

PRT - Assessoria de
Tecnologia da Informação

DIRETRIZES PARA MEDIÇÃO EM PONTOS DE FUNÇÃO

Versão 1.1



Presidente da CAESB

Maurício Leite Ludovice

Assessora de Tecnologia da Informação

Márcia Sabino Duarte

Gerente de Desenvolvimento e Manutenção de Sistemas

Bernardo Vergne Dias

PRT - Assessoria de
Tecnologia da Informação

DIRETRIZES PARA MEDIÇÃO EM PONTOS DE FUNÇÃO

Versão 1.1



Companhia de Saneamento
Ambiental do Distrito Federal

Brasília-DF, abril de 2017

*“Nós somos o que fazemos todos os dias.
Deste modo, a excelência não é um ato,
mas um hábito” – Aristóteles*

SUMÁRIO

1	INTRODUÇÃO	1
2	MUDANÇAS NAS VERSÕES 1.x	1
2.1	VERSÃO 1.0.....	1
2.2	NOVIDADES DA VERSÃO 1.1.....	2
3	FUNDAMENTAÇÃO LEGAL	2
4	CONTAGEM DE PONTOS DE FUNÇÃO	3
5	DEFINIÇÃO DOS SERVIÇOS.....	6
6	MEDIÇÃO DOS SERVIÇOS	8
6.1	DESENVOLVIMENTO (S01).....	9
6.2	MELHORIA (S02).....	9
6.3	MIGRAÇÃO DE DADOS (S03).....	11
6.4	MANUTENÇÃO CORRETIVA (S04).....	11
6.5	MUDANÇA DE PLATAFORMA - LINGUAGEM DE PROGRAMAÇÃO (S05).....	12
6.6	MUDANÇA DE PLATAFORMA - BANCO DE DADOS (S06)	13
6.7	ATUALIZAÇÃO DE VERSÃO - LINGUAGEM DE PROGRAMAÇÃO (S07)	14
6.8	ATUALIZAÇÃO DE VERSÃO - BROWSER (S08)	14
6.9	ATUALIZAÇÃO DE VERSÃO - BANCO DE DADOS (S09).....	15
6.10	MANUTENÇÃO EM INTERFACE (S10).....	15
6.11	ADAPTAÇÃO EM REQUISITOS NÃO FUNCIONAIS (S11)	16
6.12	APURAÇÃO ESPECIAL - GERAÇÃO DE RELATÓRIO (S12)	17
6.13	APURAÇÃO ESPECIAL - REEXECUÇÃO (S13)	18
6.14	ATUALIZAÇÃO DE DADOS (S14).....	18
6.15	MANUTENÇÃO DE PÁGINA ESTÁTICA (S15).....	19
6.16	MANUTENÇÃO DE DOCUMENTAÇÃO (S16).....	20
6.17	VERIFICAÇÃO DE ERRO (S17).....	20
6.18	TESTE (S18).....	21
6.19	MANUTENÇÃO DE COMPONENTE INTERNO REUSÁVEL (S19)	21
6.20	SERVIÇO EM GARANTIA (S20)	22
7	DISTRIBUIÇÃO DO ESFORÇO	22

8	PLANEJAMENTO E ACOMPANHAMENTO	23
8.1	ESTIMATIVA DE TAMANHO	24
8.2	ESTIMATIVA DE ESFORÇO.....	26
8.3	ESTIMATIVA DE PRAZO.....	26
8.4	ESTIMATIVA DE EQUIPE.....	28
8.5	GERENCIAMENTO DE PROGRESSO	28
8.6	MUDANÇA DE REQUISITOS.....	29
9	CONTAGEM COM MÚLTIPLAS MÍDIAS.....	36
10	CONSIDERAÇÕES SOBRE DADOS DE CÓDIGO	38
11	ORDENS DE SERVIÇO COMPOSTAS.....	39
12	REFERÊNCIAS BIBLIOGRÁFICAS	41

1 INTRODUÇÃO

O presente documento **define as diretrizes para mensuração dos serviços de desenvolvimento e manutenção de sistemas em regime de Fábrica de Software com base na Análise de Pontos de Função (APF) no âmbito da PRT / CAESB.**

Este documento é baseado no Manual de Práticas de Contagem (CPM - *Counting Practice Manual*) versão 4.3 do IFPUG (*International Function Point Users Group*) [1], no Roteiro de Métricas do SISP (Sistema de Administração dos Recursos de Tecnologia da Informação) versão 2.1 do MPOG (Ministério do Planejamento, Orçamento e Gestão) [2] e na experiência da CAESB na gestão de projetos de desenvolvimento e manutenção de sistemas.

O Manual de Práticas de Contagem (CPM 4.3) 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. No entanto, os projetos de software não estão limitados a projetos de desenvolvimento e de melhoria. Desta forma, torna-se essencial a definição de métricas para dimensionar o tamanho de outros tipos de projetos de manutenção, os quais são itens não mensuráveis pelo CPM.

O presente documento visa suportar contratos de prestação de serviços de Fábrica de Software, contemplando questões vivenciadas pela CAESB. Busca-se, melhor aproximar a contagem de Pontos de Função do CPM ao esforço técnico relacionado ao ciclo do desenvolvimento e manutenção de sistemas.

O restante do documento encontra-se organizado da seguinte forma: o [capítulo 2](#) traz a descrição do histórico evolutivo deste guia de contagem; o [capítulo 3](#) cita referências legais nos âmbitos Distrital e Federal a respeito da aplicação de Pontos de Função em contratos de Fábrica de Software; o [capítulo 4](#) apresenta algumas definições básicas para a contagem de Pontos de Função; o [capítulo 5](#) classifica os vários serviços de desenvolvimento e manutenção dos sistemas, esclarecendo algumas atividades que não devem ser consideradas nas contagens de Pontos de Função; o [capítulo 6](#) define métricas baseadas em Ponto de Função para dimensionar os serviços ora classificados; o [capítulo 7](#) discorre sobre a aplicação da distribuição de esforço por macroatividades do ciclo de desenvolvimento e manutenção; o [capítulo 8](#) define recomendações para o gerenciamento de projetos, incluindo estimativas e tratamentos quando da mudança de requisitos; finalmente, o [capítulo 9](#) estabelece diretrizes para contagem de múltiplas mídias.

2 MUDANÇAS NAS VERSÕES 1.x

2.1 VERSÃO 1.0

O documento de Diretrizes para Medição em Pontos de Função versão 1.0 foi criado para complementar o Termo de Referência de Fábrica de Software com respeito ao detalhamento das regras pertinentes a medição de serviços de desenvolvimento e manutenção de software, buscando o alinhamento mais próximo possível ao Roteiro de Métricas do SISP, garantindo-se as adequações à

realidade da CAESB. O presente documento assume o propósito, nos termos da APF, de Guia de Contagem, no qual pretende-se adicionar, ao longo do tempo, orientações e definições no intuito de resolver conflitos de entendimento quanto à aplicação da APF em cenários específicos de contagem, bem como contemplar a evolução das regras de acordo com mudanças contextuais da CAESB e com as diretrizes recomendadas pelo MPOG em novas versões do Roteiro de Métricas do SISP.

2.2 NOVIDADES DA VERSÃO 1.1

- **Inclusão das diretrizes para medição de Dados de Código e Conversão de Dados**, observando-se os princípios adotados no Roteiro de Métricas do SISP 2.1, as definições estabelecidas no CPM 4.3, o Roteiro SERPRO de Contagem de Pontos de Função versão 7.0 de 2015 e a realidade do contexto da Caesb;
- **Inclusão das diretrizes sobre Mudança de Requisitos**, formalizadas na ata de Abertura do Contrato 8654-2016, de 02/09/2016, observando-se as recomendações do Roteiro de Métricas do SISP 2.1 e garantindo consonância com os percentuais estabelecidos no Termo de Referência;
- **Inclusão das diretrizes sobre Ordens de Serviços compostas de múltiplos serviços**, que contempla, dentre outros cenários, a **melhoria com redocumentação** e a atualização tecnológica quando envolve mais de um aspecto dentre **versão de linguagem, versão de browser e versão de banco**;
- **Ajuste nos serviços S07, S08 e S09** para remoção da aplicação da distribuição do esforço por macroatividades, vez que o fator de esforço existe fixado nas fórmulas dos respectivos serviços, conforme entendimento a respeito do Roteiro de Métricas SISP, documentado na ata de 02/02/2017, proc. 5494-2015;
- **Ajuste da fórmula do serviço S14 para adequação ao Roteiro de Métricas do SISP 2.1**, conforme autos do processo 5494-2015, em que se distingue o esforço dos cenários: Atualização Simples de Dados, Apuração Especial de Base de Dados e Conversão de Dados;
- **Inclusão de recomendações quanto à macroatividade de Testes**;
- Ajustes menores e correções de português.

3 FUNDAMENTAÇÃO LEGAL

O Decreto No. 34.637 de 2013 do Governo do Distrito Federal (GDF) recepiona a IN (Instrução Normativa) No. 04 de 2010 da SLTI / MPOG, que veda, salvo sob justificativa adequada, a aferição de esforço por métrica homem-hora ou equivalente para serviços de tecnologia da informação.

O Tribunal de Contas da União (TCU) tem publicado vários acórdãos que recomendam a utilização da métrica Ponto de Função Não Ajustado em contratos de prestação de serviços de desenvolvimento e manutenção de sistemas, entre os quais podem ser citados:

- a. Acórdão nº 1.782/2007: recomenda o uso da métrica Ponto de Função como forma de pagamento dos serviços contratados de desenvolvimento e manutenção de sistemas, ao invés de se realizar a conversão dos Pontos de Função em horas, baseado na produtividade média da tecnologia empregada.
- b. Acórdão nº 1.910/2007: em atenção ao princípio da eficiência, faz duas recomendações: adotar a técnica de medição por ponto de função sem ajustes pelas características da aplicação (Pontos de Função não ajustados) e diferenciar, na fórmula de cálculo, os custos dos Pontos de Função para desenvolver novas funcionalidades, daqueles relativos a supressões ou alterações de funcionalidades existentes.
- c. Acórdãos nos 1.125/2009 e 1.274/2010: determinam não vincular a métrica de tamanho funcional (Ponto de Função) com a de esforço (homem-hora).
- d. Acórdãos nos 2.348/2009 e 1.647/2010: reforçam a determinação de não usar qualquer tipo de fator de ajuste na medição por Pontos de Função na contratação de serviços de desenvolvimento de software, para impossibilitar alterações na remuneração da funcionalidade medida, por se basear em interpretação subjetiva dos níveis das características gerais de sistemas, em desacordo com o previsto no art. 54, § 1º, da Lei nº 8.666/93 e art. 2º, VIII, da IN SLTI nº 04/2008. Além disso, o acórdão 1.647/2010 determina que não se use exclusivamente o Manual de Práticas de Contagem (CPM) do IFPUG nas contratações de serviços de desenvolvimento, e que sejam adicionadas cláusulas complementares que elucidem pontos não abordados pelo CPM [1].

