

Guia de Contagem



Última atualização em: 14/05/2020

PROCERGS

Praça dos Açorianos, s/n° - CEP 90010-340
Porto Alegre, RS
0 -XX - 51 - 3210-3100
<http://www.procergs.com.br>

Sumário

1 Apresentação	4
Sobre este documento.....	4
Objetivos.....	4
Público-alvo	4
Demanda	4
Identificação de versão	4
Responsáveis.....	4
Revisores.....	4
Sobre o assunto	4
2 Orientações	5
2.1 Fator de Ajuste	5
2.2 Nível de Detalhamento de Contagens	5
2.3 Metodologia de Desenvolvimento de Sistemas.....	6
3 Contagem Projetos / Manutenção.....	7
3.1 Escopo da contagem e Fronteira da Aplicação.....	7
3.2 Contagem referente ao SOE.....	8
3.3 Projeto de Melhoria.....	9
3.3.1 Função Alterada.....	11
3.4 Ciclo Preditivo	12
3.5 Ciclo Ágil.....	12
3.5.1 Refinamento da funcionalidade.....	12
3.5.2 Mudança de requisitos	13
3.5.3 Exemplo de aplicação	13
3.6 Atualização de Linguagem de Programação e BD.....	16
3.6.1 Mudança de Plataforma	16
3.6.2 Atualização de Versão	17
3.7 Contagem de Múltiplas Mídias	17
3.7.1 Mesmos dados apresentados em tela e impressos.....	18
3.7.2 Mesmos dados de saída como dados em arquivo e relatório impresso	19
3.7.3 Mesmos dados de entrada batch e on-line.....	19
3.7.4 Múltiplos canais de entrega da mesma funcionalidade	19
3.7.5 Relatórios em Múltiplos Formatos.....	19
3.8 Subdivisão de funcionalidades (abas / telas).....	20
3.8.1 Funcionalidade com etapas na forma de telas encadeadas.....	20
3.8.2 Funcionalidade com abas.....	20
3.8.3 Subdivisão em telas ou abas, com a possibilidade de salvar rascunho.....	20
3.8.4 Subdivisão em telas ou abas, salvando versão final em passo específico	21
3.8.5 Funcionalidade subdividida para atender a uma necessidade do negócio	21
3.9 Dados Histórico e Auditoria.....	21
3.9.1 Dados históricos.....	21
3.9.2 Trilha de auditoria.....	21
3.10 Contagem de Middleware	22
3.10.1 Aplicações Middleware.....	22
3.10.2 Aplicações que disponibilizam Webservices	22
3.11 Contagem para Manutenção Corretiva.....	23
3.12 Processos Batch.....	24

3.13	Manutenção em Interface	26
3.14	Manutenção Adaptativa em Requisitos não Funcionais	26
3.15	Manutenção em Dados de Código	27
3.16	Atualização de Dados de Domínio	28
3.17	Páginas Estáticas e Funcionalidades Multi Idiomas	29
3.18	Manutenção de Componente Reutilizável	29
3.19	Atualização de Versão – Browser	30
3.20	Pontos de Função de Teste	30
3.21	Atualização de Manual do Usuário e Help Online	31
3.22	Dicas para Otimizar o Custo das Manutenções	32
4	Contagem de Processos Automatizados	34
4.1	Recebimento de dados de outras aplicações utilizando interface de integração	34
4.2	Processo Elementar	35
5	Contagem de Sites	36
5.1	Definições	36
5.2	Funções de Dados	37
5.3	Tipos de Funções Transacionais	37
5.4	Complexidade Funcional	38
6	Bibliografia Consultada	39

1 Apresentação

Sobre este documento

Objetivos

Tornar mais prático o uso dos conceitos e regras definidos pelo IFPUG, exemplificando situações específicas da PROCERGS e definindo situações em que o IFPUG não oferece orientação.

Público-alvo

Equipe de Desenvolvimento de Sistemas e Analistas de Negócios.

Demanda

Orientar os desenvolvedores na contagem de APF.

Identificação de versão

Versão 1.0 – Versão Inicial (Data: 12/08/2011)

Versão 2.0 – Alterações referentes ao Edital de Fábrica de Software de Sistemas (Data: 13/06/2013)

Versão 2.1 – Alterações referente a Projeto de Melhoria, contagem do Ciclo Ágil, Middleware e Webservices (10/10/2017).

Versão 2.2 – Alterações para adequar ao edital (2018).

Versão 2.3 – Inclusão de orientações sobre Fronteira da Aplicação.

Versão 3.0 – Inclusão de orientações sobre o SOE, subdivisão de funcionalidades, dados de histórico e auditoria.

Responsáveis

Lourdes Helene Liebstein

Revisores

Lizandro Kirst da Silva

Marcelo Carpes

Sobre o assunto

Este documento tem a finalidade de facilitar o uso da técnica de análise de pontos de função dentro do contexto de desenvolvimento de Sistemas e *Sites* da PROCERGS.

Este guia funciona como um complemento ao Manual de Práticas de Contagem de Pontos de Função Versão 4.3.1 (*Function Point Counting Practices Manual* Release 4.3.1), publicado pelo IFPUG (*International Function Point Users Group*).

2 Orientações

2.1 Fator de Ajuste

A PROCERGS não utiliza a avaliação do valor do fator de ajuste em suas medições, sendo este fixado no valor 1 (um). Desta forma, as características gerais do sistema, apresentadas no manual de práticas de contagem do IFPUG não afetam o tamanho dos Pontos de Função Não Ajustados, obtidos em suas contagens. O dimensionamento das demandas considera portanto, o tamanho funcional obtido através das funções de dados e de transações.

2.2 Nível de Detalhamento de Contagens

O nível de detalhamento a ser escolhido depende de alguns fatores como a finalidade da contagem, as informações disponíveis para subsidiar a contagem, a etapa do ciclo de vida do desenvolvimento, dentre outros.

A contagem indicativa e estimada dar-se-á conforme definições da Netherlands Software Metrics Association – NESMA.

A orientação é que a contagem estimada seja utilizada para embasar o planejamento nos projetos de desenvolvimento e manutenção. Quanto à contagem detalhada, a orientação é que seja utilizada após a homologação e o aceite dos produtos gerados.

É importante destacar que, quanto mais exata se quer uma contagem de pontos de função, mais detalhados devem ser os requisitos do usuário.

Contagem Estimada

A contagem estimada é utilizada quando é possível identificar as funções do sistema, porém não se define a complexidade (tipos de dados, tipos de registros e arquivos referenciados).

Dessa forma, assume-se uma complexidade padrão para as funções, sendo as funções de dados (ALIs e AIEs) classificadas como de baixa complexidade, enquanto as funções transacionais (EEs, CEs e SEs) são classificadas como de média complexidade.

A contagem estimada é realizada da seguinte forma:

- determina-se todas as funções de todos os tipos (ALI, AIE, EE, SE, CE);
- toda função do tipo dado (ALI, AIE) tem sua complexidade funcional avaliada como Baixa;
- toda função transacional (EE, SE, CE) é avaliada como de complexidade média;
- calcula-se o total de pontos de função não ajustados.

Contagem Detalhada

A contagem detalhada dar-se-á pela aplicação da técnica de contagem seguindo as regras estabelecidas no Manual de Práticas de Contagem (CPM) versão 4.3.1 (ou superior) do IFPUG, acrescidas das definições deste Guia de Contagem. A contagem detalhada é a contagem usual de pontos de função e é realizada da seguinte forma:

- determina-se todas as funções de todos os tipos (ALI, AIE, EE, SE, CE);

- determina-se a complexidade de cada função (Baixa, Média, Alta);
- calcula-se o total de pontos de função não ajustados.

2.3 Metodologia de Desenvolvimento de Sistemas

O processo de desenvolvimento da PROCERGS possui etapas bem definidas e que podem ser gerenciadas utilizando um **Ciclo Preditivo** ou através de um **Ciclo Ágil**.

O Ciclo **Preditivo** possui as seguintes características:

- Planejamento antecipado;
- Entregas por módulos ou pacotes de entregas acordados com o cliente. Cada entrega é um subconjunto do produto final em desenvolvimento e cresce de modo incremental;
- A implementação de um módulo ou pacote passa por todas as etapas do ciclo de desenvolvimento: Análise, Projeto, Construção, Implantação;
- Se ocorrerem mudanças o prazo é estendido;
- As prioridades são definidas pelo Gerente do Projeto baseando-se nas necessidades do cliente.

O Ciclo **Ágil** tem as seguintes características:

- Foco no valor do negócio;
- Planejamento contínuo e adaptativo;
- Cliente faz parte da equipe e prioridades são definidas pela equipe;
- Pequenas entregas com prazos rígidos.

A cada projeto a equipe de desenvolvimento da PROCERGS define o ciclo de desenvolvimento que será utilizado (Preditivo ou Ágil).

3 Contagem Projetos / Manutenção

3.1 Escopo da contagem e Fronteira da Aplicação

Segundo o CPM, o escopo da contagem define o conjunto de requisitos funcionais de usuários para ser incluídos na contagem. O escopo:

- define o (sub)conjunto do software que está sendo medido;
- é determinado pelo propósito para a realização da contagem de pontos de função;
- identifica quais funções serão incluídas na medida de tamanho funcional assim como fornecer respostas relevantes para o propósito da contagem;
- pode incluir mais de uma aplicação.

A **fronteira da aplicação** pode ser entendida como a interface conceitual que delimita o software que será medido e seus usuários. O posicionamento incorreto da fronteira pode alterar a perspectiva da medição de uma visão lógica (visão funcional) para uma visão física. As principais consequências disso são a contagem duplicada de transações e arquivos de dados, a contagem incorreta de funções de transferência de dados e dificuldade na contagem de arquivos. Uma fronteira de aplicação não pode ser subdividida por contextos gerenciais de desenvolvimento, por exemplo, interno e externo ao órgão.

As seguintes regras devem ser aplicadas para fronteiras:

- a fronteira é determinada com base na visão do usuário. O foco está no que o usuário pode entender e descrever;
- a fronteira entre aplicações relacionadas está baseada nas áreas funcionais separadas como pode ser visto pelo usuário, não em considerações técnicas;
- a fronteira inicial já estabelecida para a aplicação ou aplicações que estejam sendo modificadas não é influenciada pelo escopo da contagem.

