deve-se utilizar a complexidade Média.



**Modelo Lógico da Análise de Pontos de Função**

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 (ALI):** 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 m x n 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, deve-se utilizar a complexidade Baixa.

Nº ALI Baixa:	X 7 PF
Nº ALI Média:	X 10 PF
Nº ALI Alta:	X 15 PF
Total PF:	

**Tabela 3: Identificação dos Arquivos Lógicos Internos da Aplicação**

**Tabela 3 – Contagem de Arquivos de Interface Externa (AIE):** 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 AIE. Não são considerados AIE arquivos físicos, arquivos de índice, arquivos de trabalho, tabelas de relacionamento sem atributos próprios e entidades fracas.

Nº AIE Baixa:	X 5 PF
Nº AIE Média:	X 7 PF
Nº AIE Alta:	X 10 PF
Total PF:	

**Tabela 4: Identificação dos Arquivos de Interface Externa da Aplicação**

**Tabela 4 –** Contagem de Entradas Externas (EE): funcionalidades que mantêm os Arquivos Lógicos Internos (ALI) 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, 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 Entradas Externas identificadas com complexidade Média.

Nº EE Baixa:	X 3 PF
Nº EE Média:	X 4 PF
Nº EE Alta:	X 6 PF
Total PF:	

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

**Tabela 5 –** Contagem de Consultas Externas (CE): funcionalidades que recuperam dados de pelo menos 1 ALI ou 1 AIE e apresentam 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 e 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**.

Nº CE Baixa:	X 3 PF
Nº CE Média:	X 4 PF
Nº CE Alta:	X 6 PF
Total PF:	

**Tabela 6: Identificação das Consultas Externas da Aplicação**

**Tabela 6** – Contagem de Saídas Externas (SE): 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. Se você não possui conhecimento sobre o processo elementar (funcionalidade analisada), considere as Saídas Externas com complexidade **Média**.

Nº SE Baixa:	X 4 PF
Nº SE Média:	X 5 PF
Nº SE Alta:	X 7 PF
Total PF:	

**Tabela 7: Identificação das Saídas Externas da Aplicação**

A estimativa de tamanho do projeto em PF deve ser gerada com a totalização dos PF 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:

$$PF\_DESENVOLVIMENTO = PF\_INCLUIDO + PF\_CONVERSÃO$$

### 7.1.2 Distribuição de Esforço por Fase do Projeto

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

A Tabela 7 apresenta a distribuição de esforço por macroatividade do Projeto.

Macroatividades do Processo de Desenvolvimento de Software	Percentual de esforço (%)
Engenharia de Requisitos	25%
Design / Arquitetura	10%
Implementação	40%
Testes	15%
Homologação	5%
Implantação	5%

**Tabela 8: Distribuição de Esforço por Macroatividades do Projeto**

### 7.1.3 Definição de Prazo de Projetos de Software

A tabela a seguir apresenta o prazo máximo de execução das OS, de acordo com o tamanho estimado do projeto constante na OS.

Tamanho do Projeto	Prazo Máximo
≤10 PF	9 dias úteis
> 10 PF e ≤ 20 PF	18 dias úteis
> 20 PF e ≤ 30 PF	27 dias úteis
> 30 PF e ≤ 40 PF	36 dias úteis
> 40 PF e ≤ 50 PF	45 dias úteis
> 50 PF e ≤ 60 PF	54 dias úteis

> 60 PF e ≤ 70 PF	63 dias úteis
> 70 PF e ≤ 85 PF	70 dias úteis
> 85 PF e ≤ 100 PF	79 dias úteis
> 100 PF e ≤ 125 PF	101 dias úteis
> 125 PF e ≤ 150 PF	123 dias úteis
> 150 PF e ≤ 175 PF	131 dias úteis
> 175 PF e ≤ 200 PF	138 dias úteis
> 200 PF e ≤ 225 PF	144 dias úteis
> 225 PF e ≤ 250 PF	149 dias úteis
> 250 PF e ≤ 275 PF	155 dias úteis
> 275 PF e ≤ 300 PF	160 dias úteis
> 300 PF e ≤ 350 PF	167 dias úteis
> 350 PF e ≤ 400 PF	175 dias úteis
> 400 PF e ≤ 450 PF	183 dias úteis
> 450 PF e ≤ 500 PF	191 dias úteis
> 500 PF e ≤ 600 PF	200 dias úteis

**Tabela 9: Definição de Prazo de Projetos para OS de desenvolvimento ou manutenção evolutiva.**

O prazo considera todo o ciclo de vida do projeto, desde a fase de requisitos até a implantação. Demandas com volume maior que 600 Pontos de Função deverão ser divididas em OS de volumes menores e adequadas à tabela apresentada.

Caso seja necessário receber o projeto em um prazo menor, recomenda-se propor um processo de desenvolvimento incremental, priorizando funcionalidades em cada iteração de acordo com a necessidade dele. Caso, ainda assim, o prazo definido não atenda às necessidades do cliente, pode-se reduzi-lo prazo em até 25%, observando-se o conceito de Região Impossível proposto por Capers Jones [Jones, 2007]. Quanto mais perto da Região Impossível, o esforço e o custo do projeto aumentam de maneira exponencial. Assim, a redução de prazo de 10% implica no aumento de custo do esforço de 20%; a redução de prazo de 20% implica no aumento de custo do esforço de 50%; a redução de prazo de 25% implica em um aumento de custo do esforço de 70%. Não será permitida a redução de prazo superior a 25%, dada a Região do Impossível.