4 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” [3]. 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) [1] apresenta as regras de contagem de Pontos de Função de projetos de desenvolvimento, projetos de melhoria e aplicações implantadas. A [Figura 01](#) ilustra o procedimento de contagem de Pontos de Função, descrito nas seções seguintes.

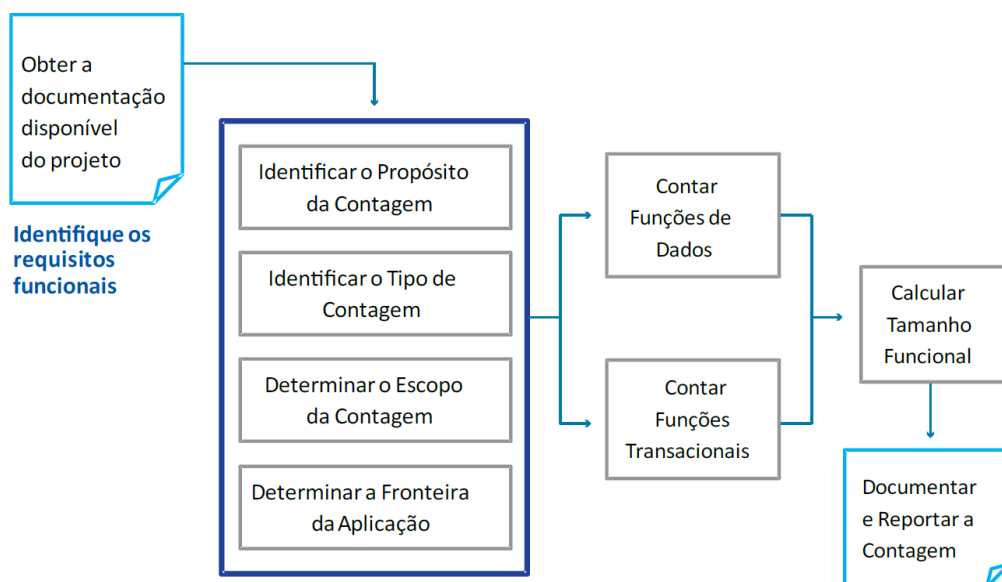


Figura 01 - Procedimento de Contagem de Pontos de Função

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.

O estabelecimento da fronteira da aplicação pode ser subjetivo. Por exemplo, em uma aplicação com vários módulos, a fronteira pode ser estabelecida para cada módulo ou subsistema ou, ainda, pode-se considerar toda a aplicação, dependendo da visão do usuário. De fato, a definição da fronteira depende de processos de negócios, além disso, o posicionamento da fronteira influencia a contagem de Pontos de Função.

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 4.3 e as definições complementares estabelecidas no presente documento [1].

Calcular Tamanho Funcional

O Manual de Práticas de Contagem de Pontos de Função (CPM 4.3) define as regras de contagem de Pontos de Função para projetos de desenvolvimento e de melhoria (manutenção evolutiva) de software [1]. No entanto, os projetos de software não estão limitados a desenvolvimento e melhoria. Desta forma, o presente documento define métricas para dimensionar também o tamanho de outros tipos de projetos de manutenção, não contemplados no CPM. Ressalta-se que todas as diretrizes estão embasadas no Roteiro de Métricas do SISP 2.1 [2].

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

- 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.

5 DEFINIÇÃO DOS SERVIÇOS

Os serviços de desenvolvimento e manutenção estão classificados conforme a [Tabela 01](#):

Tabela 01 - Classificação dos Serviços

Cód.	Serviço	Descrição
S01	Desenvolvimento	Projeto para desenvolver a primeira versão de um software.
S02	Melhoria	Mudanças em requisitos funcionais de um software existente.
S03	Migração de Dados	Caracterizado quando há esforço relativamente alto dessa atividade, tais como, nos casos de migração de dados de banco de dados hierárquico para relacional ou no tratamento de funções complexas de migração de dados. O grau de esforço será definido discricionariamente pela CAESB.
S04	Manutenção Corretiva	Altera o software para correção de defeitos (<i>bugs</i>).
S05	Mudança de Plataforma – Linguagem de Programação	Redesenvolvimento de software em outra linguagem de programação.
S06	Mudança de Plataforma – Banco de Dados	Redesenvolvimento de software para utilizar um outro sistema gerenciador de banco de dados.
S07	Atualização de Versão – Linguagem de Programação	Atualização de versão de linguagem de programação de um software existente ou de parte dele.
S08	Atualização de Versão – Browser	Atualização de software existente para executar em novas versões de um mesmo browser ou para suportar a execução em mais de um browser.
S09	Atualização de Versão – Banco de Dados	Atualização de versão do sistema gerenciador de banco de dados.
S10	Manutenção em Interface	A manutenção em interface, denominada na literatura de manutenção cosmética, é associada às alterações de interface, por exemplo: fonte de letra, cores de telas, logotipos, mudança de botões na tela, mudança de posição de campos ou texto na tela.
S11	Adaptação em Requisitos Não Funcionais	Manutenção adaptativa associada a solicitações que envolvem aspectos não funcionais, sem alteração em requisitos funcionais.
S12	Apuração Especial – Geração de Relatório	Geração de relatório em uma ou mais mídias para o usuário.

S13	Apuração Especial – Reexecução	Reexecução de relatório gerado no serviço S12.
S14	Atualização de Dados	Manutenção de dados na base de dados, que inclui atualizações simples, apurações especiais, conversões de dados e cargas que atravessam a fronteira.
S15	Manutenção de Página Estática	Desenvolvimento ou manutenção específica em página estática de portal, intranet ou website, com ou sem CMS (<i>Content Management System</i>).
S16	Manutenção de Documentação	Serviço de geração de documentação ou atualização de documentação existente, incluindo sistemas legados.
S17	Verificação de Erro	A verificação de erro é a demanda referente a todo comportamento anormal ou indevido apontado pelo cliente nos softwares. Se não for constatado o problema apontado pelo cliente ou o mesmo for decorrente de regras de negócio implementadas ou da utilização incorreta das funcionalidades, será realizada a aferição do tamanho em Pontos de Função das funcionalidades verificadas e que foram reportadas pelo cliente.
S18	Teste	Refere-se exclusivamente ao teste ou automação de teste do conjunto de funções transacionais que estão além daquelas afetadas diretamente por um projeto de manutenção.
S19	Manutenção de Componente Interno Reusável	Serviço de manutenção em componentes específicos de software que são reusados por várias funcionalidades do software.
S20	Serviço em Garantia	Encontram-se, nesta categoria, serviços de correção nos artefatos entregues pela Fábrica de Software, incluindo relatórios, documentação, configuração, software e dados, relativo a quaisquer dos serviços S01 a S19, desde que atendidas as condições de garantia previstas em contrato. Também estão nessa categoria as correções em banco de dados quando os dados incorretos forem decorrentes de defeitos em funcionalidades de software desenvolvidas pela Fábrica de Software, independente de envolver ou não cálculos ou procedimentos complexos.

Deve-se ressaltar que o processo de desenvolvimento de soluções possui várias atividades que não são escopo deste documento. Essas atividades precisam ser consideradas como um projeto separado, levando-se em conta as horas realizadas e que devem estar associadas a produtos contratados e entregáveis. São atividades categorizadas nesta condição: Treinamentos em Tecnologia da Informação; Desenvolvimento de Cursos para EaD; Mapeamento de Processos de Negócio; Elaboração de Plano Diretor de Tecnologia da Informação (PDTI); Definição de Processo de Desenvolvimento de Soluções; Treinamento para Implantação. Outros serviços prestados que também não possuem contagem de Pontos de Função associada são os seguintes:

- Administração de Dados - este serviço requer uma equipe de administradores de dados (AD) com um número de profissionais definido junto à CAESB, dedicada para atender as demandas associadas à definição e manutenção do modelo de dados de negócio do cliente. **É importante**

ressaltar que as atividades de banco de dados associadas ao projeto de desenvolvimento ou de manutenção, por exemplo, preparação de ambiente (testes, homologação, implantação), desempenhadas pelos DBA da equipe de desenvolvimento, já estão consideradas dentro do projeto de software, não cabendo cobrança adicional.

- Treinamento para Implantação - são demandas de treinamentos sobre utilização do sistema a ser implantado, para os gestores de solução do cliente e usuários. O esforço deste serviço deve ser considerado separadamente da estimativa de esforço derivada da contagem de PF. A remuneração deste serviço deve ser calculada, levando-se em conta o preço da hora de consultoria desse tipo de serviço, incluindo atividades de preparação de treinamento e de instrutoria. As responsabilidades, condições e premissas que envolvem as necessidades de serviços de treinamento devem estar descritas no contrato. Deve-se ressaltar que este treinamento para implantação pode ser definido na modalidade de EaD, sendo tratado como um projeto de treinamento a parte. O esforço deste é considerado dentro do projeto de EaD, que não faz parte do projeto de desenvolvimento ou manutenção em questão.
- Especificação de Negócio: esta é a primeira atividade a ser executada em uma demanda de projeto de desenvolvimento e/ou de manutenção. O objetivo dessa atividade é gerar a especificação da demanda, como, por exemplo, um documento de visão do projeto ou qualquer outro documento inicial de requisitos definido no processo de desenvolvimento da CAESB. **É importante ressaltar que essa atividade é de responsabilidade dos analistas de negócio da CAESB, de acordo com a Instrução Normativa SLTI/MP nº 4, de 11 de setembro de 2014.** No entanto, por falta de pessoal, alguns órgãos e entidades têm contratado estas atividades, que devem ser faturadas em horas de consultoria. O documento inicial de requisitos gerado nessa atividade é o artefato utilizado como insumo para o planejamento do projeto (estimativa de tamanho funcional em Pontos de Função) e para o processo de desenvolvimento de software.

6 MEDIÇÃO DOS SERVIÇOS

Este capítulo tem como propósito descrever os diversos tipos de projetos de software e definir métricas para seu dimensionamento baseadas nas regras de contagem de Pontos de Função do CPM 4.3[1].

Cabe ressaltar que, em alguns casos, a CAESB pode não ter necessidade de requisitar todas as fases ou macroatividades do ciclo de desenvolvimento e manutenção do software. Dessa forma, a Fábrica de Software será remunerada pela contagem de Pontos de Função considerando apenas os percentuais das macroatividades requisitadas, conforme descrito no capítulo 7. Nesses casos, o tamanho final do serviço, em Pontos de Função, será o resultado da contagem detalhada, conforme regras deste capítulo, multiplicado pela soma dos percentuais das macroatividades solicitadas pela CAESB na Ordem de Serviço.

O tamanho final de cada serviço, em Pontos de Função, deverá ser arredondado para duas casas decimais de precisão, ISO 31-0:1992 ou ABNT NBR 5891:1977. O arredondamento deverá ser realizado após a multiplicação pelos percentuais de macroatividade, quando se aplicarem.