Notas: Pode haver mais de uma aplicação incluída no escopo da contagem. Nesse caso, múltiplas fronteiras da aplicação deverão ser identificadas. Quando a fronteira não está bem definida (como no início da análise), ela deverá ser posicionada da forma mais exata possível.

Diante de algum impasse na definição da fronteira, deve prevalecer a visão de negócio do usuário, uma vez que cabe ao cliente ser o detentor do conhecimento do negócio e, como a fronteira impacta diretamente no resultado da contagem e conseqüentemente em custos.

O conceito de fronteira é tão importante que o ideal é que se estabeleça previamente um inventário das aplicações da organização. Todas as contagens de pontos de função realizadas devem usar como base as fronteiras estabelecidas previamente. A fronteira é um conceito que dificilmente muda entre as contagens de pontos de função. Ou seja, as diversas medições dos projetos de melhoria na aplicação, terão como base a mesma fronteira.

Em projetos de melhoria, a fronteira estabelecida no início do projeto deve estar de acordo com a fronteira já estabelecida para a aplicação que está sendo modificada.

Orientações práticas para identificação de fronteiras:

1. Obter no Documento de Projeto a definição dos principais processos automatizados pelo sistema ou afetados pelo projeto de melhoria.
2. Analisar o fluxo de dados da aplicação. Obtenha as informações no documento de projeto, no capítulo “Definição do Escopo” nos itens “Requisitos Funcionais” e “Sistemas que interagem”. Verifique se existe Mapeamento de Processos e Diagrama de Contexto para o sistema.
3. Verificar como os grupos de dados são mantidos: diferentes módulos mantendo os mesmos grupos de dados podem indicar que estes estão contidos na mesma fronteira.
4. Identificar os gestores do negócio (especialistas no assunto ou seus representantes) e verificar a visão deles para o software em análise. As fronteiras não são baseadas em decisões técnicas.
5. A fronteira inicial estabelecida não é influenciada pelo escopo do projeto.
6. Procure identificar o propósito de negócio de cada aplicação/sistema/módulo. É possível que um sistema de gestão integrado implemente funcionalidades de mais de uma fronteira, considerando que cada um terá propósitos de negócio diferentes.
7. Levantar em consideração que perfis de usuário que vão utilizar o sistema, distribuição das equipes de desenvolvimento e definições arquiteturais e/ou tecnológicas não são insumos para a definição da fronteira.
8. Havendo dificuldade para identificar a fronteira, entreviste os especialistas no assunto para auxiliar na identificação.

3.2 Contagem referente ao SOE

O Sistema SOE (Controle de Acesso e Segurança) tem como objetivo permitir o controle de acesso às aplicações desenvolvidas na PROCERGS, além da administração das autorizações dadas aos usuários. Este controle de acesso e as autorizações estão baseados na criação de usuários e na definição de suas permissões em cada aplicação. As aplicações são desenvolvidas em diferentes plataformas (IBM – Natural, Web – Asp/Java/Jsp e Windows – VB Cliente/servidor). Para cada ambiente, existem componentes do SOE específicos para garantir o acesso e segurança.

Definições:

■ **Requisitos Não Funcionais** (Glossário da Análise de Pontos de Função):

Requisitos do usuário que não constituem Requisitos Funcionais do Usuário incluem, mas não estão limitados aos seguintes:

- restrições de qualidade (por exemplo: usabilidade, confiabilidade, eficiência e portabilidade);
- restrições organizacionais (por exemplo: locais para operação, hardware-alvo e conformidade com os padrões);

- restrições ambientais (por exemplo: interoperabilidade, segurança e privacidade);
 - restrições de implementação (por exemplo: linguagem de desenvolvimento, prazo para entrega).
- [ISO/IEC 14143-1:2007, definition 3.8]

■ Segurança de acesso:

Capacidade do produto de software de proteger informações e dados, de forma que pessoas ou sistemas não autorizados não possam lê-los nem modificá-los e que não seja negado o acesso às pessoas ou sistemas autorizados.

[NBR ISO/IEC 9126-1 – Modelo de Qualidade]

Como Contar: na contagem de um aplicativo que utiliza o SOE, consideramos a funcionalidade de Login como uma CE/SE, conforme demais requisitos funcionais da aplicação e os arquivos referenciados do SOE como arquivos de interface externa.

Conforme definição de **requisitos não funcionais** no que se referem ao controle de acesso geral à aplicação, é que apenas conta-se o usuário/perfil como um arquivo referenciado se o seu uso for específico de uma funcionalidade (e nela os arquivos referenciados seriam somados). O controle de acesso envolve toda a aplicação e não uma ou outra transação em particular.

Exemplos de uso específico:

- um campo somente será visível se o usuário for de um perfil definido;
- o conjunto de registros recuperados será diferente dos demais se o usuário for de um perfil definido;
- somente será impresso os campos salário e índice de aumento se quem solicitou o relatório for o usuário de um perfil de gestão.

3.3 Projeto de Melhoria

O projeto de Melhoria também denominado de projeto de melhoria funcional ou manutenção evolutiva, está associado às mudanças em requisitos funcionais e não funcionais da aplicação, ou seja, a inclusão de novas funcionalidades, alteração de funcionalidades ou exclusão de funcionalidades em **aplicações implantadas**.

Além de manutenções em funcionalidades que tenham sido implantadas, também poderão ocorrer **mudanças de requisitos (retrabalhos)** no decorrer do projeto, conforme o cliente adquira mais conhecimento sobre o negócio. Estas mudanças **também serão consideradas Melhorias** e para o cálculo do tamanho será considerado o **percentual de esforço relativo às etapas contratadas já concluídas e aprovadas**.

A Análise de Pontos de Função (APF), conforme definido no Manual de Práticas de Contagem do IFPUG, não faz nenhum tipo de diferenciação com relação ao tipo de mudança realizada sobre a funcionalidade (inclusão, alteração ou exclusão) em um projeto de melhoria. A APF não mede o impacto da mudança na função, apenas que a função será alterada e o tamanho que a mesma terá após a alteração.

No entanto, ao se utilizar a técnica para realização pagamentos e estimativas de esforço é necessária uma abordagem diferenciada, com intuito de permitir uma relação mais próxima entre o tamanho funcional e o esforço estimado ou remuneração efetuada.

O esforço para construção de uma nova funcionalidade, via de regra, não é o mesmo que o despendido para realização de uma alteração na funcionalidade ou exclusão da mesma. Desta forma, a solução utilizada pela PROCERGS, e que é comum no mercado, é adoção de um **Fator de Impacto (FI)** conforme o tipo de mudança realizada sobre as funcionalidades, diminuindo a quantidade de PFs identificados para as funcionalidades alteradas e excluídas.

As regras de APF serão seguidas para a apuração do tamanho da demanda, antes da aplicação do fator de impacto por tipo de mudança.

Fator de Impacto	
Tipo de Mudança	FI
Alteração para funcionalidade de sistema desenvolvida ou mantida por meio de um projeto de melhoria pela empresa contratada.	0,50
Alteração para funcionalidade de sistema não desenvolvida ou mantida por meio de um projeto de melhoria pela empresa contratada e sem necessidade de redocumentação da funcionalidade.	0,75
Alteração para funcionalidade de sistema não desenvolvida ou mantida por meio de um projeto de melhoria pela empresa contratada e com necessidade de redocumetação da funcionalidade. O percentual de 0,90 apenas será considerado na primeira demanda de projeto de melhoria de cada funcionalidade.	0,90
Exclusão de funcionalidades já existentes	0,25

Entende-se por redocumentação a atividade de engenharia reversa para entendimento do software e elaboração de documentação prevista na etapa de Análise.

O tamanho da manutenção em pontos de função levará em consideração o fator de impacto que será aplicado conforme o tipo de mudança, bem como o percentual de esforço relativo às etapas contratadas já concluídas e aprovadas e que deverão ser atualizadas.

O percentual de esforço por etapa encontra-se definido no contrato estabelecido com a contratada.

Desta forma, para pagamento de contagem de pontos de função de Projetos de Melhoria e Mudanças de Requisitos já concluídos e aprovados (**Retrabalhos**), será utilizada a seguinte fórmula:

$$\text{PF Melhoria} = (\text{PF_INC} + (\text{PF_ALT} \times \text{FI}) + (\text{PF_EXC} \times 0,25)) \times \text{P_ETAPA}$$

Onde:

PF_INC = Total de Pontos de Função Incluídos

PF_ALT = Total de Pontos de Função Alterados

FI = Fator de Impacto, pode variar de 0,5 a 0,9

PF_EXC = Total de Pontos de Função Excluídos

0,25= Fator de Impacto de Exclusão de Funcionalidade

P_ETAPA = Percentual de conclusão do projeto considerando soma dos percentuais das etapas concluídas

3.3.1 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. Também é considerada alterada se algum tipo de dado sofrer mudança de tamanho (número de posições) ou tipo de campo (por exemplo: mudança de numérico ou alfanumérico), caso a mudança decorra de alteração de regra de negócio. Uma função transacional (Entrada Externa, Consulta Externa e Saída Externa) é considerada alterada, quando a alteração contemplar:

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

O CPM 4.3 define **lógica de processamento** como requisitos especificamente solicitados pelo usuário para completar um processo elementar. Esses requisitos devem incluir 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.

Serão tratadas como manutenções adaptativas apenas as manutenções que implicarem exclusivamente em mudanças em requisitos não funcionais. Se uma mesma funcionalidade tiver mudanças em requisitos funcionais e não funcionais, esta deve ser contada apenas uma vez, como função alterada em um projeto de Melhoria.

3.4 Ciclo Preditivo

Os Arquivos Lógicos Internos (ALI) que já foram considerados na contagem de módulos/pacotes de entrega anteriores do projeto e que não foram alterados ou excluídos, não devem ser consideradas na contagem das Funções de Dados do módulo/pacote de entrega em desenvolvimento.

Os Arquivos de Interface Externa (AIE) já considerados na contagem de módulos/pacotes de entrega anteriores e que não foram alterados ou excluídos, não devem ser considerados na contagem de Funções de Dados do módulo/pacote de entrega em desenvolvimento.