## 7.2 Diretrizes para Acompanhamento de Projetos

Esta seção apresenta considerações especiais sobre o gerenciamento de mudança de requisitos, projetos cancelados, progresso de projetos, assim como o tratamento de redução de cronograma e fator criticidade.

### 7.2.1 Considerações sobre Mudança de Requisitos

Em projetos de desenvolvimento e de manutenção de software é bastante observada a mudança de requisitos anterior à implantação do projeto, conforme o usuário e o desenvolvedor adquirem mais conhecimento sobre as necessidades e funcionalidades de negócio [Sommerville, 2007]. O CPM denomina este fenômeno de *Scope Creep*.

Nas estimativas iniciais de tamanho de projetos de desenvolvimento, após a fase de especificação, considerando-se o documento de visão inicial do projeto, será utilizado um percentual de 35% para evolução de requisitos. Por exemplo, suponha que após a análise do documento de visão de um projeto, aplicando-se a CEPF, foi obtido o tamanho de 200 PF, então o tamanho estimado desse projeto é de 270 PF (200 + 35%).

Uma mudança de requisito anterior à implantação do projeto gera retrabalho para a equipe de desenvolvimento, aumentando assim o esforço e o custo do projeto. Neste roteiro, as demandas de mudança de requisitos serão dimensionadas como PF\_RETRABALHO, contadas à parte das funções da demanda de desenvolvimento ou de manutenção, sendo gerenciadas na mesma OS da demanda original.

Cabe ressaltar que para evitar as solicitações de mudança de requisitos devido a falhas na execução da fase de engenharia de requisitos, é importante que seja dada atenção especial à atividade de validação e aceitação dos requisitos.

O método de contagem de mudança de requisitos descrito neste roteiro tem os seguintes pressupostos:

- As demandas de mudança de requisitos são contagens à parte da contagem do projeto de desenvolvimento ou de manutenção e devem considerar as funcionalidades antes da mudança;
- A quantidade de PF\_RETRABALHO apurada leva em conta o esforço já realizado no processo de desenvolvimento da funcionalidade até o momento da solicitação

de mudança de requisitos. É preciso aplicar o percentual das fases do processo de desenvolvimento até o momento da mudança de requisitos na fórmula do cálculo do PF\_RETRABALHO.

- A contagem do projeto de desenvolvimento ou de manutenção deverá ser atualizada a cada demanda de mudança de requisitos, visando refletir as funcionalidades após a mudança.
- Para fins de planejamento ou de faturamento, a quantidade total de pontos de função será obtida da seguinte forma:

$$PF\_TOTAL\_OS = PF\_DEMANDA + \sum PF\_RETRABALHO$$

Onde:

PF\_DEMANDA é a última versão da contagem do escopo da demanda.

A contagem de PF\_RETRABALHO leva em conta as seguintes características:

- Requisito original: é o requisito da demanda de desenvolvimento ou de manutenção original, que pode ser incluir, alterar ou excluir funcionalidades de um sistema.
- Tipo da mudança do requisito: é a natureza da mudança de requisitos da demanda em andamento, que pode ser acrescentar um requisito, alterar um requisito definido ou desistir de um requisito (retirar do escopo do projeto).

Seguem os percentuais que devem ser aplicados sobre as funções alteradas (considerando o tamanho antes da mudança) para obtenção de PF\_RETRABALHO:

Fator		Requisito Original			
		Incluir Função	Alterar Função	Excluir Função	
Tipo da Mudança de Requisito	Acréscimo		-	-	-
	Alteração	Alteração de Requisitos	50%	50%	-
		Alteração de Interface	0,6 PF	0,6 PF	-
	Desistência		130%	80%	30%

Cabe ressaltar que a quantidade de PF\_RETRABALHO obtida, para fins de gestão e faturamento, usa na sua fórmula o percentual das fases concluídas até o momento da solicitação da mudança de requisitos, conforme descrito acima.

A seguir são descritos os tipos de mudança nos projetos.

#### 1. Acréscimo de funcionalidades ao escopo do projeto

As mudanças de requisitos, que não trazem impacto aos requisitos originais da demanda, caracterizadas pelo acréscimo de funcionalidades ao escopo da demanda de desenvolvimento ou de manutenção, serão acrescentadas na contagem de PF do projeto e não geram contagem de PF\_RETRABALHO, ou seja, representam um trabalho adicional e não retrabalho. Enquadram-se nesta situação a inclusão, a alteração ou a exclusão de funções que não constavam no escopo da demanda original.

#### 2. Alteração de função.

A contagem de PF\_RETRABALHO referente à alteração deve considerar o percentual de 50% sobre o tamanho da função antes da alteração, independentemente do requisito original. Este item se refere somente à alteração de requisitos de funcionalidades que estavam sendo criadas ou alteradas no projeto original (Caso 1). Em caso de mudanças em interface (cosméticas), considerar 0,6 PF, independentemente da complexidade da função antes da alteração (Caso 2).

Sobre a quantidade de PF\_RETRABALHO obtida, para fins de gestão e faturamento, deverá ser aplicado o percentual das fases concluídas até o momento da solicitação de mudança de requisitos.

A contagem de PF da demanda deve ser atualizada para refletir o novo grau de complexidade da função após a mudança.