6.1 DESENVOLVIMENTO (S01)

É o projeto para desenvolver a primeira versão de um software. Seu tamanho funcional é a medida das funcionalidades entregues ao usuário no final do projeto. Segue a fórmula de cálculo utilizada no dimensionamento de projetos de desenvolvimento de software:

$$PF_DESENVOLVIMENTO = PF_INCLUÍDO$$

A conversão de dados, nos termos do CPM, quando houver, deverá ser tratada conforme [seção 6.14](#). Recomenda-se suprimir a contagem do PF_CONVERSÃO e considerar um projeto separado de migração de dados, [seção 6.3](#), nos casos onde for caracterizado um esforço relativamente maior dessa atividade, como, por exemplo, nos projetos que envolvem a migração de dados de banco de dados hierárquico para banco de dados relacional ou o tratamento de funções complexas de migração de dados.

6.2 MELHORIA (S02)

Também denominado manutenção evolutiva, o projeto de melhoria está associado às mudanças em requisitos funcionais de um software existente, ou seja, à inclusão de novas funcionalidades, alteração ou exclusão de funcionalidades.

Segundo o padrão IEEE Std 1219 [4], 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].

Segue a fórmula de cálculo utilizada no dimensionamento de projetos de melhoria de software:

$$PF_MELHORIA = PF_INCLUÍDO + (FI \times PF_ALTERADO) + (0,20 \times PF_EXCLUÍDO)$$

Fator de Impacto (FI):

- 50% para funcionalidade que foi desenvolvida ou mantida **pela mesma empresa**.
- 90% para funcionalidade que foi desenvolvida ou mantida **por empresa diferente**. Observe que o percentual de 90% apenas será considerado na primeira demanda em cada funcionalidade.

A conversão de dados, nos termos do CPM, quando houver, deverá ser tratada conforme [seção 6.14](#). Recomenda-se suprimir a contagem do PF_CONVERSÃO e considerar um projeto separado de migração de dados, [seção 6.3](#), nos casos onde for caracterizado um esforço relativamente maior dessa atividade, como, por exemplo, nos projetos que envolvem a migração de dados de banco de dados hierárquico para banco de dados relacional ou o tratamento de funções complexas de migração de dados.

Caso seja demandada a redocumentação das funcionalidades alteradas, porque a documentação não existe ou está desatualizada, gerando-se a documentação completa das mesmas, deve-se considerar as diretrizes dispostas no [capítulo 11](#).

O teste, quando demandado, pode restringir-se às regras documentadas e aprovadas na fase de especificação do projeto. Caso for fornecida para a contratada a documentação prévia completa das funcionalidades alteradas ou for solicitada a redocumentação (S18) delas, o teste deverá contemplar todas as regras dessas funcionalidades. No caso de existência de testes automatizados, recomenda-se enfaticamente a execução do Teste de Regressão.

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, um Tipo de Dados (DET – *Data Element Type*) é um atributo único, reconhecido pelo usuário e não repetido [1]. 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 para alfanumérico), caso a mudança decorra de alteração de regra de negócio.

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

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

O CPM 4.3 define **lógica de processamento** como requisitos especificamente solicitados pelo usuário para completar um processo elementar [1]. 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 documento 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 documento 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 de 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.**

6.3 MIGRAÇÃO DE DADOS (S03)

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

$$PF_MIGRAÇÃO_DADOS = PF_INCLUÍDO$$

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.

6.4 MANUTENÇÃO CORRETIVA (S04)

Altera o software para correção de defeitos (*bugs*) em sistemas existentes. Quando as funcionalidades a serem corrigidas tiverem sido desenvolvidas pela Fábrica de Software, a manutenção será do tipo Garantia se atender as condições de garantia previstas em contrato.

A aferição do tamanho em Pontos de Função das funcionalidades corrigidas deve considerar um fator de impacto (FI) sobre o PF_ALTERADO.

$$PF_CORRETIVA = FI \times PF_ALTERADO$$

Fator de Impacto (FI):

- 50% para funcionalidade que foi desenvolvida ou mantida **pela mesma empresa**.
- 90% para funcionalidade que foi desenvolvida ou mantida **por empresa diferente**. Observe que o percentual de 90% apenas será considerado na primeira demanda em cada funcionalidade.

As demandas de manutenção corretiva não contemplam atualização de documentação existente da funcionalidade corrigida, pois considera-se que, normalmente, manutenção corretiva não se refere a erros de requisitos. Caso seja erro em requisitos, essa demanda deve ser tratada como um projeto de melhoria, descrito na [seção 6.2](#). A manutenção corretiva não exclui, contudo, a produção de documentação descritiva a respeito do erro e da manutenção realizada.

Caso seja demandada a redocumentação das funcionalidades corrigidas, porque a documentação não existe ou está desatualizada, gerando-se a documentação completa das mesmas, deve-se considerar as diretrizes dispostas no [capítulo 11](#).

O teste, quando demandado, pode restringir-se às regras que tiveram ação corretiva. Caso for fornecida para a contratada a documentação prévia completa das funcionalidades alteradas ou for solicitada a redocumentação (S18) delas, o teste deverá contemplar todas as regras dessas funcionalidades. No caso de existência de testes automatizados, recomenda-se enfaticamente a execução do Teste de Regressão.

6.5 MUDANÇA DE PLATAFORMA - LINGUAGEM DE PROGRAMAÇÃO (S05)

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].

Os projetos que se enquadram em mais de uma Mudança de Plataforma (S05 e S06) devem ser contados apenas uma vez, considerando o tipo de serviço com maior contagem de Pontos de Função.

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 contadas.

PF_REDESENVOLVIMENTO_LINGUAGEM = PF_INCLUÍDO

A conversão de dados, nos termos do CPM, quando houver, deverá ser tratada conforme [seção 6.14](#). Recomenda-se suprimir a contagem do PF_CONVERSÃO e considerar um projeto separado de migração de dados, [seção 6.3](#), nos casos onde for caracterizado um esforço relativamente maior dessa atividade, como, por exemplo, nos projetos que envolvem a migração de dados de banco de dados hierárquico para banco de dados relacional ou o tratamento de funções complexas de migração de dados.

6.6 MUDANÇA DE PLATAFORMA - BANCO DE DADOS (S06)

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

Os projetos que se enquadram em mais de uma Mudança de Plataforma (S05 e S06) devem ser contados apenas uma vez, considerando o tipo de serviço com maior contagem de Pontos de Função.

Em casos de mudança de banco hierárquico para relacional, devido às mudanças envolvidas, deve-se considerar como um novo projeto de desenvolvimento, ou seja, as funções de dados e funções transacionais devem ser contadas. Assim, será utilizada a fórmula de projeto de desenvolvimento do CPM 4.3 [1], conforme fórmula abaixo:

$$PF_REDESENVOLVIMENTO_BD_HIERÁRQUICO = PF_INCLUÍDO$$

Caso a demanda de redesevolvimento seja de um sistema gerenciador de banco de dados relacional para outro relacional, deve ser utilizada a fórmula a seguir. Se não existir mudança nas funções de dados, ou seja, o banco de dados da aplicação seja mantido, então as funções de dados não devem ser contadas.

$$PF_REDESENVOLVIMENTO_BD_RELACIONAL = PF_ALTERADO \times 0,30$$

O PF_ALTERADO deve considerar apenas as funcionalidades impactadas. As funcionalidades que possuem apenas demandas de testes devem ser dimensionadas em conformidade com a [seção 6.18](#):

$$PF_REDESENVOLVIMENTO_BD_RELACIONAL \text{ (apenas teste)} = PFT \times 0,20$$

Observe que uma mesma funcionalidade não pode ser contada como Implementação (PF_ALTERADO) e também como Teste (PFT).

Caso seja demandada a redocumentação das funcionalidades, porque a documentação não existe ou está desatualizada, gerando-se a documentação completa das mesmas, deve-se considerar o fator equivalente a Requisitos, em conformidade com a [seção 6.16](#):

$$PF_REDESENVOLVIMENTO_BD_RELACIONAL \text{ (documentação)} = PF_NÃO_AJUSTADO \times 0,30$$

Em todos os casos, a conversão de dados, nos termos do CPM, quando houver, deverá ser tratada conforme [seção 6.14](#). Recomenda-se suprimir a contagem do PF_CONVERSÃO e considerar um projeto separado de migração de dados, [seção 6.3](#), nos casos onde for caracterizado um esforço relativamente maior dessa atividade, como, por exemplo, nos projetos que envolvem a migração de dados de banco de dados hierárquico para banco de dados relacional ou o tratamento de funções complexas de migração de dados.

6.7 ATUALIZAÇÃO DE VERSÃO - LINGUAGEM DE PROGRAMAÇÃO (S07)

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.

$$PF_ATUALIZAÇÃO_VERSÃO_LINGUAGEM = PF_ALTERADO \times 0,30$$

O PF_ALTERADO deve considerar apenas as funcionalidades impactadas. As funcionalidades que possuem apenas demandas de testes devem ser dimensionadas em conformidade com a [seção 6.18](#):

$$PF_ATUALIZAÇÃO_VERSÃO_LINGUAGEM \text{ (apenas teste)} = PFT \times 0,20$$

Observe que uma mesma funcionalidade não pode ser contada como Implementação (PF_ALTERADO) e também como Teste (PFT).

Caso seja demandada a redocumentação das funcionalidades, porque a documentação não existe ou está desatualizada, gerando-se a documentação completa das mesmas, deve-se considerar o fator equivalente a Requisitos, em conformidade com a [seção 6.16](#):

$$PF_ATUALIZAÇÃO_VERSÃO_LINGUAGEM \text{ (documentação)} = PF_NÃO_AJUSTADO \times 0,30$$

Quando a atualização tecnológica de versão envolver mais de um aspecto dentre versão de linguagem, versão de browser e versão de banco, considerar o disposto no [capítulo 11](#).

6.8 ATUALIZAÇÃO DE VERSÃO - BROWSER (S08)

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.

$$PF_ATUALIZAÇÃO_VERSÃO_BROWSER = PF_ALTERADO \times 0,30$$

O PF_ALTERADO deve considerar apenas as funcionalidades impactadas. As funcionalidades que possuem apenas demandas de testes devem ser dimensionadas em conformidade com a [seção 6.18](#):

$$PF_ATUALIZAÇÃO_VERSÃO_BROWSER \text{ (apenas teste)} = PFT \times 0,20$$

Observe que uma mesma funcionalidade não pode ser contada como Implementação (PF_ALTERADO) e também como Teste (PFT).

Caso seja demandada a redocumentação das funcionalidades, porque a documentação não existe ou está desatualizada, gerando-se a documentação completa das mesmas, deve-se considerar o fator equivalente a Requisitos, em conformidade com a [seção 6.16](#):

$$PF_ATUALIZAÇÃO_VERSÃO_BROWSER \text{ (documentação)} = PF_NÃO_AJUSTADO \times 0,30$$

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

É importante ressaltar que os sistemas Web devem seguir o padrão W3C, como recomendado na e-Ping. Caso seja necessário fazer a adequação do sistema para atendimento ao padrão W3C, pode-se usar a fórmula acima.

Quando a atualização tecnológica de versão envolver mais de um aspecto dentre versão de linguagem, versão de browser e versão de banco, considerar o disposto no [capítulo 11](#).