Exemplo: Para um Sistema de Gestão de Crédito contratado por módulos, num primeiro módulo o ALI Auditoria foi implementado e, portanto, considerado na contagem das funções de dados. Para um segundo módulo contratado para este mesmo sistema, este arquivo será referenciado pelas transações, mas não sofrerá alteração. Desta forma, o mesmo não será considerado na contagem das funções de dados, será apenas considerado como arquivo referenciado (AR) na contagem das funções de transação.

3.5 Ciclo Ágil

Para projetos que utilizarem Ciclo de Desenvolvimento Ágil, deve ser realizada inicialmente uma contagem estimada, baseada nos requisitos funcionais levantados no início do projeto. Ao final de cada *sprint* deve ser realizada a contagem detalhada das funcionalidades incluídas e das funcionalidades que foram alteradas ou excluídas.

Os projetos devem conter no seu cronograma de planejamento do desenvolvimento as releases e suas respectivas sprints e deve ser aprovado pela contratante. Entende-se que release é um ciclo que perpassa pelas fases do processo de desenvolvimento de software com o objetivo de entregar, ao final do ciclo, um produto pronto. O ciclo da release não deve ser igual ao ciclo da sprint e deve promover aumento do percentual de completude do sistema (entrega de valor agregado ao negócio).

A contagem das funcionalidades alteradas deve considerar o tipo de trabalho realizado:

- refinamento da funcionalidade dentro da release;
- mudança de requisitos.

3.5.1 Refinamento da funcionalidade

- São considerada neste tipo as alterações realizadas em *sprints* posteriores a que a funcionalidade foi criada e que foi previamente planejado para ser desenvolvido de forma particionada (incremental) em uma release ou quaisquer mudanças ocorridas provocadas pelo aprofundamento, detalhamento e complementação de requisitos durante o processo de desenvolvimento.
- Dentro de uma release, as mudanças em uma funcionalidade não serão contadas, devem ser absorvidas pela contratada como parte inerente do processo ágil de desenvolvimento de software.
- Se enquadra neste tipo de trabalho: acrescentar novos campos em uma tela, incluir novos filtros e campos em uma pesquisa/relatório, incluir uma nova aba em uma funcionalidade, incluir novas regras de negócio. Requisitos que estavam previstos, não de forma detalhada, mas macro no escopo inicial do projeto e que foram particionados em mais de uma *sprint*.

- A equipe do projeto deverá registrar de forma macro as alterações previstas para cada funcionalidade no artefato previsto na metodologia de desenvolvimento da PROCERGS. O particionamento da funcionalidade em mais de uma sprint deverá estar documentada na especificação da funcionalidade e representada no cronograma de projeto.
- Exemplo: um cadastro de funcionários que conterà mais de 40 campos foi dividido em 4 abas. Todas as abas fazem parte da mesma funcionalidade. Não foram levantados todos os campos que a funcionalidade completa terá, porém foi planejado quebrar o desenvolvimento desta funcionalidade em 2 sprints, de forma que na primeira sprint seria concluído a funcionalidade com 2 abas e na sprint seguinte as outras duas.

3.5.2 Mudança de requisitos

- São consideradas neste tipo as alterações em funcionalidades em virtude de mudanças de escopo, de regras de negócio, legislação, entre outros fatores e que ocorrem após o término da release na qual essas funcionalidades ficaram prontas.
- As mudanças que ocorrem após o término da release em que as funcionalidades ficaram prontas, devem ser tratadas da mesma forma que um projeto de melhoria, considerando o Fator de Impacto (FI) de acordo com o tipo de mudança e retrabalho realizado.

3.5.3 Exemplo de aplicação

Este exemplo foi extraído do Roteiro de Métricas de Software do SISP (Versão 2.2) e tem como objetivo exemplificar como tratar evolução incremental (refinamentos) ou mudança de requisitos para as funções de dados ou transações.

Suponha o planejamento das iterações (sprints) de uma release (Release N) com quatro funcionalidades a serem desenvolvidas (incluídas) e duas funcionalidades prontas em releases anteriores para serem alteradas.

Planejamento do Backlog das Sprints da Release N

<i>Release N</i> (composta de 3 sprints)	<i>Processo Elementar</i>	<i>Categoria</i> (Inc, Alt, Exc, Refin)	<i>Tipo (ALI, AIE, EE, CE, SE)</i>	<i>Complex.</i>	<i>PF</i>	<i>Observação</i>
Sprint1	Incluir Aluno	Inc	EE	Baixa	3	
	Aluno	Inc	ALI	Baixa	7	
	Incluir Disciplina	Alt	EE	Baixa	1,5	Alteração caracterizada como Projeto de Melhoria (PE desenvolvido e pronto na Release N-1). Aplicado o fator de impacto de 50% (3 PF * 0,5 = 1,5 PF)
Sprint2	Alterar Aluno	Inc	EE	Baixa	3	
	Alterar Disciplina	Alt	EE	Baixa	1,5	Alteração caracterizada como Projeto de Melhoria (PE desenvolvido e pronto na Release N-1). Aplicado o fator de impacto de 50% (3 PF * 0,5 = 1,5 PF)
Sprint3	Emitir Relatório de Alunos por Disciplina	Inc	SE	Média	5	
Total de PFs da Release N					21	

A tabela a seguir apresenta a contagem de pontos de função e o detalhamento dos serviços realizados na execução da Release N e suas sprints, incluindo:

- as funcionalidades planejadas e as mudanças em funcionalidades previamente desenvolvidas na mesma release (refinamentos);
- as funcionalidades alteradas em release anteriores o que caracteriza uma alteração de requisito, sendo tratada da mesma forma que um projeto de melhoria. Destaca-se, que o processo elementar “Alterar Disciplina” foi alterado na sprint 2, sendo essa mudança categorizada como “Alteração” do Projeto de Melhoria. Em seguida na sprint 3 houve nova mudança categorizada como “Refinamento” pois trata-se de uma mudança de funcionalidade já trabalhada na mesma release.

Contagem detalhada de Pontos de Função da Release N

<i>Release N (composta de 3 sprints)</i>	<i>Processo Elementar</i>	<i>Categoria (Inc, Alt, Exc, Refin)</i>	<i>Tipo (ALI, AIE, EE, CE, SE)</i>	<i>Complex.</i>	<i>PF</i>	<i>Observação</i>
Sprint1	Incluir Aluno	Inc	EE	Baixa	3	
	Aluno	Inc	ALI	Baixa	7	
	Incluir Disciplina	Alt	EE	Baixa	1,5	Alteração caracterizada como Projeto de Melhoria (PE desenvolvido e pronto na Release N-1). Aplicado o fator de impacto de 50% (3 PF * 0,5 = 1,5 PF)
Sprint2	Alterar Aluno	Inc	EE	Baixa	3	
	Aluno	Refin	ALI	Baixa	-	Mudança caracterizada como refinamento de funcionalidade para atender o PE “Alterar Aluno”. Sem custo de PF.
	Alterar Disciplina	Alt	EE	Baixa	1,5	Alteração caracterizada como Projeto de Melhoria (PE desenvolvido e pronto na Release N-1). Aplicado o fator de impacto de 50% (3 PF * 0,5 = 1,5 PF)
Sprint3	Emitir Relatório de Alunos por Disciplina	Inc	SE	Média	5	
	Incluir Aluno	Refin	EE	Baixa	-	Mudança caracterizada como refinamento de funcionalidade para atender o PE “Emitir Relatório de Alunos por Disciplina”. Sem custo de PF.
	Incluir Disciplina	Refin	EE	Baixa	-	Mudança caracterizada como refinamento de funcionalidade para atender o PE “Emitir Relatório de Alunos por Disciplina”. Sem custo de PF.

<i>Release N (composta de 3 sprints)</i>	<i>Processo Elementar</i>	<i>Categoria (Inc, Alt, Exc, Refin)</i>	<i>Tipo (ALI, AIE, EE, CE, SE)</i>	<i>Complex.</i>	<i>PF</i>	<i>Observação</i>
	Alterar Disciplina	Refin	EE	Baixa	-	Mudança caracterizada como refinamento de funcionalidade para atender o PE “Emitir Relatório de Alunos por Disciplina”. Sem custo de PF.
	Emitir Relatório de Disciplina	Alt	EE	Baixa	1,5	Alteração caracterizada como Projeto de Melhoria (PE desenvolvido e pronto na Release N-1). Aplicado o fator de impacto de 50% (3 PF * 0,5 = 1,5 PF)
Total de PFs da Release N					22,5	

3.6 Atualização de Linguagem de Programação e BD

3.6.1 Mudança de Plataforma

São considerados nesta categoria projetos que englobam mudança de linguagem de programação e ou banco de dados. Exemplo: Sistema em Cobol sendo migrado para plataforma em Java.

Observações:

- se não houver mudanças nas funções de dados, estas não devem ser consideradas para fins de remuneração. No entanto, deve ser realizada a contagem das funções de dados a fim de compor a contagem final do projeto;
- caso o projeto possua documentação de requisitos, a etapa de Análise não será contratada e será considerado apenas os percentuais das etapas contratadas;
- PF de conversão são os pontos de função associados às funcionalidades de conversão de dados.

Mudança de Linguagem de Programação

Se o projeto a ser migrado, não possuir documentação, deve ser considerado como novo projeto de desenvolvimento.

$$\boxed{\text{PF_MIGRAÇÃO_LING} = \text{PF_INCLUÍDO} + \text{PF CONVERSÃO}}$$

Mudança de Banco de Dados

Em casos de mudança de banco hierárquico para relacional, em sistemas sem documentação de requisitos, será considerado como um projeto de desenvolvimento (considerar as funções de dados e transações).

Observação:

- somente considerar se a contratada for responsável pela mudança de estrutura do banco de dados.

$$\text{PF_MGRACAO_BD} = \text{PF_INCLUÍDO} + \text{PF CONVERSÃO}$$

Se a demanda for de alteração de um sistema gerenciador de banco de dados relacional para outro relacional, utilizar a fórmula abaixo:

$$\text{PF_MIGRACAO_BD_REL} = \text{PF ALTERADO} * 0,30$$

- Deve ser considerado as funcionalidades impactadas e as funcionalidades que apenas possuem demandas de testes, devem ser consideradas utilizando o percentual da fase de testes.