Exemplo:

Considerando-se que uma demanda de manutenção evolutiva (melhoria) tinha como escopo a alteração de uma EE (complexidade alta - 6 PF), desenvolvida pela empresa contratada, a criação de uma CE (complexidade baixa - 3 PF) e a criação de uma SE (complexidade baixa - 4 PF). A contagem de PF\_MELHORIA é:

- Inclusão de CE e SE:  $3 \text{ PF} + 4 \text{ PF} = 7 \text{ PF}$
- Alteração de EE:  $6 \text{ PF} * 50\% = 3 \text{ PF}$
- PF\_MELHORIA v1 = 10 PF

Caso 1: Alteração de requisitos

No início da homologação foram solicitadas mudanças nos requisitos da EE e da CE, sendo que a complexidade da CE passou a ser média (4 PF) após a mudança. Nesta situação hipotética, a contagem de PF\_RETRABALHO será a seguinte:

- EE original: 6 PF
- CE original: 3 PF
- $PF\_RETRABALHO = (6\text{ PF} + 3\text{ PF}) \times 50\% \times 90\%$
- $PF\_RETRABALHO = 4,05\text{ PF}$

Sendo 50% o percentual de alteração de requisitos a ser aplicado sobre o tamanho da função original antes da sua alteração. E 90% é o fator fase, conforme Tabela 8 deste Roteiro. São consideradas todas as fases concluídas até a homologação, visto que a solicitação de alteração de requisitos ocorreu no início da fase de homologação.

A contagem de PF\_MELHORIA deverá ser atualizada para refletir o aumento da complexidade da CE alterada:

- Inclusão de CE alterada e SE:  $4\text{ PF} + 4\text{ PF} = 8\text{ PF}$
- Alteração de EE alterada:  $6\text{ PF} * 50\% = 3\text{ PF}$
- PF\_MELHORIA v2: 11 PF Caso 2: Alteração de interface No início da fase da fase de homologação foi solicitada uma alteração na função SE, que é um relatório. A demanda é para alterar o tipo de fonte do título do relatório (alteração de interface - cosmética). A complexidade da função SE se mantém a mesma (complexidade baixa - 4 PF) após a mudança. Nesta situação hipotética, a contagem de PF\_RETRABALHO será a seguinte:

- SE original: 4 PF
- $PF\_RETRABALHO = 0,6\text{ PF}$
- $PF\_RETRABALHO = 0,54\text{ PF}$

Sendo 0,6 PF a contagem da alteração cosmética de requisitos. E 90% é o fator fase, conforme Tabela 8 deste Roteiro. São consideradas todas as fases concluídas até a homologação, visto que a solicitação de alteração de requisitos ocorreu no início da fase de homologação.

Nesse caso de mudança de requisitos com alteração de interface (cosmética), a contagem de PF\_MELHORIA não sofre alteração, visto que a complexidade da função SE não foi alterada.

### 3. Desistência de incluir, alterar ou excluir uma função

Em caso de desistência de incluir, alterar ou excluir uma função, deve-se verificar qual era o requisito original, pois o percentual a ser utilizado na contagem de PF\_RETRABALHO varia para cada situação. Além do trabalho de retirar o que foi requisitado, deve-se considerar também em PF\_RETRABALHO, o trabalho realizado (fases concluídas do processo de desenvolvimento) até o momento da desistência desse requisito. Por fim, o requisito original deve ser removido do PF\_MELHORIA. Enquadram-se nesta situação somente as desistências de incluir, de alterar ou de excluir funcionalidades que constavam no escopo da demanda.

Quando a mudança na demanda for deixar de incluir uma função, aplica-se o percentual de 130% ao tamanho da função original. Esse valor é resultado da soma do percentual de 100% da inclusão (escopo original) com os 30% correspondentes à exclusão dessa mesma função.

Quando a mudança no projeto for deixar de alterar uma função, aplica-se o percentual de 80% ao tamanho da função original. Esse valor é o resultado da soma do percentual de 50% da alteração (escopo original) com os 30% referentes à exclusão dessa mesma função.

Quando a mudança no projeto for deixar de excluir uma função, aplica-se apenas o percentual de 30% referente à exclusão da função original.

Em todos os casos, a contagem de PF\_MELHORIA deve ser atualizada removendo-se as funções que não fazem mais parte do escopo da demanda.

Da mesma forma que no item 2 (Alteração de função), para fins de gestão e faturamento, sobre a quantidade de PF\_RETRABALHO é aplicado o percentual das fases concluídas até o momento da solicitação de mudança de requisitos.

Exemplos:

Desistência de incluir função

Suponha que em uma demanda de manutenção para a criação do relatório XPTO, contado como uma SE de complexidade média com 5 PF, teve uma demanda de exclusão do relatório XPTO no início da fase de implementação (ou seja, o relatório não será mais construído). Desta forma a contagem de PF\_RETRABALHO será a seguinte:

- SE original: 5 PF
- $PF\_RETRABALHO = 5\text{ PF} \times 130\% = 6,5\text{ PF} \times 35\%$
- $PF\_RETRABALHO = 2,275\text{ PF}$

Sendo que 130% é o percentual a ser aplicado sobre o tamanho da função original antes da desistência da sua inclusão. E 35% é o fator fase, conforme Tabela 8 deste Roteiro. São consideradas todas as fases concluídas até a implementação (requisitos e design/arquitetura), visto que a solicitação de alteração de requisitos ocorreu no início da fase de implementação.

A contagem de PF\_MELHORIA deve ser atualizada para que o relatório XPTO deixe de constar na medição.

Desistência de alterar função

Se, no exemplo anterior, o relatório XPTO estivesse sendo originalmente alterado (ao invés de incluído), a diferença seria no percentual aplicado em PF\_RETRABALHO:

- SE original: 5 PF
- $PF\_RETRABALHO = 5 \text{ PF} \times 80\% \times 35\%$
- $PF\_RETRABALHO = 1,4 \text{ PF}$

Sendo que 80% é o percentual a ser aplicado sobre o tamanho da função original antes da desistência da sua inclusão. E 35% é o fator fase, conforme Tabela 8 deste Roteiro. São consideradas todas as fases concluídas até a implementação (requisitos e design/arquitetura), visto que a solicitação de alteração de requisitos ocorreu no início da fase de implementação.