6.9 ATUALIZAÇÃO DE VERSÃO - BANCO DE DADOS (S09)

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.

$$PF_ATUALIZAÇÃO_VERSÃO_BD = PF_ALTERADO \times 0,30$$

O PF_ALTERADO deve considerar apenas as funcionalidades impactadas. As funcionalidades que possuem apenas demandas de testes devem ser dimensionadas em conformidade com a [seção 6.18](#):

$$PF_ATUALIZAÇÃO_VERSÃO_BD \text{ (apenas teste)} = PFT \times 0,20$$

Observe que uma mesma funcionalidade não pode ser contada como Implementação (PF_ALTERADO) e também como Teste (PFT).

Caso seja demandada a redocumentação das funcionalidades, porque a documentação não existe ou está desatualizada, gerando-se a documentação completa das mesmas, deve-se considerar o fator equivalente a Requisitos, em conformidade com a [seção 6.16](#):

$$PF_ATUALIZAÇÃO_VERSÃO_BD \text{ (documentação)} = PF_NÃO_AJUSTADO \times 0,30$$

Quando a atualização tecnológica de versão envolver mais de um aspecto dentre versão de linguagem, versão de browser e versão de banco, considerar o disposto no [capítulo 11](#).

6.10 MANUTENÇÃO EM INTERFACE (S10)

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, 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, cabeçalho ou rodapé de um relatório;
- Alteração de *labels* de uma tela de consulta.

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.

$$PF_INTERFACE = 0,6 PF \times \text{Quantidade de Funções Transacionais Impactadas}$$

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, cabendo, a critério da CAESB, solicitar adicionalmente um serviço [S16](#).

Observação 1: *Help*

As demandas de projetos de desenvolvimento de sistemas ou de manutenção de funcionalidades 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. 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].

6.11 ADAPTAÇÃO EM REQUISITOS NÃO FUNCIONAIS (S11)

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:

- 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 webservice 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 estar navegando 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]:

$$PF_ADAPTATIVA = FI \times PF_ALTERADO$$

Fator de Impacto (FI):

- 50% para funcionalidade que foi desenvolvida ou mantida **pela mesma empresa**.
- 90% para funcionalidade que foi desenvolvida ou mantida **por empresa diferente**. Observe que o percentual de 90% apenas será considerado na primeira demanda em cada funcionalidade.

Caso seja demandada a redocumentação das funcionalidades adaptadas, porque a documentação não existe ou está desatualizada, gerando-se a documentação completa das mesmas, deve-se considerar as diretrizes dispostas no [capítulo 11](#).

O teste, quando demandado, pode restringir-se às regras documentadas e aprovadas na fase de especificação do projeto. Caso for fornecida para a contratada a documentação prévia completa das funcionalidades alteradas ou for solicitada a redocumentação (S18) delas, o teste deverá contemplar todas as regras dessas funcionalidades. No caso de existência de testes automatizados, recomenda-se enfaticamente a execução do Teste de Regressão

6.12 APURAÇÃO ESPECIAL - GERAÇÃO DE RELATÓRIO (S12)

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 a contagem de Pontos de Função das funcionalidades desenvolvidas. Frequentemente, estas funcionalidades são classificadas como Saídas Externas. Também podem ser classificadas como Consultas Externas, caso não possuam cálculos ou criação de dados derivados.

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

$$PF_APURAÇÃO_RELATÓRIO = PF_INCLUÍDO$$

6.13 APURAÇÃO ESPECIAL - REEXECUÇÃO (S13)

Em alguns casos, a CAESB pode ter interesse em executar uma apuração especial mais de uma vez.

A reexecução de uma apuração especial 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:

$$PF_REEXECUÇÃO_APURAÇÃO = PF_APURAÇÃO \times 0,10$$

6.14 ATUALIZAÇÃO DE DADOS (S14)

Esta seção agrupa os cenários identificados no Roteiro de Métricas do SISP 2.1 [2] pertinentes a manutenção de dados em banco de dados, que inclui atualização simples, apuração especial e conversão de dados. Adicionalmente, são estabelecidas diretrizes para manutenção de Dados de Código, quando não mantidos por processos elementares de aplicação, em demandas após o projeto de desenvolvimento ou manutenção.

Caso a atualização de dados 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.

Caso 1: *Alteração Simples (AS)*

Se enquadram nesta categoria as demandas associadas a atualizações manuais, normalmente 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.

$$PF_ATUALIZAÇÃO_BD = PF_INCLUÍDO \times 0,10$$

Se enquadram nesta categoria as manutenções de **Dados de Código**, observando-se o disposto no [capítulo 10](#).

Caso 2: *Apuração Especial (APE)*

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 o ajuste de dados inconsistentes na base de dados ou atualização em função de modificação da estrutura de dados ou mudanças de regras negociais. Normalmente, nesse tipo de atualização são afetados múltiplos registros. As funcionalidades desenvolvidas são classificadas, geralmente, como Entradas Externas.

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

$$PF_APURAÇÃO_BD = PF_INCLUÍDO$$

Caso 3: *Conversão de dados (CONV)*

As funcionalidades de conversão de dados associadas a projetos de desenvolvimento e manutenção de sistemas são contadas conforme o CPM 4.3 [1]:

$$PF_CONVERSÃO_BD = PF_CONVERSÃO$$

A conversão de dados pode se aplicar a cargas iniciais de Dados de Código caso os dados atravessem a fronteira da aplicação, conforme explicado no [capítulo 10](#).

O Roteiro de Métricas do SISP considera o PF_CONVERSÃO nas fórmulas relativas a projetos de desenvolvimento e manutenção de sistemas. Contudo, o presente documento moveu esse fator para esta seção. Trata-se apenas de mudança na classificação dos serviços, o que não afeta a contagem total dos projetos. Ou seja, quando for o caso, conforme sinalizado em cada serviço, a conversão de dados será contada como um serviço S14 complementar ao projeto.

6.15 MANUTENÇÃO DE PÁGINA ESTÁTICA (S15)

Nesta seção são tratados desenvolvimentos, manutenções específicas e publicações de páginas estáticas de portais, *intranets* ou *websites*. Por exemplo: criação de página HTML, atualização de menu estático, atualização de texto ou *banner* estáticos em páginas HTML existentes.

Caso o desenvolvimento de páginas estáticas esteja contido em um projeto de desenvolvimento, então elas serão contabilizadas no projeto de desenvolvimento e não devem ser mensuradas em separado. Ou seja, esta seção se aplica quando ocorrer a demanda exclusivamente para o desenvolvimento ou manutenção de páginas estáticas.

Estas demandas são consideradas como desenvolvimento de consultas. Nestes casos, considera-se 20% dos Pontos de Função das consultas desenvolvidas. Cada página é contada como uma consulta. As consultas são consideradas Consultas Externas simples (3 PF). Ou seja, 0,6 PF por cada página desenvolvida ou mantida, de acordo com a fórmula abaixo:

$$PF_PUBLICAÇÃO = 0,6 PF \times \text{Quantidade de Páginas Alteradas ou Excluídas}$$

As demandas de criação de logomarcas ou identidade visual, além de outras demandas de criação de arte, associadas à área de Comunicação Social, não são enquadradas nessa categoria. Tais demandas não se referem a contratos de prestação de serviços de desenvolvimento e manutenção de sistemas, portanto não são consideradas neste documento.

É recomendada a construção de portais com ferramentas que apoiem a construção de conteúdo pelo usuário, os chamados Gerenciadores de Conteúdo, de modo a minimizar as demandas de criação de páginas estáticas.

6.16 MANUTENÇÃO DE DOCUMENTAÇÃO (S16)

Nesta seção são tratadas demandas de geração de documentação ou atualização de documentação existente. Observe que o desenvolvedor pode precisar realizar uma engenharia reversa da aplicação para gerar a documentação. Para este tipo de projeto foi definido o fator de impacto de 30% dos Pontos de Função, considerando a macroatividade de Requisitos e a geração de artefatos associados a requisitos, em consonância com o [capítulo 7](#).

$$PF_DOCUMENTAÇÃO = PF_NÃO_AJUSTADO \times 0,30$$

6.17 VERIFICAÇÃO DE ERRO (S17)

As verificações de erro ou a análise e solução de problemas são as demandas referentes a todo comportamento anormal ou indevido apontado pelo cliente nos sistemas existentes. Neste caso, a equipe de desenvolvimento da Fábrica de Software se mobilizará para encontrar as causas do problema ocorrido.

Se for constatado o problema apontado pelo cliente, então não se aplica o serviço de Verificação de Erros e, nesse caso, a demanda será atendida como manutenção de software ou de dados, conforme a necessidade, ou Serviço em Garantia (S20) se aplicáveis as cláusulas contratuais.

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 das quais o cliente reportou erro.

$$PF_VERIFICAÇÃO = PF_FUNCIONALIDADE_REPORTADA_COM_ERRO \times 0,20$$

No caso específico em que o erro procede e a CAESB opta por implementar a correção, não se aplicando a garantia contratual, recomenda-se a apuração do serviço pela fórmula acima, em decorrência do esforço de análise realizado pela Fábrica de Software.

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 deste documento de métricas, não se aplicando verificação de erros nestes casos.

6.18 TESTE (S18)

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 [5]. 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.

Esta seção não se restringe à realização de testes manuais ou exploratórios. Contempla-se, quando solicitado pela CAESB, a automação de teste, como, por exemplo, automação de Teste de Aceitação.

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 Pontos de Função deve ser feita de acordo com a fórmula abaixo. Para este tipo de projeto foi definido o fator de impacto de 20% considerando a macroatividade de Testes e a geração de artefatos associados a testes, em consonância com o [capítulo 7](#).

$$\text{PF_TESTES} = \text{PFT} \times 0,20$$

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 Fábrica de Software considerando-se a documentação de Testes definida no processo de desenvolvimento da PRT/CAESB. Observe que estas funções farão parte do escopo do projeto de manutenção.

6.19 MANUTENÇÃO DE COMPONENTE INTERNO REUSÁVEL (S19)

Em alguns casos são demandadas manutenções em componentes específicos de uma aplicação e estes são reusados por várias funcionalidades da aplicação. Por exemplo, uma mudança em uma rotina de validação de um CPF usada em várias funcionalidades de cadastro. Se considerarmos o método de contagem de projetos de melhoria do CPM, seriam contadas todas as funcionalidades impactadas por essa mudança [1].

No entanto, este documento propõe que o componente, o qual deverá ser testado, 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 CAESB e dimensionadas por meio da métrica Pontos de Função de Teste proposta na [seção 6.18](#).

$$\text{PF_COMPONENTE} = \text{PF_NÃO_AJUSTADO}$$

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”.

6.20 SERVIÇO EM GARANTIA (S20)

Encontram-se, nesta categoria, serviços de correção de artefatos entregues pela Fábrica de Software, incluindo relatórios, documentação, configuração, código-fonte e dados, relativo a quaisquer dos serviços S01 a S19, desde que atendidas as condições de garantia previstas em contrato.

Também estão nessa categoria as correções em banco de dados quando os dados incorretos forem decorrentes de defeitos em funcionalidades de software desenvolvidas pela Fábrica de Software, independente de envolver ou não cálculos ou procedimentos complexos.