3.6.2 Atualização de Versão

São as demandas de atualização de versão de linguagem de programação de sistemas ou de banco de dados. Utilizar a fórmula:

$$\text{PF_ATUALIZAÇÃO_LING_BD} = \text{PF_ALTERADO} * 0,30$$

- Deve ser considerado as funcionalidades impactadas e as funcionalidades que apenas possuem demandas de testes, devem ser consideradas utilizando o percentual da fase de testes.
- As funções de dados não devem ser contadas.

3.7 Contagem de Múltiplas Mídias

Considerando-se a contagem de Pontos de Função de funcionalidades entregues em mais de uma mídia, a aplicação das regras de contagem de Pontos de Função definidas no CPM 4.3 leva a duas abordagens alternativas: *single instance* e *multiple instance*.

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

O IFPUG reconhece ambas abordagens, *single instance* e *multiple instance*, para a aplicação das regras definidas no CPM. Neste guia será estabelecida a abordagem que será seguida pela PROCERGS em cada caso.

As estimativas e contagens de PF abordadas neste documento serão baseadas em *multiple instance*, com exceção dos casos de consultas em .pdf, .doc, .xls e consultas idênticas em tela e papel, que serão consideradas uma única funcionalidade.

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

- **Canal:** também refere-se a mídia. Múltiplos canais é sinônimo de múltiplas mídias.
- **Mídia:** descreve a maneira que os dados ou informações se movimentam para dentro e para fora de uma fronteira de aplicação, por exemplo, apresentação de dados em tela, impressora, arquivo, voz. Este termo é utilizado para incluir, dentre outros: diferentes plataformas técnicas e formatos de arquivos como diferentes mídias.
- **Múltiplas Mídias:** quando a mesma funcionalidade é entregue em mais de uma mídia. Frequentemente, somente uma mídia é requisitada para um usuário específico em um determinado momento, por exemplo consulta de extrato bancário via internet como oposto a consulta de extrato bancário via terminal do banco.
- **Multi-Mídia:** quando mais de uma mídia é necessária para entregar a função, por exemplo, uma nova notícia publicada na Internet que é apresentada em vídeo e texto. Observe que a notícia completa só é apresentada para o usuário se ele ler o texto e assistir o vídeo.
- **Abordagem *Single Instance*:** esta abordagem não reconhece que a mídia utilizada na entrega da função transacional é uma característica de diferenciação na identificação da unicidade da função transacional. Se duas funções entregam a mesma funcionalidade usando mídias diferentes, elas são consideradas a mesma funcionalidade em uma contagem de Pontos de Função.
- **Abordagem *Multiple Instance*:** esta abordagem especifica que o tamanho funcional é obtido no contexto de objetivo da contagem, permitindo uma função de negócio ser reconhecida no contexto das mídias que são requisitadas para a funcionalidade ser entregue. A abordagem *multiple instance* reconhece que a mídia para entrega constitui uma característica de diferenciação na identificação da unicidade da função transacional.

Os cenários descritos nas seções seguintes não representam uma lista completa de situações de múltiplas mídias. O entendimento destes exemplos facilitará o entendimento de outros cenários envolvendo múltiplas mídias. Este Guia deve ser atualizado considerando a publicação de novas diretrizes do IFPUG, novos cenários que emergirão nas contagens de PFs dos projetos dos órgãos do SISP (Sistema de Administração dos Recursos de Informação e Informática) e nas contagens de projetos a serem contratados pela PROCERGS.

3.7.1 Mesmos dados apresentados em tela e impressos

Nos casos em que uma aplicação apresenta uma informação em uma consulta (tela) e a mesma informação pode ser impressa, caso requisitado pelo usuário na tela em questão, considerando que os dados apresentados e relatório impresso são idênticos, não há diferença na lógica de processamento e esforço adicional para construção e formatação da impressão, a PROCERGS adotará a abordagem *single instance*, ou seja, será contada como uma única função.

Se a impressão de parte da consulta (aba) for idêntica (como um print da tela) e realizada pelo browser/framework, não deve ser contada como um processo elementar.

Caso as lógicas de processamento da consulta em tela e do relatório impresso sejam distintas e casos em que há esforço adicional para construção e formatação da impressão, o processo

elementar não é único e portanto a funcionalidade será contada duas vezes, serão contadas uma função para a apresentação de dados em tela e outra função para apresentação de dados impressos.

3.7.2 Mesmos dados de saída como dados em arquivo e relatório impresso

Quando uma aplicação grava dados em um arquivo de saída e imprime um relatório com informações idênticas as gravadas no arquivo, a PROCERGS adotará a abordagem *single instance*, considerando que os dados impressos e os dados apresentados no arquivo de saída sejam idênticos e que a ferramenta de desenvolvimento apoie a geração dessas múltiplas saídas. Neste caso 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 impresso sejam distintas, o processo elementar não é único e portanto a funcionalidade será contada duas vezes.

Ou ainda nos casos em que 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*. Neste cenário, duas funções são contadas – geração arquivo e apresentação dos dados impressos.

3.7.3 Mesmos dados de entrada batch e on-line

Nos casos em que uma informação pode ser carregada na aplicação por meio de dois métodos arquivo batch e entrada on-line a PROCERGS utilizará a abordagem *multiple instance* que conta duas funcionalidades: a entrada de dados batch e a entrada de dados on-line.

3.7.4 Múltiplos canais de entrega da mesma funcionalidade

Nos casos em que uma funcionalidade deve ser disponibilizada em múltiplos canais, por exemplo consulta de dados em página *Web* e consulta de dados no telefone celular, ou quaisquer outros dispositivos móveis (*Smartphone, iPhone, iPad, Tablet*), onde a funcionalidade é desenvolvida duas vezes, uma para cada canal, a PROCERGS adotará a abordagem *multiple instance* que conta duas funcionalidades: a consulta de dados na *Web* e a consulta de dados no dispositivo móvel.

Nos casos em que um componente, *plugin* ou ferramenta gere o conteúdo no formato necessário para disponibilizar a consulta no dispositivo móvel, a funcionalidade será contada apenas uma vez, observando que a funcionalidade será da ferramenta e não da aplicação.

Algumas vezes poderão ser desenvolvidos projetos distintos, um projeto relativo ao sistema *Web* e outro para o sistema do dispositivo móvel. Nestes casos serão contadas as funcionalidades relativas a cada projeto.

3.7.5 Relatórios em Múltiplos Formatos

Nos casos em que um relatório deve ser entregue em diferentes formatos, por exemplo, em um arquivo *html* e um formato de valores separados por vírgula a PROCERGS irá considerar a ferramenta utilizada na geração de 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 o usuário visualize o relatório em tela e o gerador permita ao usuário imprimir o

relatório, salvar em *html* ou salvar no formato de valores separados por vírgula, então se contará apenas uma vez, observando que a funcionalidade será da ferramenta e não da aplicação.

3.8 Subdivisão de funcionalidades (abas / telas)

Quando uma funcionalidade é subdividida em abas ou telas encadeadas, é importante avaliar a funcionalidade em questão a partir de uma perspectiva do negócio, verificando se é completa e reconhecida pelo usuário.

As orientações básicas para contar subdivisão de funcionalidades são:

- verificar se, caso a funcionalidade não fosse fragmentada e houvesse uma interface única, a necessidade de negócio seria atendida independentemente de a funcionalidade ter menor grau de usabilidade ou desempenho insatisfatório ou oferecer maior complexidade técnica;
- avaliar se há usuários de áreas de negócio distintas responsáveis por preencher telas ou abas específicas da funcionalidade, não tendo competência (mesmo que munidos de todas as informações necessárias) para o preenchimento completo do formulário.

3.8.1 Funcionalidade com etapas na forma de telas encadeadas

Cenário no qual as telas são encadeadas de forma semelhante a um wizard, isto é, a transação deve ser reiniciada desde a primeira quando não concluído o processo. Neste caso, a funcionalidade foi quebrada em etapas com a única finalidade de tornar a entrada de dados para um mesmo usuário mais intuitiva e organizada, ou seja, apenas para atender a requisitos não funcionais de usabilidade. Portanto, como as telas constituem etapas para se atingir um objetivo único, deve ser contado apenas 1 (um) processo elementar.

3.8.2 Funcionalidade com abas

A funcionalidade foi dividida em etapas com a exclusiva finalidade de tornar a entrada de dados para um mesmo usuário mais intuitiva e organizada, ou seja, apenas para atender a requisitos não funcionais de usabilidade. A transação em si só estará completa quando todas as abas estiverem preenchidas, isto é, as abas não são funcionalidades autocontidas. Portanto, deve ser contado apenas 1 (um) processo elementar.

3.8.3 Subdivisão em telas ou abas, com a possibilidade de salvar rascunho

Esta é a situação na qual, antes de prosseguir para o próximo passo (tela ou aba), o usuário tem a opção de salvar os dados inseridos como rascunho, isto é, sem finalizar o passo ou a transação. Ainda que salvar como rascunho ou com versão final sejam duas ações em um único formulário, em última instância trata-se de um campo que é informado pelo usuário. Portanto, o recurso de salvar como rascunho na mesma interface (tela ou aba) na qual os dados são inseridos não é condição determinante para definir quantos processos elementares devem ser contados. Sendo assim, neste caso devem ser analisados outros critérios para essa definição.

3.8.4 Subdivisão em telas ou abas, salvando versão final em passo específico

Este é o caso em que as informações são salvas como rascunho em todos os passos, havendo um passo final em que não há entrada de dados, mas somente uma revisão das informações após a qual elas podem ter alterada sua condição para versão final. Nesta situação, a promoção dos dados à condição de versão final é um processo elementar completo e independente dos passos anteriores de inclusão ou alteração. Logo, nesta situação há, no mínimo, 2 (dois) processos elementares.

3.8.5 Funcionalidade subdividida para atender a uma necessidade do negócio

Encaixam-se nessa situação telas que devem ser preenchidas por usuários diferentes a fim de se alcançar o objetivo final da funcionalidade. Neste caso, há o indício de que pode se tratar mais de um processo elementar.