A contagem de PF\_MELHORIA deve ser atualizada para que o relatório XPTO deixe de constar na medição.

#### 4. Desistencia de alterar uma função seguida de exclusão da função

Quando a solicitação de mudança seja não só deixar de fazer o que estava previsto na demanda original, mas também excluir a função da aplicação, deve-se considerar esses dois aspectos separadamente, como se fossem duas mudanças consecutivas:

- A) Conta-se a desistência de alterar a função conforme descrito no item 3 (Desistência de incluir, alterar ou excluir uma função), apurando a quantidade de PF\_RETRABALHO correspondente e a atualização do PF\_MELHORIA;
- B) Conta-se o acréscimo ao escopo do projeto (excluir a função da aplicação) conforme descrito no item 1 (Acréscimo ao escopo), atualizando-se PF\_MELHORIA.

## 7.2.2 Considerações sobre Projetos Cancelados

Em alguns casos, devido a 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 à construção da funcionalidade em questão, observando a Tabela 7 de distribuição de esforço contida na subseção 6.1.2, considerando apenas o esforço das fases concluídas.

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, contudo, apenas serão consideradas na definição do percentual de esforço as fases de fato concluídas.

Segue um exemplo de aferição do tamanho funcional de um projeto cancelado:

Requisito	Tamanho	Engenharia de Requisitos	Design, Arquitetura	Implementação	Testes	Homologação	Implantação
Caso de Uso Relatório de Projetos	5 PF	100%	100%	50%	20%	0%	0%

O projeto encontrava-se com a engenharia de requisitos e design/arquitetura concluídos, a implementação estava na metade da execução e os testes estavam em 20% realizados.

Supondo o cancelamento do projeto por iniciativa da contratante, a contagem de PF a ser apurada é a seguinte:

Caso de Uso – Relatório de Projetos – 5 PF

Macroatividade	Esforço da Fase	Tamanho	Esforço realizado	Tamanho realizado
Engenharia de Requisitos	25%	1,25 PF	<b>100%</b>	<b>1,25 PF</b>
Design, Arquitetura	10%	0,5 PF	<b>100%</b>	<b>0,5 PF</b>
Implementação	40%	2 PF	50%	0PF
Testes	15%	0,75 PF	20%	0PF
Homologação	5%	0,25 PF	0%	0 PF
Implantação	5%	0,25 PF	0%	0 PF
			<b>Total:</b>	<b>1,75 PF</b>

O tamanho realizado é de 1,75 PF, pois conforme definido neste roteiro, são consideradas apenas as fases concluídas.

### 7.2.3 Considerações sobre Redução de Cronograma

Alguns projetos, devido à legislação e a outros fatores externos, já se iniciam com um prazo imposto.

No entanto, se o projeto tiver um prazo imposto inferior ao prazo definido, conforme apresentado na subseção 6.1.3, então o requisitante deverá registrar a motivação para a abertura de OS com prazo reduzido. Os prazos reduzidos acarretam aumento de custos, conforme a definição da tabela 10:

Redução de prazo	Aumento de custo (PF)
1%	2%
2%	4%
3%	6%
4%	8%
5%	10%
6%	12%
7%	14%
8%	16%
9%	18%
10%	20%
11%	23%
12%	26%
13%	29%
14%	32%
15%	35%
16%	38%

17%	41%
18%	44%
19%	47%
20%	50%
21%	55%
22%	60%
23%	65%
24%	70%
25%	75%

**Tabela 10: Percentuais de redução de prazo e aumento de custo.**

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 processo incremental.

Não é recomendado o uso de redução de cronograma, sugere-se trabalhar com definição de prioridades.

## 5. Contagem de Pontos de Função de Projetos Ágeis

Este capítulo descreve orientações sobre o uso da métrica Ponto de Função na medição de projetos de desenvolvimento e de manutenção de software com métodos ágeis, a fim de realizar estimativas e dimensionamento de tamanho funcional de projetos de software que utilizam métodos ágeis.

Como características inerentes ao processo ágil, entende-se que as mudanças em funcionalidades são mais constantes e recorrentes nesse cenário de desenvolvimento de software, pois se pressupõe um escopo mais aberto. Entretanto, o processo ágil de desenvolvimento de software não deve comprometer o princípio de efetividade dos resultados previstos e entregues com a garantia da exequibilidade do projeto.

É importante observar que, conforme a Súmula TCU 269, a remuneração nas contratações de serviços de Tecnologia da Informação deve estar vinculada à entrega de resultados ou ao atendimento de níveis de serviço.

Os objetivos e premissas considerados neste trabalho foram:

- Buscar a simplicidade na medição do desenvolvimento de software com métodos ágeis para viabilizar o seu uso com responsabilidade e garantir o alcance dos benefícios do processo ágil ao negócio.
- Simplificar o ônus de gestão e controle de mudanças de forma a minimizar impactos sobre a agilidade do processo de desenvolvimento de software.
- Prever e medir o esforço e o volume de mudanças em funcionalidades em um projeto de software com desenvolvimento ágil.
- Apoiar o uso de desenvolvimento ágil nos projetos de software.
- Propor índices de produtividade coerentes com o processo de desenvolvimento ágil utilizando a base histórica de projetos do Serpro.

Nesse sentido, são apresentadas orientações de medição em Ponto de Função para o tratamento das mudanças em funcionalidades durante o projeto.