PF_GARANTIA = 0

7 DISTRIBUIÇÃO DO ESFORÇO

Com o objetivo de representar a distribuição do esforço da produção de software no âmbito da PRT/CAESB, foram definidas três macroatividades: Requisitos, Implementação e Testes.

Para cada serviço encaminhado à Fábrica de Software, a CAESB terá flexibilidade na solicitação de todas ou apenas de alguma(s) das três macroatividades acima definidas. A Fábrica de Software será remunerada de acordo com os percentuais das macroatividades que forem solicitadas pela CAESB, conforme estabelecido na [Tabela 02](#).

Tabela 02 - Percentuais de Pontos de Função por Macroatividade

Macroatividade	Percentual de Pontos de Função
Requisitos	30% sobre a contagem final
Implementação	50% sobre a contagem final
Testes	20% sobre a contagem final

A distribuição do esforço pela [Tabela 02](#) aplica-se aos serviços:

- S01 - Desenvolvimento
- S02 - Melhoria
- S04 - Manutenção Corretiva
- S05 - Mudança de Plataforma - Linguagem de Programação
- S06 - Mudança de Plataforma - Banco (exceto Relacional-Relacional)

S11 - Adaptação em Requisitos Não Funcionais
S19 - Manutenção em CIR

Para os serviços abaixo elencados, suas fórmulas contemplam fator(es) de esforço específico(s), conforme Roteiro de Métricas do SISP, e não segue distribuição de esforço conforme [Tabela 02](#):

S06 - Mudança de Plataforma - Banco (Relacional-Relacional)
S07 - Atualização de Versão - Linguagem de Programação
S08 - Atualização de Versão - Browser
S09 - Atualização de Versão - Banco de Dados

Não se aplica a distribuição de esforço para os demais casos, pois são serviços demandados de maneira completa:

S03 - Migração de Dados
S10 - Manutenção em Interface
S12 - Apuração Especial - Geração de Relatório
S13 - Apuração Especial - Reexecução
S14 - Atualização de Dados
S15 - Manutenção de Página Estática
S16 - Manutenção de Documentação (equiv. esforço de Requisitos)
S17 - Verificação de Erro
S18 - Teste (equivale ao esforço de Teste)

Entende-se por macroatividade de Requisitos o conjunto de tarefas associado à elicitação, organização, análise e documentação de requisitos, além da redocumentação, através, por exemplo, da engenharia reversa de software.

Considera-se, para fins de remuneração de esforço, que outras macroatividades, tais como Análise e Design, Implantação, Gerenciamento de Configuração e Mudança, Gerenciamento de Projetos e Gerenciamento de Ambiente, estão agregadas ao longo do processo de desenvolvimento e manutenção de sistemas, independentemente de quais macroatividades foram solicitadas no serviço demandado à Fábrica de Software.

8 PLANEJAMENTO E ACOMPANHAMENTO

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, são apresentados: diretrizes para estimar o tamanho dos projetos de software em PF, o modelo simplificado de estimativas para estimar o esforço dos projetos em homem-hora (HH) e a fórmula de Capers Jones para estimar os prazos dos projetos [6]. Também são apresentadas recomendações para

gerenciamento de progresso de projetos e de mudança de requisitos. A Figura 02 ilustra um processo de estimativas de projetos de software.

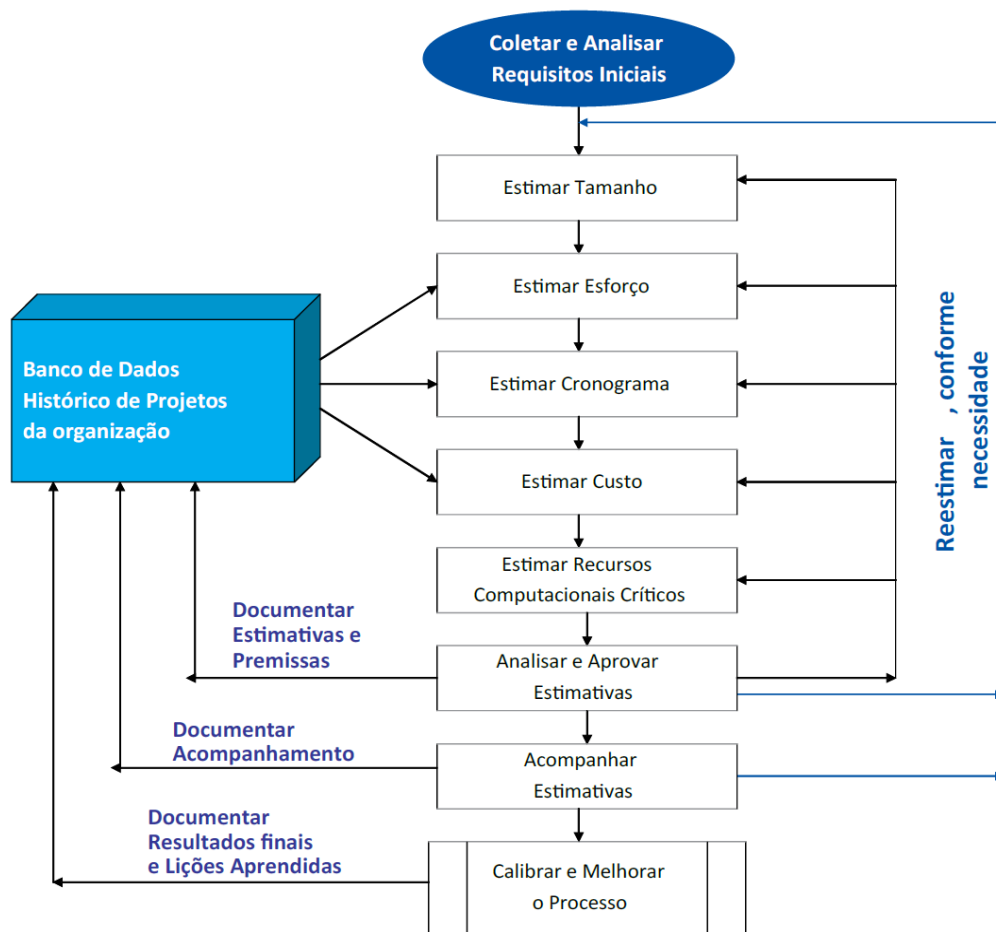


Figura 02 - Processo de Estimativas de Projetos de Software [7]

8.1 ESTIMATIVA DE TAMANHO

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 como, por exemplo, o documento de concepção ou formalização simples de requisitos. O estimador deve analisar os requisitos para garantir a qualidade e então estimar o tamanho do projeto de software. O próximo passo é a derivação das estimativas de esforço, prazo (cronograma), custo (orçamento) com base na estimativa de tamanho e nos dados históricos de projetos concluídos da organização, assim como o estabelecimento da estimativa de recursos computacionais críticos e dos recursos da equipe a ser alocada ao projeto. Neste ponto, as principais estimativas foram geradas e precisam ser documentadas. As premissas e suposições utilizadas na geração das estimativas, dentre as quais: complexidade do projeto, plataforma

de desenvolvimento, tipo do projeto, percentual de evolução de requisitos, também devem ser documentadas [7].

A realização das estimativas por um analista de métricas que não atue na equipe do projeto constitui uma prática recomendada. O analista de métricas deve analisar também a consistência da documentação utilizada na estimativa. No decorrer do processo de desenvolvimento, as estimativas devem ser acompanhadas conforme o refinamento dos requisitos. O projeto deve ser reestimado após a fase de elaboração ou requisitos, quando for gerada a especificação de casos de uso, e sempre que ocorrerem mudanças significativas nos requisitos funcionais ou não funcionais. Quando o projeto é concluído, deve-se aferir e documentar o tamanho, prazo, custo, esforço e recursos realizados, assim como outros atributos relevantes do projeto, visando a coleta de dados para a melhoria do processo de estimativas. As lições aprendidas também devem ser documentadas [7].

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

- **Estimativa Inicial** - Realizada após o fechamento do escopo do projeto. Geralmente é baseada em um documento inicial de requisitos como, por exemplo, o documento de concepção. 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 [8].
- **Contagem de Pontos de Função de Referência** - Realizada após o aceite dos requisitos. Geralmente, leva em consideração a especificação dos casos de uso e regras de negócio da aplicação. Pode ser aplicada a Contagem Estimada da NESMA [5] ou a detalhada, conforme este documento.
- **Contagem de Pontos de Função Final** - Realizada após o aceite da aplicação. Esta contagem considera as funcionalidades efetivamente entregues para o usuário pela aplicação. Neste caso, deve ser aplicada a contagem detalhada.

Em um cenário ideal, os resultados da Estimativa Inicial 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.

É importante ressaltar que as mudanças de requisitos também serão consideradas no tamanho do projeto a ser faturado (ver [seção 8.6](#)). Além disso, se estas mudanças forem significativas, maiores que a evolução de requisitos (*scope creep*) prevista na Estimativa Inicial, o prazo e o custo do projeto devem ser reestimados. Toda mudança de requisito deve passar por uma análise de impacto entre CAESB e Fábrica de Software.

8.2 ESTIMATIVA DE ESFORÇO

Uma vez que o tamanho do projeto foi estimado em Pontos de Função, o próximo passo é estimar o esforço de desenvolvimento do projeto. A Engenharia de Software possui vários modelos para estimar esforço de projetos de software baseados em Pontos de Função, sendo o Modelo Simplificado de Estimativas [9] e o Modelo COCOMO II [10] os mais utilizados. Neste documento é adotado o Modelo Simplificado de Estimativas.

O Modelo Simplificado de Estimativas consiste em obter um índice de produtividade em horas/PF para o projeto específico em questão, e então multiplicar o tamanho em PF do projeto pelo índice de produtividade, conforme a fórmula [9]:

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

O índice de produtividade depende de diversos atributos dos projetos, dentre outros: plataforma tecnológica, complexidade do domínio, segurança, desempenho, usabilidade, tamanho do projeto, tipo de manutenção, desenvolvimento de componentes.

A CAESB poderá construir sua própria tabela de produtividade para cada linguagem, com base em seus dados históricos de projetos já realizados, ou utilizar os padrões existentes no mercado.

8.3 ESTIMATIVA DE PRAZO

As estimativas de prazo não são lineares com o tamanho do projeto. O melhor tempo de desenvolvimento é definido como a melhor relação custo x benefício de alocação de recursos e menor prazo de desenvolvimento, dado o tamanho de um projeto específico.

Jones [6] propõe uma fórmula para o cálculo do melhor tempo de desenvolvimento, denominado **Td**, e da **Região Impossível (RI)** de desenvolvimento (Figura 03). Na Região Impossível (RI), a adição de mais recursos ao projeto não implicará em redução no prazo. Note que a curva mostra que quanto menor o prazo almejado para a conclusão do projeto, maior será o esforço requerido e, conseqüentemente, maior o custo do projeto. O aumento do esforço para reduzir o prazo acontece através da realização de horas extras e da inclusão de pessoal adicional. Conseqüentemente, há aumento de custo e o risco de retrabalho. No entanto, a redução de prazo tem um limite, como demonstra a Região Impossível da Figura 03.