3.9 Dados Histórico e Auditoria

3.9.1 Dados históricos

Gravar histórico consiste em registrar os dados anteriores a uma atualização, a fim de preservá-los para que seja possível consultar a evolução da informação ao longo do tempo. Isto pode ser exigido pelo próprio cenário de negócio ou por um contexto externo, como órgãos de controle, por exemplo. De qualquer forma, a existência de histórico, bem como a consulta a ele, devem ser solicitados pelo usuário para que façam parte do tamanho funcional da aplicação.

A função de consulta aos dados deverá ser contada normalmente como uma função transacional, considerando o histórico como um registro lógico do ALI (Arquivo Lógico Interno) relacionado. Somente pode ser contada função de transação separada para incluir, alterar ou excluir as informações históricas para qualquer dessas operações quando ela não for parte integrante das mesmas funcionalidades que processam os dados de negócio.

3.9.2 Trilha de auditoria

Uma trilha de auditoria tem o objetivo de armazenar informações referentes às ações realizadas pelos usuários da aplicação, para que seja possível apurar quais foram as ações executadas dentro do sistema.

Uma vez que a trilha de auditoria seja solicitada pelo usuário, na contagem ela será considerada como um registro lógico referenciado do ALI (Arquivo Lógico Interno) relacionado, devendo existir funcionalidade de consulta a tais dados. Somente pode ser contada função de transação separada para incluir, alterar ou excluir os dados de trilha de auditoria para qualquer dessas operações quando ela não for parte integrante das mesmas funcionalidades que processam os dados de negócio.

3.10 Contagem de Middleware

3.10.1 Aplicações Middleware

Para correta medição de aplicações middleware, é essencial avaliar e definir corretamente a sua fronteira e os usuários que irão interagir com o sistema. O negócio destes aplicativos é servir como intermediário para comunicação entre dois sistemas, se a fronteira do middleware não estiver clara, esta poderá englobar parte das fronteiras das aplicações responsáveis por interagir com o middleware, comprometendo a medição realizada.

A contagem de aplicativos middleware será realizada conforme orientação do IFPUG: Pontos de Função & Contagem de Software Aplicativo Middleware - Versão 1.0

3.10.2 Aplicações que disponibilizam Webservices

Em uma aplicação se os serviços oferecidos são usados internamente ao aplicativo sendo contado, de tal forma que cumpre o papel de um **componente interno ao software sendo medido**, ele por si só não é medido pela APF. As funcionalidades quando avaliados em termos de sua complexidade, devem ter incluídos nessa avaliação os arquivos referenciados por esses webservices.

Os webservices que "exportam" serviços para outras aplicações devem ser avaliados em termos de funções do usuário e contados normalmente conforme a sua principal intenção.

Para que sejam considerados na contagem deve existir uma evidência que comprove que a solicitação de criação desse webservice foi motivada pelo cliente (o requisito funcional deve constar no Documento do Projeto no item Requisitos Funcionais) e devem ter a regra de negócio definida.

Os webservices que "exportam" serviços para outras aplicações não serão considerados na contagem se for acessado internamente por tela para teste. Neste caso tela e serviço contam como um processo elementar.

Exemplo:

Sistema A: responsável pelo agendamento e controle de serviços das regiões de uma empresa.

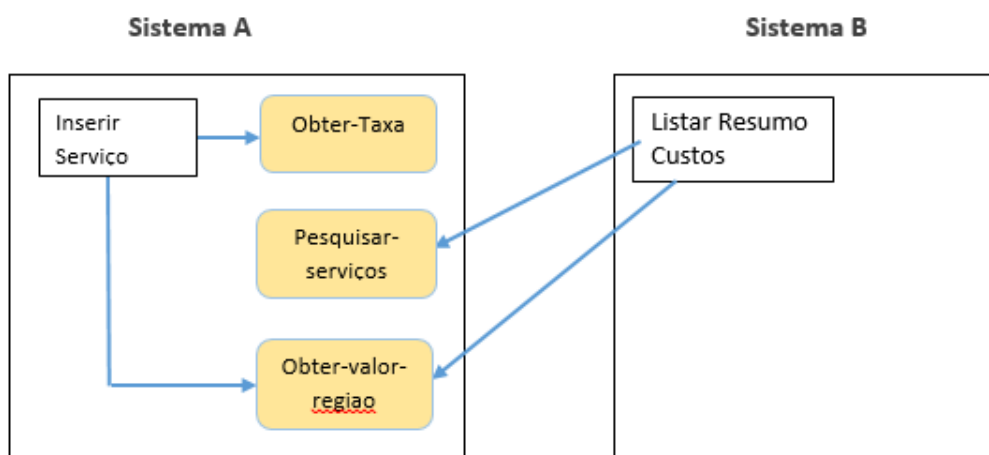
Disponibiliza:

- funcionalidade "Inserir Serviço" que irá gravar dados do serviço previsto e o custo estimado. Para o cálculo do custo é considerado a taxa do serviço do período e o valor cobrado por região;
- consulta do valor previsto por região e pesquisa dos serviços realizados por região no período para que o Sistema B responsável pelos custos utilize estas informações.

Sistemas B: controle de custo da empresa.

Disponibiliza:

- funcionalidade que lista o resumo de custos da empresa e deve considerar os serviços previsto no período em análise. Os dados de serviços são obtidos do Sistema A



Sistema A:

- o webservice “Obter-taxa” não é contado separadamente porque faz parte da funcionalidade “Inserir Serviço”.
- o webservice “Pesquisar-servicos” e o “Obter-valor-regiao” são contados separadamente porque são requisitos explicitados pelo cliente.

Observe que o serviço “Obter-valor-regiao” é considerado para fins de contagem, mesmo ele sendo consumido/utilizado por uma funcionalidade interna ao sistema A (Inserir Serviço), isso se deve ao fato desse serviço ser utilizado também por uma funcionalidade externa ao sistema A, (no caso a função Listar Resumo Custo do sistema B) e esse "compartilhamento" do serviço ter sido explicitamente demandado pelo cliente. Por outro lado, o serviço “Obter-taxa” é desconsiderado pois ele não é compartilhado com nenhum outro sistema, sendo consumido apenas internamente no sistema A."

Sistema B:

- o arquivo Serviços e Regiões são considerados como AIE e somente são contados como TD do AIE os campos utilizados na aplicação B.
- a função de transação “Listar Resumo de Custos” deve considerar como arquivo referenciado Serviços e Regiões e os campos que cruzam a fronteira da aplicação devem ser contados.

3.11 Contagem para Manutenção Corretiva

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

O padrão IEEE [IEEE,1998] define um tipo de manutenção corretiva, denominado de Manutenção Emergencial como “manutenção corretiva não programada executada para manter o sistema em estado operacional”.

Quando o sistema tiver sido desenvolvido pela empresa contratada, a manutenção corretiva será do tipo Garantia, conforme prazos e demais cláusulas do contrato em questão. Desta forma, manutenções corretivas estão compreendidas dentro do custo do desenvolvimento do projeto.

Quando o sistema estiver fora da garantia, deverá ser estimado e calculado o tamanho do projeto de manutenção corretiva. Nestes casos, a aferição do tamanho em pontos de função da funcionalidade ou das funcionalidades corrigidas deve considerar o fator de impacto que será aplicado sobre as funcionalidades alteradas.

Fórmula:

$$PF_CORRETIVA = PF_ALTERADO \times FI$$

Fator de Impacto (FI)	
FI	Regra de aplicação
0,50	Fora da garantia e a correção executada pela empresa que desenvolveu a funcionalidade.
0,75	Fora da garantia e a correção executada por empresa diferente daquela que desenvolveu a funcionalidade.

As demandas de manutenção corretiva não contemplam atualização de documentação da funcionalidade corrigida, pois este guia considera que, normalmente, manutenção corretiva não se refere a erros de requisitos. Caso seja erro em requisitos, essa demanda deve ser tratada como projeto de melhoria (alteração de funcionalidade).

Se for demandada a redocumentação da funcionalidade corrigida, porque a documentação não existe ou está desatualizada, deve-se adicionar ao FI um fator de redocumentação (FI + 0,15).

3.12 Processos Batch

Os processos batch disparados pelo relógio do sistema (clock) em que nenhuma informação cruza a fronteira, não são considerados como processos elementares. Eles complementam algum processo elementar da aplicação e este processo elementar deverá ser identificado e medido. Processos batch onde nenhum dado atravessa a fronteira mas que estão automatizando processos que poderiam ser processados de forma online, serão considerados processos elementares se:

- o usuário identifica como processos elementares, ou seja, são menores unidades de atividades com significado para o usuário;
- são auto-contidos;
- deixam o negócio da aplicação contada em um estado consistente.

Entradas Batch e On-line:

■ Funções On-line com Processamento Batch – Item aplicável apenas para sistemas transacionais

Alguns sistemas possuem funcionalidades on-line com processamento bastante pesado. Assim, na visão do usuário todo processo é on-line, mas na prática, os dados são gravados por processos batch. Nestes casos, seguindo a visão do usuário, as funções batch não são contadas separadamente, ou seja, o processamento batch faz parte da função on-line.

Exemplo: O cliente realiza um DOC em um caixa eletrônico. Essa transação só será efetivada após as 16h, que é o momento que uma rotina batch irá processar todas as transações de DOCs. O processo elementar Efetuar DOC é único e só é completo após a execução da rotina batch.

■ Funções Executadas em Modo on-line e Batch – Item aplicável apenas para sistemas transacionais

Em alguns sistemas, pode-se observar a opção para o usuário de executar uma função on-line ou em modo batch.

Exemplo: suponha a função “excluir processos inativos” no Sistema de Controle de Processo. Esta função pode ser realizada de forma manual pelo usuário em modo on-line. No entanto, a funcionalidade em modo batch exclui, automaticamente, todos os processos com a situação de inativo a mais de 1 (um) ano. Já a rotina de modo on-line permite a exclusão de processos inativos em período informado pelo usuário. Neste caso, apesar das funções on-line e batch possuírem o mesmo propósito: excluir processos devem ser contadas separadamente por possuírem lógica de processamento distinta. As funções executadas em modo batch e modo on-line constituem dois processos elementares distintos. A contagem de Pontos de Função será baseada nos critérios de unicidade de funções (Entrada Externa, Consulta Externa e Saída Externa) do CPM, ou seja, se estes processos possuírem os mesmos arquivos referenciados, os mesmos tipos de dados e a mesma lógica de processamento serão contados apenas uma vez. A entrada de dados em modo batch ou on-line não caracteriza uma lógica de processamento do CPM. É necessário o atendimento a uma lógica de processamento para justificar a contagem separada.