## 8.1. Conceitos

No cenário de desenvolvimento de software com métodos ágeis e dentro do contexto deste roteiro, é importante alinhar os seguintes conceitos:

**Release:** É um ciclo que perpassa pelas fases do processo de desenvolvimento de software com o objetivo de entregar, ao final do ciclo, um produto pronto a ser colocado em produção para uso. A duração de cada release será definida pelo Gestor de Negócios e PO em acordo com o time de desenvolvimento na fase de planejamento do projeto conforme seu backlog priorizado e de forma a garantir uma entrega de valor antecipada aos usuários. Recomenda-se que as releases sejam curtas com até 4 sprints, pois ao praticar as "releases curtas" o retorno do investimento é acelerado e as evoluções do produto podem ser guiadas pelo feedback dos seus usuários.

**Sprint:** É uma unidade de período de tempo fixo (time box) dentro da release, com datas de início e fim pré-definidas, dentro da qual é executado um conjunto de atividades de desenvolvimento do projeto previamente estabelecidas, gerando ao final um incremento do produto aceito e potencialmente implantável. Recomenda-se que a duração da sprint seja de 2 a 4 semanas.

**Produto Pronto:** entende-se que um produto está "pronto" se foi entregue e validado pelo cliente. Cabe observar que o desenvolvimento de uma funcionalidade pode perpassar mais de uma sprint e conter várias histórias de usuários prontas e validadas em sprints diferentes; nesse caso, a funcionalidade será considerada pronta ao término da release.

**Product Owner (PO):** Membro do Time Ágil designado pelo cliente como responsável pelo produto, com conhecimento do negócio e autonomia para decidir que resultados devem ser gerados

visando maximizar o valor do produto e do trabalho do time. Deve estar disponível para interação contínua, responder dúvidas de forma tempestiva e manter um diálogo sincero e transparente com os demais membros do Time Ágil.

## 8.2. Orientações

O desenvolvimento de software utilizando métodos ágeis deve respeitar uma abordagem específica que considere as características dos métodos ágeis, tanto no desenvolvimento quanto na gestão do projeto. Entretanto, essas características podem requerer adaptações para o contexto de contratação de software em pontos de função, no sentido de atender o cumprimento da legislação e dos princípios de economicidade e eficiência.

Nesse cenário, algumas considerações e sugestões são propostas:

- Divisão do projeto de desenvolvimento ou manutenção em releases.
- Releases curtas com até 4 sprints.
- Ciclo da release sempre deve promover o aumento do percentual de completude do sistema (entrega de valor agregado ao negócio).
- Ciclo da sprint de duas até quatro semanas.
- Para as funcionalidades que precisem de mais de uma sprint para serem desenvolvidas, devem ser consideradas para efeito de contagem de Pontos de Função somente na sprint em que forem entregues e aceitas, ou seja, validadas pelo cliente no evento de validação da sprint.
- Realizar a contagem estimada ou estimativa de Pontos de Função do projeto e das releases, para apoiar o planejamento do projeto e geração das estimativas de prazo e esforço.

Caso o projeto seja faturado por sprint, sugere-se dividir o valor da estimativa inicial da release, sem scope creep, pela quantidade de sprints da release, visando a obtenção do tamanho estimado da sprint para o faturamento. Essa sugestão visa não onerar as partes envolvidas com a medição, controle e gestão de mudanças a cada sprint, principalmente, considerando que a duração dessas sprints são muito curtas.

Este roteiro adota a abordagem de que as contagens de início e fim da release sejam consideradas como marcos de controle para apuração de Pontos de Função. As estimativas/contagens de PF iniciais e finais dos projetos e das releases são obrigatórias. Caso ocorra replanejamento do projeto/release, torna-se necessário atualizar as estimativas iniciais do projeto e da release. Caso ocorra o cancelamento do projeto ou de uma release, deverá ser realizada contagem de PF conforme item “7.5 Cancelamento de Projetos Ágeis”.

### 8.3. Mudanças de Requisitos em Funcionalidades no Processo Ágil

Esta seção apresenta alguns fatores que influenciam o quantitativo de mudanças de requisitos em funcionalidades nos projetos de software com métodos ágeis e orientações para tratamento destas mudanças usando a métrica Ponto de Função.

Alguns fatores devem ser considerados e avaliados para a estimativa do volume de mudanças em funcionalidades em projetos de desenvolvimento e de manutenção de software com métodos ágeis:

- maturidade dos requisitos do projeto;
- conhecimento do negócio pelo product owner;
- disponibilidade e experiência do cliente com o uso dos métodos ágeis;
- tamanho da sprint;
- tamanho da release; • volume de mudanças em funcionalidades de projetos similares já executados.

Considerando os aspectos do desenvolvimento ágil, este roteiro propõe que:

- Mudanças em funcionalidades que ocorrerem após o término da release em que essas funcionalidades ficaram prontas (mudanças em releases diferentes), devem ser tratadas de acordo com o item Projeto de Melhoria deste Roteiro, uma vez que este guia considera que, no desenvolvimento de software com métodos ágeis, o ciclo de trabalho evolutivo em funcionalidades desenvolvidas em uma release encerra-se ao final da release. A contagem de encerramento da release deve conter todas as funcionalidades que foram entregues com sua complexidade final.
- Mudanças em funcionalidades ocorridas dentro da mesma release não sejam contadas. Porém, essas mudanças podem ser registradas com a finalidade de compor um histórico sobre retrabalho e esforço empregado em seu atendimento.