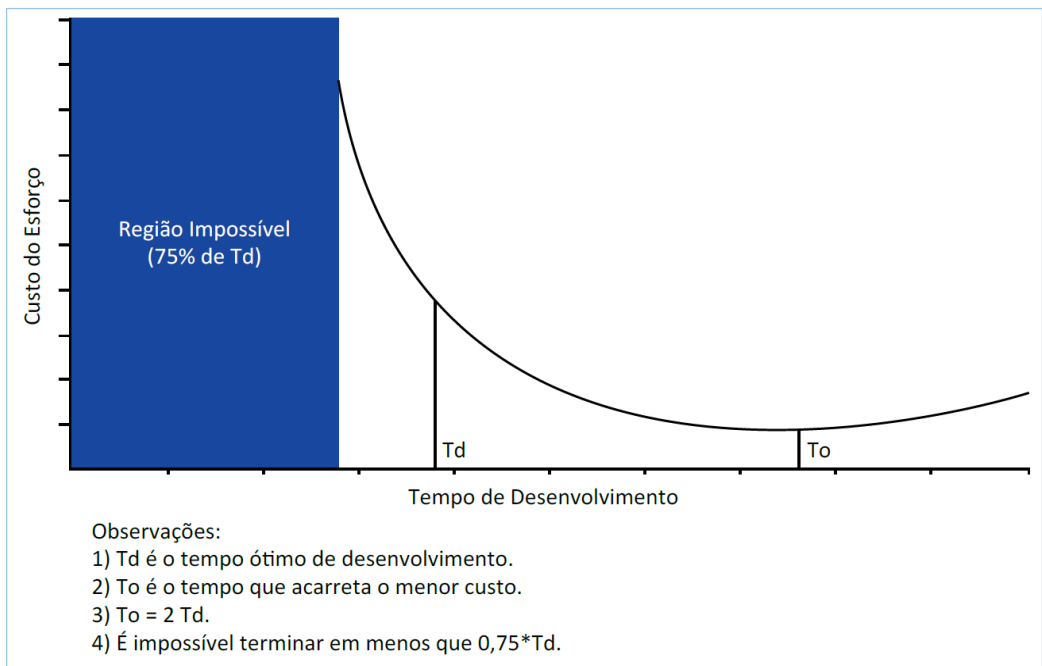


Figura 03 - Relação entre a Estimativa de Prazo e de Esforço

O método utilizado para estimar o prazo dos projetos (Td) é baseado na fórmula de Capers Jones [6]. Esta estima o prazo, baseando-se no tamanho do projeto em Pontos de Função, da seguinte maneira:

$$T_d = V^t$$

Onde:

Td: melhor tempo de desenvolvimento em meses

V: tamanho do projeto em Pontos de Função

t: definido de acordo com a [Tabela 03](#)

Tabela 03 - Expoente t por Tipo de Projeto

Tipo de Sistema	Expoente t
Sistema Comum – Mainframe (desenvolvimento de sistema com alto grau de reuso ou manutenção evolutiva)	0,32 a 0,33
Sistema Comum – WEB ou Cliente Servidor	0,34 a 0,35
Sistema OO (se o projeto OO não for novidade para equipe, não tiver o desenvolvimento de componentes reusáveis, considerar sistema comum)	0,36
Sistema Cliente/Servidor (com alta complexidade arquitetural e integração com outros sistemas)	0,37
Sistema Cliente/Servidor (com alta complexidade arquitetural e integração com outros sistemas)	0,39
Software Básico, Frameworks, Sistemas Comerciais	0,40
Software Militar (ex: Defesa do Espaço Aéreo)	0,45

É importante destacar que o método só deve ser aplicado de forma discricionária, de acordo com a necessidade da CAESB e para projetos com mais de 100 Pontos de Função. Caso o projeto seja menor, o prazo pode ser obtido conforme a [Tabela 04](#), adaptada do SISP 2.1 [2].

Tabela 04 - Estimativa de Prazo de Projetos menores que 100 PF

Tamanho do Projeto	Prazo máximo (em dias úteis)
Até 10 PF	15 dias
De 11 PF a 20 PF	30 dias
De 21 PF a 30 PF	45 dias
De 31 PF a 40 PF	60 dias
De 41 PF a 50 PF	75 dias
De 51 PF a 60 PF	90 dias
De 61 PF a 70 PF	105 dias
De 71 PF a 85 PF	110 dias
De 86 PF a 99 PF	110 dias

Observação: Foi considerada a produtividade de 12 HH/PF, sendo o limite de 110 dias úteis equivalente a 5 meses, que é o resultado da fórmula de Capers Jones para projetos de 100 PF: $Td = 100^{0,35} = 5$ meses. O prazo calculado considera desde a fase de elaboração até a transição.

8.4 ESTIMATIVA DE EQUIPE

Na alocação de equipe, deve-se considerar a estimativa de prazo e de esforço. Sugere-se utilizar a fórmula seguinte:

$$\text{Equipe} = \text{Esforço (HH)} / (21 \times \text{ProdDiária} \times \text{Prazo})$$

Onde:

Prazo = Td em meses

ProdDiária = Recomenda-se considerar 6 horas/dia

21 = dias úteis contidos em 1 mês

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

8.5 GERENCIAMENTO DE PROGRESSO

O acompanhamento do projeto deve identificar o progresso de cada requisito do projeto, ou seja, o percentual de conclusão ao longo do processo de software.

A apuração do percentual concluído em cada macroatividade/fase deve ser definido em comum acordo entre a CAESB e a Fábrica de Software, de acordo com os artefatos entregues. Os artefatos que estão na fábrica, mas não foram entregues, não devem ser considerados nessa apuração.

Segue um exemplo de acompanhamento do progresso do desenvolvimento de um Sistema de Gestão de Projetos, que mostra para cada um dos requisitos o percentual concluído de cada macroatividade:

Tabela 05 - Quadro de Avaliação de Progresso

Requisito	Tamanho	Macroatividade		
		Req.	Impl.	Testes
Caso de Uso 1 Atividade Incluir Ativ. Alterar Ativ Excluir Ativ Consultar Ativ	19 PF	50 %	0 %	10 %
Caso de Uso 2 Relatório de Projetos	5 PF	100 %	50 %	20 %
Caso de Uso ...				

No exemplo acima, a quantidade concluída, em Pontos de Função, do Caso de Uso 2 é:

Tabela 06 - Contagem com Avaliação de Progresso do Caso de Uso 2

Macroatividade	Esforço da Macroatividade	Tamanho	Esforço Realizado	Tamanho Realizado
Requisitos	30 %	1,5 PF	100 %	1,5 PF
Implementação	50 %	2,5 PF	50 %	1,25 PF
Testes	20 %	1,0 PF	20 %	0,2 PF
				Total: 2,95 PF

8.6 MUDANÇA DE REQUISITOS

Os processos de desenvolvimento amplamente utilizados na engenharia de software são baseados no princípio "iterativo-incremental", a citar o processo unificado (UP), o RUP da IBM, o Processo de Software do SISP/MPOG e quase a totalidade das metodologias específicas de cada órgão, entidade ou empresa, não sendo diferente na CAESB.

No processo iterativo, para uma determinada funcionalidade, pode ocorrer mais de um ciclo de trabalho ao longo das atividades de desenvolvimento. Por efeito desses ciclos e refinamento das regras, natural e eventualmente ocorrem mudanças de requisitos que impactam em documentações, análises realizadas e código implementado, gerando aquilo que chamamos de "retrabalho". O retrabalho deve ser bem gerenciado para que não ocorra em excesso, causando impacto negativo no tempo e custo do projeto e violando as vantagens ágeis do processo iterativo, que é comprovadamente a metodologia que reduz o risco de fracasso do projeto.

A medição em Pontos de Função do serviço de retrabalho recebe um capítulo especial no Roteiro de Métricas do SISP, bem como no presente documento, por se tratar de esforço real inerente a projetos de software.

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 [11]. Este fenômeno é denominado *scope creep*.

Nas estimativas iniciais de tamanho de projetos de desenvolvimento, após a fase de iniciação, considerando-se o documento de concepção do projeto, recomenda-se utilizar um percentual de 35% para evolução de requisitos. Por exemplo, suponha que após a análise do documento de concepção de um projeto, aplicou-se a Contagem Estimativa da NESMA e obteve-se o tamanho de 200 PF. Então, o tamanho estimado desse projeto é de 270 PF (200 + 35%). A premissa do projeto deve ser documentada no Termo de Abertura do Projeto (TAP). Nas estimativas, após a fase de elaboração ou requisitos, utilizando-se como insumo as especificações detalhadas de requisitos, recomenda-se utilizar um percentual de 25% para evolução de requisitos.

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 documento, as mudanças de requisitos, ou retrabalhos, serão dimensionados em Pontos de Função, como PF_RETRABALHO, contados de maneira complementar ao projeto de desenvolvimento ou manutenção.

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

A contagem de mudança de requisitos descrita neste documento tem os seguintes pressupostos:

- Deve-se 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, conforme [seção 8.5](#);
- 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:

$$PF_TOTAL = PF_PROJETO + \Sigma PF_RETRABALHO$$

Onde:

PF_PROJETO é a última versão da contagem do escopo do projeto (PF_DESENVOLVIMENTO, PF_MELHORIA, PF_ADAPTATIVA, etc).

A contagem de PF_RETRABALHO leva em conta as seguintes características:

- Requisito original - É o requisito do projeto de desenvolvimento ou de manutenção original, que pode ser incluir, alterar ou excluir funcionalidades de um aplicativo;
- Tipo da mudança do requisito - É a natureza da mudança de requisitos no projeto em andamento, que pode ser acrescentar um requisito, alterar um requisito definido ou desistir de um requisito (retirar do escopo do projeto).

A [Tabela 10](#) resume os percentuais que devem ser aplicados sobre as funções alteradas (considerando o tamanho antes da mudança) para obtenção de PF_RETRABALHO:

Tabela 10 - Percentuais para Mudança de Requisitos

Tipo de Mudança de Requisito	Requisito Original		
	Incluir Função	Alteração Função	Excluir Função
Acréscimo	-	-	-
Alteração (Requisito)	MRI = 50 %	MRA = 50 %	-
Alteração (Interface)	MRI = 0,6 PF	MRA = 0,6 PF	-
Desistência	DI = 120 %	DA = 70 %	DE = 20 %

A seguir são descritos os tipos de mudança nos projetos.

Tipo 1: Acréscimo de funcionalidades ao escopo do projeto

As mudanças que não tragam impacto aos requisitos originais do projeto, caracterizadas pelo acréscimo de funcionalidades ao escopo do projeto 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 do projeto original.

Tipo 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), conforme apresentado na [seção 6.10](#), considerar o percentual de 20% da contagem de uma função transacional de mais baixa complexidade (3 PF), ou seja 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 de conclusão das macroatividades até o momento da solicitação de mudança de requisitos, conforme descrito na [seção 8.5](#).