■ Funções Batch sem Dados Negociais Cruzando a Fronteira

Alguns sistemas possuem funcionalidades batch que não possuem dados negociais cruzando a fronteira da aplicação.

Exemplo: funcionalidade para exclusão de clientes inativos da base de dados, funcionalidade para atualização de saldos de contas correntes. Nestes casos, a funcionalidade existe para atender requisitos funcionais do usuário e devem ser contadas como Entradas Externas. No entanto, como não existem dados negociais cruzando a fronteira da aplicação, conta-se 1 DER (tipo de dado) para a capacidade de iniciar o processo e quantos ALRs forem necessários para completar o processo elementar.

3.13 Manutenção em Interface

A manutenção em interface, denominada na literatura de manutenção cosmética, é associada às demandas de alterações de interface, por exemplo: fonte de letra, cores de telas, logomarca, mudança de botões na tela, mudança de posição de campos ou texto na telas ou relatórios, sem que haja alteração em elementos de dados, arquivos referenciados ou informações de controle. Também se enquadram nessa categoria as seguintes manutenções:

- mudanças de texto em mensagens de erro, validação, aviso, alerta, confirmação ou conclusão de processamento;
- inclusão, alteração ou exclusão de cabeçalhos, títulos, máscaras de campos, alteração de nome de botões, *labels* ou qualquer outro tipo de literal;
- mudanças de posição de campos em telas, relatórios ou *layout* de arquivos, divisão de telas e/ou relatórios, mudança de posição de botões em telas;
- 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.

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 10% da contagem de uma função transacional de mais baixa complexidade (3 PF), ou seja 0,3 PF, independentemente da complexidade da funcionalidade alterada. Neste tipo de manutenção não são contadas funções de dados.

Fórmula:

$$\text{PF_INTERFACE} = \text{QTD FUNCOES TRANSACIONAIS ALTERADAS} \times 0,30 \text{ PF}$$

O esforço atrelado à fórmula definida compreende não só o esforço de implementação da demanda, como também a atualização de todos os artefatos necessários, de acordo com a Metodologia de Desenvolvimento da PROCERGS, impactados pela manutenção das demandas desta categoria.

A medição não é cumulativa dentro da mesma funcionalidade. Por exemplo, no caso de uma funcionalidade ter sido alterada em função de um requisito funcional, apenas a alteração funcional será considerada. Em analogia, o mesmo se aplica nos casos de manutenção adaptativa em requisitos não funcionais e manutenção em interface (cosmética) na mesma funcionalidade, a funcionalidade será contada apenas uma vez como manutenção adaptativa em requisito não funcional.

3.14 Manutenção Adaptativa em Requisitos não Funcionais

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

Exemplos:

- aumentar a quantidade de linhas por página em um relatório;
- colocar paginação em um relatório;
- limitar ou alterar 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 funcionalidade existente em outra tela da aplicação. Por exemplo: inclusão de botão na tela para 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.

Nestes casos, a aferição do tamanho em pontos de função da funcionalidade ou das funcionalidades que sofreram impacto deve considerar um fator de impacto sobre o PF_ALTERADO, seguindo os conceitos do CPM 4.3, apresentados na seção 4.2.

Fórmula:

$$\text{PF ADAPTATIVA} = \text{PF_ALTERADO} \times \text{FI}$$

Fator de Impacto (FI)	
FI	Regra de aplicação
0,50	Funcionalidade de sistema desenvolvida ou mantida por meio de um projeto de melhoria pela empresa contratada;
0,75	Funcionalidade de sistema não desenvolvida ou mantida por meio de um projeto de melhoria pela empresa contratada;

Deve-se destacar que além da adequação das funcionalidades em questão, a documentação das funcionalidades impactadas deverá ser atualizada. Se for demandada a redocumentação dessas funcionalidades, deve ser adicionado ao FI um fator de redocumentação (FI + 0,15).

3.15 Manutenção em Dados de Código

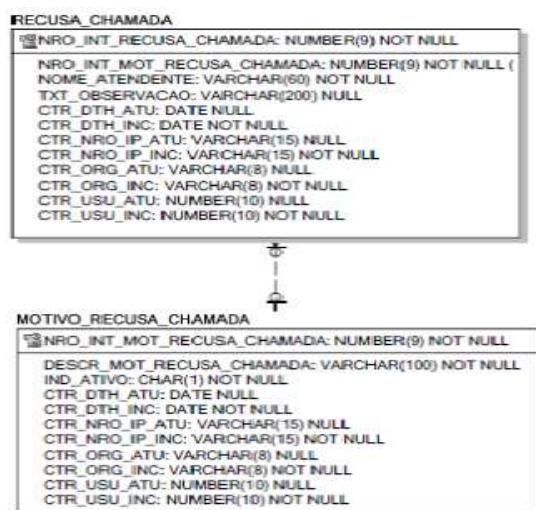
Contempla a necessidade de criação, alteração e exclusão de tabelas de Dados de Código e funções de manutenção dos mesmos (inclusão, alteração, pesquisa, consulta e exclusão dos dados de código) em **Projetos de Melhoria**. Será aplicado esta contagem somente após finalização do projeto, conforme definido no Documento de Projeto.

São considerados dados de código:

- entidade de dados de substituição, que contém um código e um nome ou descrição explicativos;
- entidade de ocorrência única, que contém um ou mais atributos que raramente ou nunca mudam;
- entidade que contém dados basicamente estáticos, ou que muito raramente mudam;
- entidade de valores *default*, que contém valores para popular atributos;
- entidade de valores válidos, que contém valores disponíveis para seleção ou validação;
- entidade que contém uma faixa de dados para validação.

As entidades acima referidas como dados de código normalmente contem atributos como código, descrição e/ou outros atributos como abreviação, finalidade, data de inicio de vigência, data de expiração, além de outros atributos para fins de auditoria e possuem comportamento padrão dos métodos CRUD (*Create, Read, Update e Delete*).

Exemplo:



A entidade Motivo_Recusa_Chamada é um exemplo de Tabela de Dado de Código.

Nestes casos, serão consideradas 0,30 pontos de função **por tabela e/ou transação** incluída, alterada ou excluída.

Fórmula:

$$\text{PF_DADOS_CODIGO} = \text{QTD_TABELAS_TRANSACOES} \times 0,30 \text{ PF}$$

3.16 Atualização de Dados de Domínio

Em alguns casos, as demandas de atualização em base de dados estão associadas a atualizações em um único registro em uma Tabela de Domínio do tipo *Code Data* (Dado de Código). Por exemplo, a atualização de uma Tabela de Situação de Projetos (Andamento, Concluído, Cancelado) para inclusão da Situação “Suspensão”. Esta demanda será classificada como Atualização de Dados de Domínio (PF_DOMÍNIO) e contada como 10% da contagem de Pontos de Função de uma Entrada Externa de complexidade baixa, ou seja 0,3 PF. No

exemplo acima, EE: Atualizar Situação de Projetos (AR:1 Situação do Projeto; TD: 1 Situação do Projeto) – Baixa – 3 PF x 10% = 0,3 PF.

Fórmula:

$$\boxed{\text{PF_DOMÍNIO} = \text{PF_INCLUÍDO} \times 0,10}$$

3.17 Páginas Estáticas e Funcionalidades Multi Idiomas

Nesta seção são tratados desenvolvimento e manutenção específicas em páginas estáticas de *sites*, *subsites* e portais que não estão contidos em um projeto de desenvolvimento, bem como a publicação de páginas *web* com conteúdo estático. Por exemplo: criação de página HTML, atualização de menu estático, atualização de texto, imagem, ou *banner* estáticos em páginas HTML. Além de demandas de atualizações em *sites*, *subsites* e portais para contemplar mais de um idioma, sem que ocorram alterações nos requisitos funcionais.

As demandas de criação de logomarcas ou identidade visual, além de outras demandas de criação de arte, associadas a área de comunicação social não são enquadradas nesta categoria.

Seguem alguns exemplos de demandas consideradas nesta categoria:

- adição ou reestruturação de menus de navegação estáticos;
- alteração, inclusão ou exclusão de páginas estáticas na aplicação, portais ou *sites*;
- alterações em textos de imagens, para *sites* que forem desenvolvidos em mais de um idioma em que há necessidade de alterar o texto da imagem para outro idioma. A alteração deve considerar a quantidade de páginas estáticas ou funcionalidades alteradas em função da atualização das imagens;
- alterações nas funcionalidades de um *site* para contemplar mais de um idioma, sem que haja alterações nos requisitos funcionais. A alteração deve considerar a quantidade de funcionalidades alteradas.

Nestes casos, serão consideradas 0,30 pontos de função por página desenvolvida ou mantida, ou por funcionalidade alterada.

Fórmula:

$$\boxed{\text{PF_PAGINA_EST_SITE_PORTAL} = \text{QTD_PAGINAS_FUNCOES} \times 0,30 \text{ PF}}$$

3.18 Manutenção de Componente Reutilizável

Para o desenvolvimento de um novo sistema, o uso de componentes de código reutilizáveis não afetará o tamanho da aplicação e nem a remuneração do serviço. Entretanto, o reuso proporcionará melhor produtividade e naturalmente será uma opção adotada.

Para um projeto de melhoria, se houver a necessidade de atualizar somente um componente que é utilizado por várias funcionalidades da aplicação, será considerado na contagem o componente que deverá ser testado como um processo elementar independente e contado como uma funcionalidade. Além disso, os processos elementares que o Analista de Sistemas da PROCERGS indicar que devem ser testados para validar a implementação devem ser dimensionados por meio da métrica Pontos de Função de Teste proposta na seção 3.13. Ainda

que não sejam testadas todas as funcionalidades que façam uso do componente reutilizável, estas ainda sim continuarão sendo cobertas pela garantia de correção de defeitos.