### 8.4. Recomendações para o Retrabalho em Projetos Ágeis

Cada mudança de requisitos ao longo do ciclo de vida do projeto é considerada retrabalho e os Pontos de Função correspondentes são adicionados ao tamanho funcional atendido no projeto. Como as mudanças são inerentes ao processo ágil, recomenda-se aplicar um fator de impacto para absorção do esforço de construção do retrabalho, denominado Fator Ágil.

Em uma análise do retrabalho dos projetos ágeis do Serpro, percebeu-se que o volume de mudanças nos requisitos aumenta significativamente conforme aumenta a quantidade de sprints de uma release. Desta forma, seguem as recomendações:

A. Utilizar um Fator Ágil, visando não realizar as contagens de alteração de requisitos (retrabalho) em sprints dentro da release, que oneram a gestão da demanda pelo cliente. B. O Fator Ágil deve ser acrescido em 8% a cada sprint, a partir da Sprint 2.

Cabe ressaltar que as desistências de funcionalidades dentro da Release devem ser contadas. O Fator Ágil contempla apenas as alterações de requisitos que ocorrem em uma Release específica, bastante comum em projetos ágeis.

## 8.5. Cancelamento de Projetos Ágeis

A contagem de PF de um projeto ágil cancelado deve levar em consideração as contagens de PF das Releases e/ou Sprints concluídas e a contagem da Sprint em andamento. No caso da Sprint em andamento, deve-se aplicar um percentual de Fase sobre as funcionalidades da Sprint considerando a fase do processo de desenvolvimento em que se encontra a Sprint em questão.

## 6. Atividades Sem Contagem de Pontos de Função

Deve-se ressaltar que, no processo de desenvolvimento de um projeto de software, há atividades que devem ser consideradas como complementares ou pré-requisitos ao processo de desenvolvimento, de modo que os esforços e produtos entregues devem ser contratados e remunerados em itens distintos do desenvolvimento por não se tratarem de atividades de desenvolvimento do software ou inerentes ao processo desenvolvimento do software. São atividades categorizadas nessa condição:

- Definição de Processo de Desenvolvimento de Soluções: são as demandas para definição de Processos de Software, aderentes às melhores práticas do CMMI e à Instrução Normativa SLTI n° 4, de 12 de novembro de 2010 que devem estar definidos antes da contratação de serviços de desenvolvimento de software.
- Desenvolvimento de Cursos para EaD: são as demandas de elaboração de conteúdo e montagem de material para um curso na modalidade de Ensino a Distância (EaD).
- Mapeamento de Processos de Negócio: são as demandas de elaboração de documentação contendo o mapeamento de processos de negócio de uma organização ou de parte de uma organização. Essa é uma atividade que deve ser realizada antes da abertura do projeto de desenvolvimento de software. É importante ressaltar que essa atividade é de responsabilidade dos analistas de negócio da empresa contratante, de acordo com a Instrução Normativa SLTI n° 4, de 12 de novembro de 2010. No entanto, por falta de pessoal,

alguns órgãos e entidades têm contratado estas atividades, que antecedem a fase de requisitos – primeira fase do processo de software.

- **Análise de Viabilidade:** Serviço de apoio destinado à análise de arquitetura, processos e regras de negócio implementados em soluções de TI.
- **Soluções de Design:** Criação de Marcas; Definição de Arquitetura de Informação para sítios e portais; Criação de Interfaces Visuais especializadas para sítios, sistemas e aplicativos; Criação de Manuais de Interface para sistemas; Criação de Bibliotecas de Elementos Visuais para frameworks; Consultorias de Identidade, Usabilidade e, Acessibilidade.
- **Demandas de Desenvolvimento ou Manutenção de Componentes Arquiteturais:** Refere-se ao desenvolvimento ou manutenção de componente arquitetural para atender requisitos não funcionais, sem funcionalidade associada, considerando inclusive a preparação e execução de testes não funcionais associados. Como exemplo, temos desenvolvimento/manutenção de Componentes webservices não funcionais que possuem como função a integração de dois sistemas.
- **Treinamentos em Tecnologia da Informação:** são as demandas de treinamentos em linguagens de programação, ferramentas de gestão, processos, modelos da qualidade, métricas, etc.
- **Elaboração de Plano Diretor de Tecnologia da Informação (PDTI):** são as demandas para elaboração de PDTI para clientes.
- **Consultoria:** São serviços especializados e tratados em contratações em separado, como consultoria em métricas e consultoria para apoio na contratação de soluções de TI.

Outras atividades contidas em um processo de software devem ser gerenciadas dentro do projeto de desenvolvimento e são inerentes ao processo de desenvolvimento de software, não devendo ser mensuradas separadamente. São elas:

- **Acompanhamento de Projetos:** é a atividade que a contratada faz internamente de modo a se organizar e planejar o atendimento dos cronogramas e outras demandas recebidas da contratante, cuja natureza é intrínseca ao desenvolvimento e manutenção de sistemas. Ou seja, ao desenvolver e manter sistemas, a tarefa de acompanhar e gerir o projeto por parte da contratada figuram como seus deveres contratuais, não cabendo pagamento por atividades que dizem respeito à sua própria gestão interna;
- **Correção de erros:** erros e bugs que venham a se manifestar em ambiente de produção dentro do período de garantia contratado.