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

Exemplo:

Considere que um projeto de melhoria tinha como escopo a alteração de uma EE (complexidade alta - 6 PF), 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 PF + 4 PF = 7 PF

Alteração de EE = 6 PF * 50% = 3 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:

CE original = 3 PF

EE original = 6 PF

PF_RETRABALHO = (3 PF x 50%^{Nota 1} + 6 PF x 50%^{Nota 2}) x 90%^{Nota 3}

PF_RETRABALHO = 4,5 x 90% = 4,05 PF

Nota 1: 50% é o percentual a ser aplicado sobre o tamanho antes da mudança de requisito da função sendo incluída na aplicação, conforme apresentado na [Tabela 10](#) - MRI.

Nota 2: 50% é o percentual a ser aplicado sobre o tamanho antes da mudança de requisito da função sendo alterada na aplicação, conforme apresentado na [Tabela 10](#) - MRA.

Nota 3: Supondo que as funcionalidades registram progresso de 90% ([seção 8.5](#)).

Assim, para fins de gestão e faturamento, o valor do PF_RETRABALHO seria o correspondente a 4,05 PF "cheios".

A contagem de PF_MELHORIA deverá ser atualizada para refletir o aumento da complexidade da CE alterada:

Inclusão de CE alterada e SE = 4 PF + 4 PF = 8 PF

Alteração de EE = 6 PF * 50% = 3 PF

PF_MELHORIA_{v2} = 11 PF

O tamanho total do projeto seria então:

PF_TOTAL = PF_PROJETO + Σ PF_RETRABALHO

PF_TOTAL = 11 PF + 4,05 PF = 15,05 PF

Caso 2: Alteração de interface

Durante a fase de implementação 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 PF^{Nota 4} x 35%^{Nota 5}

PF_RETRABALHO = 0,21 PF

Nota 4: 0,6 PF corresponde a 20% de uma função de baixa complexidade (3PF), independentemente do tamanho da função original antes da sua alteração, conforme apresentado na [Tabela 10](#) - MRA.

Nota 5: Supondo que as funcionalidades registram progresso de 35% ([seção 8.5](#)).

Assim, para fins de gestão e faturamento, o valor do PF_RETRABALHO seria o correspondente a 0,21 PF “cheios”.

Nesse caso de mudança de requisitos com alteração de interface (cosmética), a contagem de PF_MELHORIA do projeto original não sofre alteração, visto que a complexidade da função SE não é alterada.

O tamanho total do projeto seria então:

$$PF_TOTAL = PF_PROJETO + \Sigma PF_RETRABALHO$$

$$PF_TOTAL = 0,6 PF + 0,21 PF = 0,81 PF$$

Tipo 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, conforme apresentado na [Tabela 10](#). Além do trabalho de retirar o que foi requisitado (percentuais definidos na [Tabela 10](#)), deve-se considerar também em PF_RETRABALHO o trabalho realizado (fases ou macroatividades concluídas do processo de desenvolvimento) até o momento da desistência desse requisito. Por fim, o requisito original deve ser removido do PF_PROJETO (PF_DESENVOLVIMENTO, PF_MELHORIA, PF_ADAPTATIVA, etc). Enquadram-se nesta situação somente as desistências de incluir, de alterar ou de excluir funcionalidades que constavam no escopo do projeto.

Quando a mudança no projeto for deixar de incluir uma função, aplica-se o percentual de 120% ao tamanho da função original. Esse valor é resultado da soma do percentual de 100% da inclusão (escopo original) com os 20% correspondentes à exclusão dessa mesma função.

Quando a mudança no projeto for deixar de alterar uma função, aplica-se o percentual de 70% ao tamanho da função original. Esse valor é o resultado da soma do percentual de 50% da alteração (escopo original) com os 20% referentes à exclusão dessa mesma função.

Quando a mudança no projeto for deixar de excluir uma função, aplica-se apenas o percentual de 20% referente à exclusão da função original.

Em todos os casos, a contagem de PF_PROJETO deve ser atualizada removendo-se as funções que não fazem mais parte do escopo do projeto.

Da mesma forma que no item 2 (Alteração de função), para fins de gestão e faturamento, sobre a quantidade de PF_RETRABALHO deverá ser aplicado o percentual de conclusão das macroatividades até o momento da solicitação de mudança de requisitos, conforme descrito na [seção 8.5](#).

Exemplos:

Desistência de incluir função

Suponha que um projeto de melhoria para a criação do relatório XPTO, contado como uma SE de complexidade média com 5 PF, teve uma demanda de exclusão do projeto de melhoria durante a 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

$$PF_RETRABALHO = 5 PF \times 120\%^{Nota\ 6} \times 35\%^{Nota\ 7} = 2,1 PF$$

Nota 6: 120% é o percentual a ser aplicado sobre o tamanho antes da desistência da função sendo incluída na aplicação, conforme apresentado na [Tabela 10 - DI](#).

Nota 7: Supondo que a funcionalidade registra progresso de 35% ([seção 8.5](#)).

A contagem de PF_MELHORIA do projeto deve ser atualizada para que o relatório XPTO deixe de constar na medição, conforme fórmula abaixo:

Inclusão de SE = 5 PF

$$PF_MELHORIA_{v2} = PF_MELHORIA_{v1} - (\text{Inclusão de SE})$$

$$PF_MELHORIA_{v2} = PF_MELHORIA_{v1} - 5 PF$$

Desistência de alterar função

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

SE original = 5 PF

$$PF_RETRABALHO = 5 PF \times 70\%^{Nota\ 8} \times 35\%^{Nota\ 9} = 1,225 PF$$

Nota 8: 70% é o percentual a ser aplicado sobre o tamanho antes da desistência da função sendo alterada na aplicação, conforme apresentado na [Tabela 10 - DA](#).

Nota 9: Supondo que a funcionalidade registra progresso de 35% ([seção 8.5](#)).

A contagem de PF_MELHORIA do projeto deve ser atualizada para que o requisito original (alteração do relatório XPTO, contado como uma SE de complexidade média com 5 PF) deixe de constar na medição.

Alteração de SE = 5 PF * 50% = 2,5 PF

$$PF_MELHORIA_{v2} = PF_MELHORIA_{v1} - (\text{Alteração de SE})$$

$$PF_MELHORIA_{v2} = PF_MELHORIA_{v1} - 2,5 PF$$

Tipo 4: *Desistência de alterar uma função seguida de exclusão da função*

Quando a solicitação de mudança seja não só deixar de fazer o que estava no projeto original, mas também excluir duas funções consecutivas:

- 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;
- Conta-se o acréscimo ao escopo do projeto (excluir a função da aplicação) conforme descrito no item 1 (Acréscimo ao escopo do projeto), atualizando-se PF_MELHORIA.

A seguir são apresentadas algumas recomendações e orientações para a gestão de Mudança de Requisitos:

- Não há implicação que para cada Sprint mede-se retrabalho.
- Mudança de conjuntura, como alteração de responsável negocial, reestruturação organizacional, mudanças de legislação, decisões judiciais ou inclusão de novos *stakeholders* e de processos de trabalho, fundamentam, geralmente, a Mudança de Requisito e prosseguimento com nova versão da especificação do projeto.
- Caso o cliente (*stakeholder*) informe regras de que diferem de informações passadas por ele anteriormente e isso impacte na documentação e análise realizada, considera-se motivação para prosseguimento com nova versão da especificação do projeto (Mudança de Requisito). Deve-se prezar pela boa qualidade da elicitação dos requisitos visando minimizar esse cenário.
- Caso foram identificadas novas regras ou alteração de regras após a aprovação da especificação do projeto pela CAESB, recomenda-se prosseguir com nova versão da especificação do projeto, cenário esse tratado como Mudança de Requisito.
 - Caso iniciar a implementação antes da aprovação da especificação pela CAESB, a Fábrica de Software assumirá o ônus de eventuais mudanças de requisitos que impactam o código, sendo que essas alterações não implicarão tacitamente retrabalho remunerável.
- Se a Mudança de Requisito ocorrer após conclusão de todas as macroatividades pela Fábrica de Software, recomenda-se considerar nova Ordem de Serviço. Excetuam-se a este caso os cenários em que a mudança se restringe a pequenos ajustes no ato da homologação pelo usuário e esses ajustes são pouco significativos e aumentam em menos de 10% o tamanho ou esforço da demanda, conforme TR.

9 CONTAGEM COM MÚLTIPLAS MÍDIAS

Este capítulo tem como propósito apresentar as diretrizes de contagem de Pontos de Função em relação ao tema Múltiplas Mídias, cuja abordagem é reconhecida pelo IFPUG. As definições apresentadas têm como base o artigo “Considerations for Counting with Multiple Media” Release 1.1 publicado pelo IFPUG [12].

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* [1].

A determinação da contagem de PF seguindo a abordagem *multiple instance* ou *single instance* depende da avaliação da PRT/CAESB.

A seguir são descritos os termos comuns definidos pelo IFPUG [12]:

- Canal - Também se refere 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. 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. 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 de múltiplas mídias. O entendimento dos exemplos a seguir facilitará o entendimento de outros cenários envolvendo múltiplas mídias.

Cenário 1: Mesmos dados apresentados em tela e impressos

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

Nesses casos, sugere-se a abordagem *single instance*, considerando que dados idênticos sendo apresentados em tela e em relatório impresso devem ser contados como uma única função. Caso as lógicas de processamento da consulta em tela e do relatório em papel sejam distintas, o processo elementar não é único e, portanto, a funcionalidade será contada duas vezes (*multiple instance*). Neste caso, duas funções são contadas: apresentação de dados em tela e apresentação de dados impressos.

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, sugere-se a utilização da abordagem *single instance* considerando que os dados impressos e os dados apresentados no arquivo de saída sejam idênticos e que a ferramenta de desenvolvimento apoie a geração dessas múltiplas saídas. Assim, apenas uma funcionalidade será incluída na contagem de Pontos de Função. Caso as lógicas de processamento da geração do arquivo de saída e do relatório em papel sejam distintas, o processo elementar não é único e, portanto, a funcionalidade será contada duas vezes. Além disso, se a geração das múltiplas saídas não seguirem o padrão da ferramenta de desenvolvimento e tiverem que ser customizadas para o cliente, então será utilizada a abordagem *multiple instance*.

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*. Sugere-se a utilização da abordagem *multiple instance* se a lógica de processamento utilizada nas validações em modo *batch* é diferente da lógica de processamento das validações nas entradas de dados *on-line*. Nesse caso, conta-se duas funcionalidades: a entrada de dados *batch* e a entrada de dados *on-line*.

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, sugere-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. Vale ressaltar que, caso o projeto seja claro o suficiente para dizer que o desenvolvimento é o mesmo, poderá ser utilizada a abordagem *single instance*.

Cenário 5: Relatório em múltiplos formatos

Um relatório deve ser entregue em diferentes formatos, por exemplo: um arquivo com hipertexto (.html) e um arquivo com valores separados por vírgula (.csv).

Nestes casos, conforme sugerido na abordagem *multiple instance*, considera-se a ferramenta utilizada na geração dos relatórios. Se a equipe de desenvolvimento precisar desenvolver o relatório nos dois formatos na ferramenta em questão, serão contadas duas funcionalidades. No entanto, se a ferramenta de desenvolvimento suportar um gerador de relatórios que permita ao usuário visualizar o relatório em tela e, além disso, permita imprimir o relatório, salvar em hipertexto ou salvar no formato de valores separados por vírgula, então a funcionalidade será contada apenas uma vez. Atente-se para o fato de que, neste último caso, a funcionalidade pertence à ferramenta e, não, à aplicação.

10 CONSIDERAÇÕES SOBRE DADOS DE CÓDIGO

No CPM, “Dados de Código” são entendidos como respostas técnicas a requisitos não funcionais, não sendo especificados diretamente pelo usuário. São casos, por exemplo, de ID/Descrição ou dados de substituição, usados internamente no sistema. Os Dados de Código, independentemente se implementados no código-fonte da aplicação, no banco de dados, através de tabelas de apoio, ou em ambos, seja de maneira componentizada ou não, tratam de requisitos técnicos para melhorar a manutenibilidade da aplicação.

Entretanto, em algumas situações, por necessidade do usuário, os Dados de Código possuem manutenção por processos elementares da aplicação. Neste caso, os Dados de Código serão doravante tratados como Dados Negociais.

Nesse contexto, o presente documento assim estabelece:

- Em projetos de desenvolvimento e manutenção, como melhorias, mudanças adaptativas e redesevolvimentos, os Dados de Código, se não mantidos por processos elementares da aplicação, não serão considerados na contagem de Pontos de Função.
 - Quando mantidos por processos elementares da aplicação, os Dados de Código, então chamados de Dados Negociais, serão tratados como Arquivo Lógico Interno ou Registro Lógico, observando-se as regras de contagem de Funções de dados do CPM. As funções transacionais de manutenção desses dados serão mensuradas normalmente nos projetos de desenvolvimento e manutenção.

Entende-se que as funcionalidades de manutenção dos Dados de Negociais, uma vez especificadas, aprovadas, implementadas e testadas, devem ser mensuradas como Funções do CPM, pois trata de serviço técnico efetivamente executado.

- As alterações de domínio de Dados de Código serão apuradas conforme a fórmula do serviço S14, [subseção 6.14 - caso 1](#), que será aplicada para cada registro incluído, alterado ou

excluído. A atualização dos Dados de Código deverá ser efetuada na aplicação e/ou banco de dados conforme necessário.

- Cargas iniciais de Dados de Código, quando os dados não atravessam a fronteira, não serão apuradas em Pontos de Função e, portanto, não se aplica o tipo de serviço S14, vez que não se enquadram como “Conversão de Dados” pelo CPM 4.3 [1]. Essas cargas são partes integrantes do projeto de desenvolvimento ou manutenção.

No caso de sistema legado em que a tabela de código está replicada nas várias funções transacionais, de maneira *hardcoded*, a CAESB poderá, eventualmente, solicitar manutenção adaptativa para adequação técnica à arquitetura de software vigente e às boas práticas, como, por exemplo, implementar a tabela de código na forma reusável. Neste cenário, recomenda-se considerar serviço S11 para as funções impactadas e, a partir da manutenção subsequente, tratar como serviço S14 – atualização simples.

As definições descritas neste capítulo são fundamentadas na experiência, realidade e base histórica de serviços de software da CAESB, observando os princípios adotados no CPM e considerando referências de mercado, em especial o Guia de Contagem do SERPRO [13]. Cabe ressaltar que o Roteiro de Métricas do SISP 2.1 [2] não estabelece diretrizes para o tratamento de Dados de Código, ficando essa incumbência à contratante.

11 ORDENS DE SERVIÇO COMPOSTAS

Uma Ordem de Serviço encaminhada para a Fábrica de Software poderá conter mais de um serviço vinculado, dentre aqueles classificados no [capítulo 6](#). A Ordem de Serviço deverá ser composta por múltiplos serviços somente quando esses serviços forem necessários para atender um mesmo propósito ou projeto.

Algumas situações possíveis são:

- S01 + S14 para desenvolvimento com atualizações dados (ex: cargas e conversões)
- S01 + S03 para desenvolvimento com migração de dados
- S02 + S14 para melhoria com atualizações dados
- S02 + S03 para melhoria com migração de dados
- S02 + S16 para melhoria com redocumentação
- S02 + S16 + S14 para melhoria com redoc. e atualizações dados
- S02 + S16 + S03 para melhoria com redoc. e migração de dados
- S03 + S12 para migração de dados com relatórios de levantamento e/ou verificação
- S04 + S14 para correção de sistema e de dados
- S05 + S14 para mudança (linguagem) com atualizações de dados
- S05 + S03 para mudança (linguagem) com migração de dados
- S06 + S14 para mudança (banco) atualizações de dados
- S06 + S03 para mudança (banco) migração de dados
- S12 + S14 para atualizações dados com relatórios de levantamento e/ou verificação
- S12 + S14 + S02 para levantamento de dados seguido de atuali. de dados e de sistema
- S12 + S14 + S04 para levantamento de dados seguido de correção de dados e de sistema

Outras combinações devem ser utilizadas com cautela, como nos exemplos abaixo, vez que uma mesma Função, em determinado projeto, não pode ser contada em mais de um tipo de serviço, conforme as diretrizes apresentadas nos capítulos anteriores. Exemplos:

- S02 + S10 + S11 para melhoria com adapt. técnicas ou cosméticas em algumas funções
- S02 + S10 para melhoria com ajustes cosméticos em determinadas funções
- S02 + S11 para melhoria com adaptações técnicas em determinadas funções

Nos exemplos acima, se determinada função transacional possuir melhorias funcionais, não funcionais e cosméticas, ela deverá ser contada apenas como S02, que prevalece nesse caso, vez que a melhoria contempla todos esses aspectos. Se houver apenas adaptações não funcionais e cosméticas, prevalece a contagem como tipo de serviço S11.

Existem combinações de serviços que são incoerentes e não deveriam ocorrer, salvo sob contexto específico justificado. A citar:

- S03 + S14, pois a migração de dados contempla a conversão prevista em S14
- S07 + S16, pois nas fórmulas de S07 se inclui a contagem do esforço de Requisitos
- S07 + S18, pois nas fórmulas de S07 se inclui a contagem do esforço de Teste
- Idem para S08 e S09

Manutenção com Redocumentação

No caso em que for demandado "melhoria com redocumentação", recomenda-se adotar Ordem de Serviço composta "S02 + S16", tal que:

- S16 visa contemplar a engenharia reversa para geração ou atualização da documentação, cujo artefato resultando seria uma especificação completa (*baseline* da funcionalidade) contendo, ao mínimo, as regras mantidas e as novas. A parcela da apuração da Ordem de Serviço relativa ao serviço S16 seria $0,3 * PF$, em que 0,3 refere-se ao esforço da macroatividade de Requisitos.
- S02 visa contemplar a aplicação das mudanças e os testes. Ou seja, se aplicaríamos as macroatividades de Implementação e Teste, porém não de Requisitos. A apuração seria, portanto, $(0,5 + 0,2) * PF * FI$, em que 0,5 e 0,2 referem-se aos esforços das macroatividades e FI refere-se ao Fator de Impacto.

Para fator de 50% (mesma empresa), tem-se que a medição total da Ordem de Serviço fica valerá $0,3 * PF + 0,35 * PF = 0,65 * PF$. Observe que, ao final, obtém-se o equivalente ao adicional de 15% previsto no Roteiro do SISP para demandas com redocumentação.

Em projetos de manutenção para os quais há documentação prévia completa das funcionalidades, então não se recomenda tratar a atualização da documentação como S16, ou seja, sem fator de impacto. Contudo, é suficiente, em geral, que seja produzida a documentação relativa ao projeto de melhoria ou adaptação, salvo se requisitado diferente pela CAESB.

Estende-se a formulação acima para todos os casos de manutenção com redocumentação, a citar:

- S04 + S16 para corretiva com redocumentação
- S06 + S16 para mudança de plataforma com redocumentação
- S07 ou S08 ou S19 + S16 para atualização de versão com redocumentação
- S11 + S16 para adaptativa com redocumentação
- S19 + S16 para CIR com redocumentação

Atualizações de Versão que envolvem mais de um aspecto

Mantendo-se consistência com a abordagem do SISP e visando a economicidade para a Administração Pública, quando um projeto englobar dois ou mais tipos de serviços dentre atualização de versão de linguagem (S07), browser (S08) e banco (S09), o escopo do PF_ALTERADO será composto das funções transacionais que necessitarem de adaptação técnica em função de um ou mais desses serviços, de forma que será contada **apenas uma vez** as funções cujas adaptações decorram da atualização de dois ou mais aspectos dentre linguagem, browser e banco.

Exemplo: Identificou-se na CAESB um cenário específico em que a atualização do banco SQL2000 para SQL2012 requer a atualização da linguagem Innovation 7 para 8. A necessidade principal reside na atualização do banco, considerando que a versão SQL2000 foi descontinuada, não possuindo suporte pelo fabricante ou fornecedor, além de outras implicações de infraestrutura. Ocorre que é necessária a atualização do Innovation 7 para 8 em conjunto com a versão do SGBD, como etapas de uma única demanda, vez que a compatibilidade com SQL2012 foi introduzida no Innovation versão 8. Observe que a demanda é composta (S07 + S09) e não faria sentido realizar dois projetos separados, pois o sistema não funcionará com apenas uma das etapas.

12 REFERÊNCIAS BIBLIOGRÁFICAS

[1] International Function Point Users Group. Counting Practices Manual (CPM). 4.3 (2009)

[2] Ministério do Planejamento, Orçamento e Gestão. Roteiro de Métricas de Software do SISP. 2.1 (2015)

[3] Dekkers, Carol. "Measuring the "logical" or "functional" Size of Software Projects and Software Application." Spotlight Software (2003): 10-13.

[4] Institute of Electrical and Electronics Engineers. IEEE Standard for Software Maintenance. Std 1219 (1998)

[5] NESMA. Function Point Analysis for Software Enhancement Guidelines. Version 2.2.1 (2009)

[6] Jones, T. Capers. Estimating software costs. McGraw-Hill, Inc. (2007)

[7] Hazan, Cláudia. "Análise de Pontos de Função: Uma aplicação nas estimativas de tamanho de Projetos de Software." Engenharia de Software Magazine, Edição 2 (2008): 25-31.

[8] Parthasarathy, M. A. Practical software estimation n: function point methods for insourced and outsourced projects. Addison Wesley, New York (2007)

- [9] Vazquez, Carlos E., et al. "Análise de Pontos de Função: Medição, Estimativas e Gerenciamento de Projetos de Software." Editora Érica, São Paulo (2012)
- [10] Boehm, Barry W., et al. Software cost estimation with Cocomo II. Prentice Hall, New Jersey (2000)
- [11] Sommerville, I. Software Engineering. Pearson Education Inc., 9th Edition (2010)
- [12] International Function Point Users Group. Considerations for Counting with Multiple Media. Release 1.1 (2010)
- [13] Serviço Federal de Processamento de Dados (SERPRO). Roteiro SERPRO de Contagem de Pontos de Função e Estimativas versão 7.0 (2015)