Exemplo: no sistema de Folha de Pagamento, o componente que valida o código do funcionário é utilizado por 25 processos elementares. Este componente precisa ser alterado para considerar um novo departamento da empresa que possuirá uma validação diferenciada. O Analista de Sistemas definiu que para validar esta alteração, devem ser testados os seguintes processos: Incluir Funcionário, Alterar Funcionário, Selecionar Funcionário e Pesquisar Funcionário por código. Na contagem será considerado: a contagem do componente que valida o código do funcionário como uma funcionalidade; a contagem dos 4 processos elementares que serão testados, pela métrica de Ponto de Função de Teste.

Fórmula:

$$\text{PF_COMPONENTE} = \text{PF_FUNCIONALIDADE_COMPONENTE} + \text{PF_TESTES}$$

3.19 Atualização de Versão – Browser

Nesta categoria encontram-se as demandas de atualização de aplicações *web* para executar em novas versões de um mesmo *browser* e para suportar a execução em mais de um *browser* (além dos previstos na demanda inicial do projeto). 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.

Fórmula:

$$\text{PF_ATUALIZAÇÃO_VERSÃO_BROWSER} = \text{PF_ALTERADO} \times 0,25$$

O PF_ALTERADO deve considerar apenas as funcionalidades alteradas. As funcionalidades apenas testadas devem ser contadas como PF_TESTES, conforme orientações descritas na seção 3.13 deste Guia.

Essas atualizações podem implicar em manutenções em componentes específicos da plataforma utilizada. Nesse caso, a demanda deve ser contada como manutenção de componentes, descrita na Seção 3.11 deste Guia.

3.20 Pontos de Função de Teste

Muitas vezes, em projetos de manutenção, o conjunto de funções transacionais a serem testadas é maior do que a quantidade de funções a serem implementadas, isto é, além das funcionalidades que são afetadas diretamente pelo projeto de manutenção, outras precisam ser testadas [NESMA, 2009]. O tamanho das funções a serem apenas testadas deve ser aferido em Pontos de Função de Teste (PFT). Não considerar as funcionalidades incluídas, alteradas ou excluídas do projeto de manutenção na contagem de Pontos de Função de Teste.

A contagem de PFT será o somatório dos tamanhos em pontos de função das funções transacionais envolvidas nos testes multiplicado pelo percentual da etapa de teste executada.

No caso de uma função ser testada várias vezes, com cenários diferentes, a função só pode ser contada uma vez.

Fórmula:

$$\text{PF_TESTES} = \text{SOMATORIO_PF_FUNÇÕES_TRANSACIONAIS} \\ \times \text{PERCENTUAL_ETAPA_TESTES_EXECUTADAS}$$

Onde:

PERCENTUAL_ETAPA_TESTES_EXECUTADAS = Percentual da Etapa de Testes completo ou a Soma das Etapas de Testes Executadas (Planejamento de Testes, Automatização dos Testes, Execução dos Testes), conforme etapas e percentuais de esforço definidos no contrato estabelecido com o fornecedor que irá executar a demanda.

Por exemplo: A soma das funcionalidades que deverão ser testadas em função da inclusão de uma nova funcionalidade na aplicação é 30 PFs. O fornecedor fez somente a Execução dos Testes que corresponde a 5%. O cálculo do PF_Testes será: $30 \times 5\% = 1,5$ PF.

3.21 Atualização de Manual do Usuário e Help Online

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 desenvolvimento de sistemas. No caso de demandas específicas de desenvolvimento ou atualização de help estático de funcionalidades, sem alteração nos requisitos funcionais, estas podem ser enquadradas nesta seção. Por exemplo: alterações na 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:

- inclusão ou alteração de fluxos;
- mudanças no significado, finalidade, origem dos campos e bibliografia.

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 0,10 PF, independentemente da complexidade da funcionalidade alterada, conforme fórmula abaixo. Neste tipo de manutenção não são contadas funções de dados.

Fórmula:

$$\text{PF_ATUALIZACAO_HELP} = \text{QTD. FUNCOES TRANSACIONAIS ALTERADAS} \times 0,10 \text{ PF}$$

Em caso de requisitos de usuário para o desenvolvimento de funcionalidades de manutenção de help, deve-se contar a função de dados de help e as funcionalidades de manutenção de help (por exemplo: incluir help de tela, consultar help de campo) de acordo com o CPM 4.3.

3.22 Dicas para Otimizar o Custo das Manutenções

É fundamental que se tenha uma atenção especial na gestão de demandas de manutenção de sistemas. Se toda demanda de manutenção que chegar for encaminhada diretamente para execução, a tendência é que o custo destas manutenções ao final sejam superiores ao que poderiam ser, caso houvesse um controle sobre estas solicitações.

A seguir algumas dicas que podem ajudar a melhorar o cenário.

- **Consolidar manutenções na mesma funcionalidade em uma única demanda**

Fazer uma manutenção para atender um único requisito ou para atender a vários requisitos de manutenção na mesma função terá o mesmo tamanho funcional, se elas forem solicitadas no mesmo momento. Se solicitadas em momentos distintos, as mesmas funções serão pagas várias vezes, para cada um dos projetos de melhoria.

No entanto, nem sempre é possível represar uma necessidade do usuário para que esta seja agrupada com outras, já que há demandas com prazos críticos. O importante é tentar avaliar ao máximo quais ajustes realmente são críticos e quais não são, visando evitar o cenário descrito acima.

- **Reutilizar funções existentes em outros sistemas**

Muitas vezes algumas funções já existem em outros sistemas e mesmo assim ocorre a replicação destas funcionalidades, principalmente com os requisitos de armazenamento (Arquivos Lógicos).

Exemplo:

Supondo que vários sistemas utilizam informações de endereço (estado, cidade, bairro), mantendo em sua base local tabelas e respectivas funcionalidades de manutenção destes dados. Por conta de um acréscimo de informação na tabela de cidades (número de habitantes, latitude, longitude), diversos sistemas tiveram que sofrer manutenção. Em alguns deles pagou-se pelas mesmas funções em várias aplicações: o ALI que armazena os dados e as transações que mantém estes dados.

Se estas funcionalidades estivessem centralizadas em um único sistema, as funções de manutenção (inclusão, alteração, consulta) seriam contadas no projeto de melhoria do sistema responsável por estes dados e os demais contariam apenas um AIE alterado e suas funções específicas impactadas, que seriam contadas de qualquer forma. Desta forma se reduziria o total de pontos de função alterados.

- **Análise crítica dos requisitos**

Em muitas situações é possível ter uma única função que faça o papel de duas existentes. Isto é muito comum no caso de consultas e relatórios com diferença apenas de alguns atributos apresentados. Ou seja, uma transação mais completa poderia ser elaborada para evitar a criação de várias funções distintas, porém semelhantes.

É mais barato pagar pela criação de uma função nova do que por uma manutenção em duas ou mais funcionalidades, principalmente a longo prazo.

- **Evitar particionar funcionalidades em mais de uma *release***

Sempre que possível deve ser evitado o particionamento de uma funcionalidade em mais de uma *release*. Deve-se procurar construir funcionalidades completas a cada *release* e quando necessário particioná-las deve-se realizar um planejamento prévio e registrar no documento de projeto.

4 Contagem de Processos Automatizados

Os processos organizacionais automatizados são executados dentro do ambiente BPMS. Um BPMS (*Business Process Management System*) é um Sistema no qual se pode desenvolver aplicativos integrados para gerenciamento de processos de negócios contemplando recursos como: documentação de processos, execução automatizada de processos, *upload* e trâmite de documentos eletrônicos, com possibilidade de certificação digital e integração com sistemas através de desenvolvimento orientado a serviço.

Na contagem de pontos de função um dos passos iniciais da análise é a determinação da fronteira da aplicação. Para a contagem de pontos de função de automação de processos deve-se considerar como fronteira da aplicação, apenas a interface conceitual do processo de negócio que será automatizado. Tendo o cuidado para não considerar na contagem, as funcionalidades que já integram e fazem parte do ambiente BPMS, no qual o mesmo será desenvolvido. Ou seja, as funcionalidades pertinentes à ferramenta não farão parte do escopo da contagem. Exemplos: Consultar Minhas Tarefas, Meus Processos, Monitoramento dos Processos.

As ferramentas de BPMS disponibilizam algumas funcionalidades de cadastro como por exemplo: usuários e grupos organizacionais. Onde os usuários podem ser alocados aos grupos organizacionais. Os projetos de automação processos devem considerar essas funcionalidades de cadastro como um AIE que é mantido pela aplicação BPM, e os processos automatizados acessarão seus dados, contando estes como Tipos de Dados.

4.1 Recebimento de dados de outras aplicações utilizando interface de integração

Em projetos de automação de processos que realizam integração e troca de informações com outras aplicações externas, é comum se deparar com algumas situações durante a análise de pontos de função:

Cenário 1:

O primeiro caso são situações onde o processo automatizado precisa ler/consultar um arquivo mantido por uma aplicação externa. Exemplo: Um processo X precisa consultar os dados de um funcionário através de sua matrícula que são mantidos num sistema de RH. O entendimento para esse cenário é de contagem de um AIE no processo automatizado referente aos dados consultados e mantidos no sistema de RH e tantos Tipos de Dados que foram consultados. Nesta situação o arquivo consultado já é contado como um ALI na aplicação externa.

Acentuando que a interface de integração utilizada nessa transação é apenas a tecnologia para acessar os dados mantidos em outras aplicações, assim não afetam a contagem.

Cenário 2:

O segundo caso são situações onde o processo automatizado necessita verificar ou consistir alguma informação em uma aplicação externa que possui total domínio da regra de negócio envolvida na transação. Exemplo: Um processo X necessita verificar se um determinado funcionário pode solicitar uma licença sem vencimento. Toda regra envolvida para essa verificação é de responsabilidade e realizada na aplicação externa, onde o processo X solicita essa verificação na aplicação externa. Para esse cenário onde toda a regra de negócio esta fora do processo X não se justifica nenhuma contagem de pontos de função.