- Especificação de Requisitos: em metodologias ágeis, o levantamento de requisitos é inerente ao processo de desenvolvimento de software, não devendo ser mensurado e remunerado separadamente. Em outras metodologias, caso o órgão opte por realizar o levantamento de requisitos separadamente do processo de desenvolvimento de software, esse deve ser remunerado por horas de consultoria.
- Projeto e desenvolvimento de Banco de Dados: as atividades de banco de dados associadas ao projeto de desenvolvimento, modelagem dos bancos seguindo as políticas de dados da contratante, preparação de ambiente (testes, homologação, implantação), desempenhadas pela contratada já devem ser consideradas dentro do projeto de software, não cabendo cobrança adicional.
- Treinamento para Implantação: são demandas de treinamentos sobre utilização do sistema desenvolvido pela contratada a ser implantado, para os gestores de solução do cliente e usuários e devem ser tratadas no escopo da fase de transferência do conhecimento para a contratante.
- Atividades cuja a natureza difere totalmente do objeto contratado (serviços de desenvolvimento de software) não podem ser remuneradas por pontos por função, são exemplos:
  - Deslocamentos e viagens de integrantes da contratada para prestação dos serviços em diferentes localidades;
  - Suporte ao Usuário e à Rede no uso do software desenvolvido, principalmente quando englobando atividades como instalação de microcomputadores e demais periféricos.

Essa relação não é taxativa, uma vez que existem outras atividades que precisam ser consideradas, mas que não serão aferidas pela métrica de Pontos de Função.

## **7. Processo de Revisão do Roteiro de Contagem**

### **a. Revisão para Correção de Inconsistências e Situações não Previstas**

A revisão deste roteiro será feita sempre que se verificarem inconsistências entre uma definição do CPM e uma regra constante deste documento, e situações não previstas neste roteiro. Essas situações, sempre que necessário, serão documentadas, gerando novas versões deste roteiro.

## **b. Revisão para Adoção de Novas Versões do CPM e do Roteiro de Métricas de Software do SISP**

A adoção de nova versão do CPM ou do Roteiro de Métricas de Software do SISP como referência para este roteiro de contagem não será imediata às suas publicações. Nesse caso, haverá uma avaliação das novas versões para se decidir sobre a atualização deste documento. A atualização do roteiro para utilização em contrato deve ser negociada com a empresa contratada.

## **8. Dicas para otimizar o custo das manutenções**

É importante que se tenha uma atenção especial na gestão de demandas de manutenção de sistemas. A seguir algumas dicas que podem ajudar a melhorar esse cenário:

**Consolidar Manutenções na mesma funcionalidade em uma única demanda:** fazer uma manutenção para atender um único requisito ou para atender a vários requisitos na mesma função terá o mesmo tamanho funcional se elas forem solicitadas na mesma ordem de serviço. Se solicitadas em momentos distintos, as mesmas funções serão pagas várias vezes, para cada um dos projetos de melhoria/manutenção.

**Análise crítica dos requisitos:** em muitas situações é possível ter uma única função que faça o papel de duas ou mais funções existentes. É muito comum existirem consultas e relatórios com objetivo semelhante mas que apresentam uma diferença mínima entre os campos (atributos) apresentados. Nesses casos, é importante analisar criticamente os requisitos e, se possível, optar por uma transação mais completa e evitar a criação de funções semelhantes. É mais barato pagar pela criação de uma função nova, do que por uma manutenção em duas ou mais funcionalidades, principalmente a longo prazo.

## **9. Conclusão**

Este documento apresentou um roteiro para o dimensionamento de tamanho de vários tipos de projetos de software, visando a aderência desses tipos de projetos desenvolvidos na instituição às diretrizes da Instrução Normativa SLTI n° 4, de 11 de Setembro de 2014. A estimativa de tamanho utiliza a métrica Ponto de Função Não Ajustado como unidade de medida, conforme recomendado nos Acórdãos 1.910/2007, 2.348/2009 e 1.647/2010 do Tribunal de Contas da União (TCU) e na Portaria SLTI/MP N° 31, de 29 novembro de 2010.

É importante ressaltar que o uso de métricas em contrato de software é uma boa 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.

Como trabalho futuro, recomenda-se a revisão e atualização deste roteiro sempre que for verificada inconsistência entre alguma definição do IFPUG 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 roteiro.

## 10. Referências Bibliográficas

- [FNDE.2010] FNDE. Guia de Contagem do FNDE, 2010.
- [SISP 2.2] SISP. Roteiro de Métricas de Software do SISP. Versão 2.2, 2016.
- [IFPUG, 2010] IFPUG. *Counting Practices Manual*. Version 4.3.1, January, 2010.
- [IFPUG,2009] IFPUG. Considerations for Counting with Multiple Media. Release 1.0, September, 2009.
- [IFPUG, 2013] White Paper Applying Function Point Analysis to Scrum Agile Software Development
- [IFPUG, 2007] Projects, versão 1.0.
- IFPUG. International Function Point Users Group. Function Points & Counting Enterprise Data Warehouses, Release 1.0. NEC White Paper. [S.I.]: IFPUG. 2007.
- [JONES, 2007] JONES, C. Estimating Software Costs. Second Edition, Mc Graw Hill, 2007.
- [NESMA, 2014] NESMA. Netherlands Software Metrics Association. Definitions and counting guidelines for the application of function point analysis - NES MA functional size measurement method conform ISO/IEC 24570, version 2.1. [S.I.]: NESMA. 2004.
- [KIMBALL e KIMBALL, Ralph; ROSS, Margy. The Data Warehouse Toolkit. 2ª. Rio de Janeiro: Ed. ROSS, 2002] Campus. 2002.
- [TCU, 2013] TRIBUNAL DE CONTAS DA UNIÃO. Levantamento de auditoria. Conhecimento acerca da utilização de métodos ágeis nas contratações para desenvolvimento de software pela Administração Pública. TC 010.663/2013-4. Relator: José Múcio Monteiro. Brasília, 2013.