4.2 Processo Elementar

Algumas atividades modeladas em BPMN, apesar de se apresentarem distintas no modelo, podem constituir apenas uma transação completa para o negócio, com sentido de completude de determinado requisito funcional para o usuário. Assim, devem ser contadas como apenas uma transação. Assim sendo, é factível que se tenha na modelagem BPMN mais de uma atividade ou mesmo um conjunto de atividades que sejam correspondentes a um único requisito funcional, sendo portanto considerado na contagem como uma única transação.

Alguns modelos BPMN podem apresentar atividades que foram modeladas apenas para melhor entendimento do negócio e, que não constitui um processo elementar para o sistema. Assim não devem ser levadas em consideração na contagem.

Uma mesma atividade que pode ser realizada por pessoas/entidades diferentes deve ser contada apenas uma vez.

Atividades que são diferentes dependendo da pessoa/entidade que realiza devem ser contadas separadamente.

Em casos que uma funcionalidade da aplicação ativa um fluxo na ferramenta de automação de processos deve ser considerado um único processo elementar.

Para as atividades que tenham um prazo para serem realizadas, onde após a expiração do prazo uma notificação/lembrete é disparada, essa notificação seja por *e-mail*, *sms* ou outro meio, não deve ser contada, pois faz parte do negócio da atividade.

5 Contagem de Sites

Definição de *Site*: um *Site* é constituído por um conjunto de informações, bem estruturadas e facilmente encontradas por seu público. Normalmente estas informações concentram-se em notícias, institucional, fotos, vídeos e informações sobre serviços. Este tipo de construção atende um padrão para vários clientes. Exemplos: www.spm.rs.gov.br, www.sdr.rs.gov.br.

O conteúdo dos *Sites* desenvolvidos pela PROCERGS é dinâmico e contém, em sua maior parte, funcionalidades previstas no CMS PROCERGS (ferramenta de gestão de conteúdo implementada pela PROCERGS).

Utilizando as atuais práticas de contagem, conforme estabelecido no CPM 4.3 os vários recursos fornecidos a um usuário final de um *Site* podem ser classificados como funções de dados e de transação (ALIs, AIEs, EEs, CEs e SEs) e a complexidade funcional determinada pela quantidade TDs e TRs dessas funções.

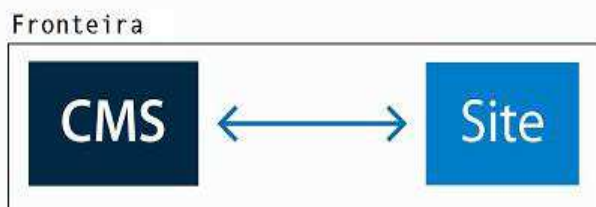
5.1 Definições

Usuário

Qualquer pessoa que utiliza o *Site* (por exemplo, realiza buscas, consulta notícias, atualiza e insere dados em formulários). O *Webmaster*, no papel de administrador do *Site*, mantendo o conteúdo do *Site*, usando a funcionalidade de aplicativos ou ferramentas externas, como por exemplo a ferramenta CMS, desenvolvida pela PROCERGS para gestão de conteúdo, também é considerado um usuário.

Fronteira da Aplicação

Uma aplicação é definida como um conjunto de funções logicamente relacionadas, interativas, que cumprem um requisito de negócio específicos definidos por um usuário. O aplicativo CMS, onde o usuário administrador mantém o conteúdo do Sistema, e o Aplicativo *Web* do *Site*, onde o conteúdo é disponibilizado para os demais usuários que acessam o *Site*, fazem parte da mesma fronteira da aplicação, conforme ilustrado na imagem abaixo.



Fronteiras devem ser baseados em funções de negócio separadas, como visto pelo usuário, e não em considerações tecnológicas. Por exemplo, um *Site* pode realmente ser composto de

múltiplas aplicações. Os usuários podem acessar diferentes divisões de uma empresa por meio de aplicativos *Web Site*, como o Fórum *Web*, catálogo de vendas e entrada de pedidos. Cada um desses teria um propósito de negócios diferenciado. Ou pode-se ter um *Site* e um Sistema/Aplicativo *Web* com funcionalidades que sustentam o *Site* e ambos devem ser considerados uma mesma aplicação, fazem parte da mesma fronteira, pois tem um mesmo propósito de negócio para o usuário.

Deve-se procurar o ponto de vista dos usuários que fazem uso da aplicação para ajudar a determinar os limites de aplicação.

Além disso deve-se determinar quais são as funções do aplicativo (*Site*) e quais são as funções da ferramenta de gestão de conteúdo (ferramenta CMS) para ajudar a determinar as funcionalidades a serem contadas na aplicação(*Site/Portal*). Funções de Dados e de Transação implementadas e disponibilizadas pela ferramenta CMS não devem fazer parte do escopo da contagem do *Site*.

5.2 Funções de Dados

Assim como a contagem de quaisquer outros dados em um sistema usando regras do CPM, os dados devem ser reconhecidos e mantidos pelo usuário para ser considerado um ALI ou AIE.

A exceção no caso de *Sites* que utilizam a ferramenta de gestão de conteúdo, é que arquivos que contêm o conteúdo do *Site* e que fazem parte do CMS não serão contados. Somente serão contados os arquivos que forem específicos do *Site* que será desenvolvido, tais como dados de formulários que podem ser mantidos por transações disponíveis através do *Site* e que cumprirem um requisito de usuário.

5.3 Tipos de Funções Transacionais

Entradas Externas

Se uma transação permite ao usuário armazenar dados (por exemplo, inclusão de uma inscrição em um formulário) dentro do aplicativo do *Site*, será contada como uma Entrada Externa.

Consultas Externas

De acordo com o tipo de conteúdo disponibilizado no *Site* (notícias, fotos, artigos, entre outros), há uma forma de apresentação do conteúdo. Esta forma de apresentação é definida na ferramenta CMS pelo **grupo de classificação**. Atualmente a ferramenta CMS disponibiliza os seguintes grupos de classificação:

- Conteúdo puro sem categorias ou conteúdo puro sem listas
- Conteúdo categorizado
- Listas de conteúdo
- Galeria de Fotos
- Linque externo
- Conteúdo puro e conteúdo em categorias
- Conteúdo puro e conteúdo em lista
- Conteúdo puro e Galeria de Fotos

Para cada grupo de classificação disponibilizado pela ferramenta CMS deve-se considerar uma Consulta Externa (CE).

A funcionalidade de busca (busca de conteúdo no *Site* a partir de uma descrição informada pelo usuário) também deve ser considerada uma Consulta Externa (CE).

Uma pesquisa de conteúdo específica para um determinado link/menu também deve ser considerada como uma CE. No caso de vários links/menus, disponibilizarem uma pesquisa de conteúdo, todas as pesquisas seriam contadas como uma única CE.

Formulários que envia informações para fora da fronteira da aplicação (para um e-mail, por exemplo) sem que os dados sejam mantidos em arquivos (por exemplo formulários de Fale Conosco), também devem ser considerados como CE.

Uma página física pode ter múltiplas páginas lógicas com base na visão do usuário da informação e múltiplas transações podem estar contidas em uma única tela.

No caso de várias páginas físicas serem necessárias para satisfazer as necessidades do usuário, todas as páginas físicas devem ser contadas como uma única CE, como é o caso da página inicial de um *Site*, que deve ser contada como uma única consulta.

Cada Consulta Externa, não importa se acessada de diferentes formas (*link* ou menu, por exemplo), será contada uma vez por aplicativo, não importa quantas vezes ela pode ser referenciada a partir do aplicativo.

Importante: Links que fornecem navegação para outras áreas ou aplicações não são contados.

Saídas Externas

Se uma saída contém dados derivados (por exemplo, um contador), a página é contada como uma SE.

5.4 Complexidade Funcional

Arquivos

Para arquivos que contêm os detalhes de uma página *Web Site*, deve-se contar: um DET para cada campo único que descreve a página e um DET para o texto da própria página. Além disso deve-se contar 1 TR para cada subgrupo lógico de dados mantido nos ALI ou AIE.

A exceção no caso de *Sites* que fazem uso da ferramenta de gestão de conteúdo CMS, é que arquivos que contêm o conteúdo do *Site* e que fazem parte da ferramenta CMS não serão contados. Somente serão contados os arquivos que forem específicos do *Site* que será desenvolvido, tais como dados de formulários que podem ser mantidos por transações disponíveis através do *Site* e que cumprirem um requisito de usuário.

Transações

Para as páginas *Web Site*, deve-se contar 1 DET para o texto da página e 1 DET para cada hiperlink que permite ao usuário controlar o acesso a outra CE. O hiperlink é considerado como uma variável reconhecida pelo usuário ou campo, pois é único e fornece a capacidade de acessar uma página exclusiva.

Deve-se contar 1 TR para cada arquivo lido ou referenciado para recuperar os dados solicitados para a página.

6 Bibliografia Consultada

[ATI, 2012] ATI. *Guia de Contagem da ATI*. 2012.

[FNDE, 2010] FNDE. *Guia de Contagem do FNDE*. 2010.

[IFPUG, 1998] IFPUG. *White Papers under CPM 4.0 Counting Rules - New Environments Committee Hints to Counting Web Sites*. Released November, 1998.

[IEEE,1998] IEEE Computer Society. *IEEE Standard for Software Maintenance*. IEEE Std 1219, 1998.

[IFPUG,2009] IFPUG. *Considerations for Counting with Multiple Media*. Release 1.0, September, 2009.

[IFPUG,2010] IFPUG. *Counting Practices Manual*. Version 4.3, January, 2010.

[NESMA, 2009] NESMA. *Function Point Analysis for Software Enhancement Guidelines*. Version 2.2.1, 2009

[SISP, 2016] SISP. *Roteiro de Métricas de Software do SISP*. Versão 2.2, 2016.

[Vazquez, 2010] VAZQUEZ, C. E.; SIMÕES, G. S.; ALBERT, R. M. *Análise de Pontos de Função: Medição, Estimativas e Gerenciamento de Projetos de Software*. 9ª Edição. Editora Érica, São Paulo.

[IFPUG,2009] IFPUG. *Pontos de Função & Contagem de Software Aplicativo Middleware*. Versão 1.0.