# Orientações para preenchimento da Planilha de Contagem

A planilha está dividida em três partes: Identificação, Contagem e Fundamentação (Anexos I, II, III).

Cada aba possui campos específicos que devem ser preenchidos adequadamente, conforme descrito nos anexos a seguir.

Observe nos anexos I e II que os campos de preenchimento automático (fundo cinza claro) não devem ser alterados, uma vez que seus valores são carregados com base em outros campos da planilha.

## ANEXO 1 – Identificação da Planilha de Contagem de Ponto de Função

			
<b>Planilha de Contagem de Ponto de Função</b> <b>Versão 1.0</b>			
<b>Identificação da Contagem</b>			
<b>Empresa:</b>		<b>R\$/PF:</b>	
<b>Nome do Sistema:</b>		<b>PF:</b>	
<b>Nº da OS:</b>		<b>Custo:</b>	R\$
<b>Elaborador:</b>		<b>Data da Contagem:</b>	DD/MM/AAAA
<b>Revisor:</b>		<b>Data da Revisão:</b>	DD/MM/AAAA
<b>Tipo de Contagem</b>	<input type="checkbox"/> Estimada	<b>Tipo de Atividade:</b>	<input type="checkbox"/> Manutenção
	<input type="checkbox"/> Detalhada		<input type="checkbox"/> Desenvolvimento
<b>Histórico de Versão</b>			
<b>Versão</b>	<b>Data:</b>	<b>Descrição:</b>	<b>Responsável:</b>
<b>Propósito da Contagem</b>			
<b>Escopo da Contagem</b>			

A primeira parte é a identificação da contagem. Nela são apresentadas as informações sobre a contagem, empresa, número da OS, o propósito da contagem, o tipo de contagem, etc.

## **ANEXO 2 – Planilha de Contagem de Ponto de Função**

 <p style="text-align: center;"><b>Planilha de Contagem de Ponto de Função Versão 1.0</b></p>										
<b>Empresa:</b>							<b>PF:</b>			
<b>Nome do Sistema:</b>							<b>Nº da OS:</b>			
<b>Elaborador:</b>							<b>Revisor:</b>			
<b>Tipo de Projeto</b>	<b>Fator de Ajuste</b>	<b>Função</b>	<b>Tipo</b>	<b>DER</b>	<b>ALR/RL</b>	<b>Compl.</b>	<b>PF</b>	<b>Fund.</b>	<b>OBS</b>	

A segunda parte contém todos os itens necessários para a medição de pontos de função:

**Campos automáticos:** Empresa, PF, Nome do Sistema, Nº da OS, Elaborador e Revisor.

**Tipo do Projeto:** É o fato gerador para uma determinada função ter entrado no escopo da contagem.  
Ex: Desenvolvimento, Manutenção de Interface, Verificação de Erros, etc.

**Fator de Ajuste:** Fator referente ao Tipo do Projeto. Exemplo: 0,20 é o fator de ajuste para Verificação de erros.

**Função:** Preencha com o nome da função, seja ela uma função de dados ou uma função de transação. Se for função de dados, deve-se utilizar um substantivo ou grupo de palavras que identifiquem o grupamento de dados (ex: Pessoa, Remediador, Empreendimento, Unidade Organizacional, Processo), e não utilizar o nome físico da tabela associada. Se for função transacional, iniciar o nome da função com verbo significativo, ex: Incluir funcionário, Listar

Empreendimentos, Exportar Dados do Relatório de Transferências, Visualizar Requerimento, Consultar Processo, Editar Documento, Visualizar dados do Documento (consulta implícita).

**Tipo:** É o tipo de função. Preencha com as siglas abaixo:

1. Entrada Externa – EE
2. Saída Externa – SE
3. Consulta Externa – CE
4. Arquivo Lógico Interno – ALI
5. Arquivo de Interface Externa – AIE

**DER:** Quantidade de DER necessária na determinação da complexidade de uma função de dados. A relação textual de DERs da função deve ser detalhada em caixas de comentário flutuante nas células correspondentes, ou em outro local que se avalie apropriado, mantendo-se a legibilidade e correspondência com o registro.

**ALR/RL:** Quantidade de Arquivos Lógicos Referenciados ou quantidades de registros lógicos. A relação textual de ALR/RL da função deve ser detalhada em caixas de comentário flutuante nas células correspondentes, ou em outro local que se avalie apropriado, mantendo-se a legibilidade e correspondência com o registro.

**Complexidade:** Calculado pela planilha de acordo com a quantidade de DER e ALR/RL e conforme o Tipo de Função.

**PF:** Contribuição dos Pontos de Função da medição da respectiva função. Também calculado pela planilha.

**Fundamentação:** número da tela de fundamentação (Anexo III).

**Observação:** campo opcional para comentários adicionais. Utilize este espaço para documentar qual o insumo que teve para identificar a função relativa àquela linha. Qual caso de uso, qual relatório, enfim, algo que permita a você justificar a contagem com agilidade.

## ANEXO 3 – Fundamentação da Contagem de Ponto de Função

		
<b>Planilha de Contagem de Ponto de Função FUNDAMENTAÇÃO</b>		
<b>Fundamentação da Contagem</b>		
ID	Título	Imagem tela

Na terceira parte são anexadas as telas de fundamentação para cada função transacional do anexo II:

**ID:** Número da Tela;

**Título:** Título da Tela;

**Imagem tela:** imagem da tela correspondente a função transacional. A imagem pode ter o formato jpg, png ou gif.

Exemplo para a aba de fundamentação: