

## 1. INTRODUÇÃO

### 1.1 PROPÓSITO DO GUIA DE CONTAGEM

A Análise de Pontos de Função (APF) é uma técnica padronizada pela *International Function Point Users Group* – IFPUG ([www.ifpug.org](http://www.ifpug.org)) – que visa medir o desenvolvimento e manutenção de software em termos significativos para os seus usuários, com base na visão de negócio. O ponto de função é a unidade utilizada para tal fim e busca em um único número ponderar os requisitos funcionais de armazenamento e processamento de uma aplicação ou projeto. As regras, procedimentos e práticas de contagem estão definidos no *Manual de Práticas de Contagem de Pontos de Função*, versão 4.3 - (CPM 4.3).

Na medida em que a técnica se baseia na visão do usuário e também em aspectos de sua aplicação que demandam uma interpretação das regras e procedimentos, o próprio manual de contagem recomenda que haja um Guia Local de Contagem com as interpretações locais dessas regras gerais.

A finalidade deste documento é cumprir esse papel, facilitando o uso da técnica de análise de pontos de função dentro do contexto de desenvolvimento e manutenção de sistemas específicos da Prodemge. Portanto, funciona como um complemento ao CPM.

O princípio é fornecer uma referência local comum que torne mais prática a aplicação dos conceitos e regras definidos pelo IFPUG com a sua exemplificação em situações peculiares à Prodemge e também oriente quanto à contagem nas situações em que o IFPUG ainda não oferece orientação prática ou objetiva.

Este documento está sujeito a novas atualizações sempre que necessário. Qualquer sugestão, questionamento ou esclarecimento sobre o seu conteúdo deve ser enviado para [gps@prodemge.gov.br](mailto:gps@prodemge.gov.br).

### 1.2 PÚBLICO-ALVO

Este documento deve ser utilizado como referência por gestores de projetos, analistas de requisitos, analistas especializados em contagem de pontos de função, revisores e auditores da Prodemge e, no caso de empresas contratadas pela Prodemge para desenvolvimento em regime de fábrica de software, pelos responsáveis pela contagem de pontos de função dessas empresas.

### 1.3 EQUIPE TÉCNICA RESPONSÁVEL

A responsabilidade pela elaboração desse guia é da Gerência de Processo de Software (GPS) da Prodemge. Pedidos de análise, atualização, sugestões, questionamentos ou esclarecimentos sobre o seu conteúdo devem ser enviados para o endereço eletrônico [gps@prodemge.gov.br](mailto:gps@prodemge.gov.br).

### 1.4 PROCESSO DE ATUALIZAÇÃO E PUBLICAÇÃO

Este guia de contagem pode ser atualizado sempre que necessário para incluir, excluir ou alterar critérios ou procedimentos nele descritos, tanto por necessidade da Prodemge quanto de fábricas de software que estejam prestando serviço à empresa e que tenham seu trabalho afetado por disposições deste guia.

O processo de atualização do guia deverá seguir as seguintes etapas:

### 1. Comunicação da necessidade de alteração à equipe técnica responsável

Nesta etapa, um interessado ou envolvido num processo que utiliza a técnica de APF descreve de forma sumária o problema cujo tratamento ele deseja incluir no guia e solicita a iniciação do processo de alteração. Um processo de alteração pode ser iniciado sempre que for encontrada uma situação envolvendo a aplicação da técnica de APF cujo tratamento não esteja definido no guia, ou seja, identificada uma necessidade de aprimoramento na abordagem estabelecida. Para iniciar formalmente o processo, o interessado deve enviar um email à equipe técnica responsável pelo guia ([gps@prodemge.gov.br](mailto:gps@prodemge.gov.br)) descrevendo o contexto, a alteração proposta e sua respectiva justificativa.

Nota: Processos de alteração podem ser iniciados por membros da equipe responsável pela manutenção do guia. Nesse caso, o processo inicia-se na etapa 3 e não há necessidade de aprovação pela equipe técnica (etapa 4).

### 2. Decisão da equipe técnica pela avaliação ou não do caso

A equipe técnica analisa preliminarmente o caso e decide sobre o seu andamento, enviando uma resposta formal ao solicitante. Esta decisão poderá, em princípio:

- Aprovar a análise do caso. Nesse caso, a equipe técnica irá pedir a formalização da demanda, através do envio da Solicitação de Análise (descrita a seguir).
- Aprovar a análise do caso, mas decidir pela não avaliação imediata do mesmo, com a devida justificativa. Nesse caso, a equipe estabelecerá um prazo para o tratamento da questão.
- Decidir pela insuficiência de justificativas para o tratamento da solicitação. Nesse caso, também, a equipe técnica apresentará as devidas justificativas para sua decisão.

### 3. Solicitação de análise

O interessado na alteração deverá produzir uma solicitação de análise descrevendo em detalhes os problemas que motivaram o pedido de alteração do guia. Este documento formaliza a demanda junto à equipe técnica responsável. Uma mesma solicitação pode tratar de várias questões. Veja o item 1.6 para uma descrição do documento Solicitação de Análise.

### 4. Avaliação da equipe técnica

Nesta etapa a equipe técnica analisa o pedido de alteração fundamentado pela Solicitação de Análise, decidindo pela sua aprovação ou rejeição, com base em critérios técnicos.

- Caso aprovada, a proposta de modificação segue para a etapa de aprovação dos envolvidos se houver impacto em contratos celebrados com fornecedores. Caso não afete contratos vigentes, a proposta segue para a etapa de publicação.
- Caso a alteração seja reprovada, a equipe técnica produzirá a devida justificativa.

### 5. Aprovação dos envolvidos

Nesta etapa a Prodemge e as contratadas afetadas pelas modificações propostas analisam os impactos das modificações sob seus respectivos pontos de vista e aprovam ou não a modificação. Existem basicamente 3 situações:

- a) A modificação foi proposta pela Prodemge.

Neste caso, os impactos contratuais, caso existam, serão discutidos com as contratadas e, se a modificação for aprovada por todos, ela será incluída na próxima versão do Guia de Contagem. Considerações sobre a vigência e aplicabilidade do critério, caso tenham sido acordadas, serão refletidas no guia. Caso a modificação não seja aprovada, os contratados continuarão utilizando como referência a última versão do guia acordada. A modificação rejeitada poderá ou não ser incluída em novas versões do guia da Prodemge, a critério da equipe responsável, porém sem produzir efeitos nos contratos vigentes.

- b) A modificação foi proposta por uma contratada.

Neste caso, os impactos contratuais, caso existam, deverão ser aprovados pela Diretoria da Prodemge, após consulta às áreas internas envolvidas. Considerações sobre a vigência e aplicabilidade do critério, caso tenham sido acordadas, serão refletidas no guia.

- c) A modificação foi acordada previamente entre as partes.

Neste caso, os impactos contratuais, caso existam, deverão ser aprovados pela Diretoria da Prodemge. A modificação será incluída na próxima versão do guia de contagem. Considerações sobre a vigência e aplicabilidade do critério, caso tenham sido acordadas, serão refletidas no guia.

## 6. Publicação da nova versão do guia de contagem

Uma nova versão do guia, contendo a alteração proposta, é publicada e formalmente enviada aos envolvidos.

### 1.5 VERSÕES DO GUIA DE CONTAGEM

A Prodemge armazena o guia de contagem em versões e ramos, visando gerenciar o processo de atualização do guia e os seus impactos, tanto internos quanto externos.

A Prodemge poderá, a seu critério, utilizar versões diferentes do guia de contagem em seus processos internos e em contratos com fornecedores. Nesse caso em particular, cada contrato poderá referenciar seu próprio guia de contagem, em função de questões legais (por exemplo, adesão ao disposto no edital de referência), acordos em separado com contratadas, aprovação ou não de modificações, etc. Assim, a cada contrato será associado um ramo de versões do guia de contagem, que poderá evoluir de forma convergente ou não com o guia de contagem oficial da Prodemge.

A Prodemge buscará reduzir o número de versões do guia ao mínimo necessário à gestão adequada dos contratos já celebrados.

### 1.6 SOLICITAÇÃO DE ANÁLISE

Uma solicitação de análise é um instrumento de modificação do guia de contagem. Ela descreve um ou mais problemas no contexto de APF e propõe uma abordagem para a apropriação organizacional da técnica, fundamentada tecnicamente. O documento deverá conter tantas partes quantos forem os problemas analisados. Cada parte deverá ser estruturada em cinco seções de detalhamento e análise, como segue:

1. Contexto/Problema: descreve o contexto onde se insere o problema.
2. Abordagens: enfoca as alternativas de abordagem para tratamento do problema, devidamente justificadas por literatura técnica ou práticas de mercado.
3. Posição das entidades de referência técnica (IFPUG, BFPUG, NESMA, SISP, etc.): detalha a posição das entidades ou especialistas utilizados como referência técnica.
4. Apropriação sugerida: descreve a abordagem proposta para o tratamento da situação.
5. Exemplo: fornece um exemplo prático de aplicação da abordagem proposta.

Uma solicitação de análise pode ser aprovada ou não. Se aprovada, ela pode gerar uma modificação no guia. Caso contrário, o solicitante receberá uma justificativa para a rejeição ou para a opção da empresa pelo critério vigente.

## 1.7 REFERÊNCIAS TÉCNICAS

Livros & Manuais:

1. Function Point Counting Practices Manual (Manual de Práticas de Contagem de Pontos de Função, versão 4.3.1, publicado pelo IFPUG).
2. NESMA - Análise de Pontos de Função para Melhoria de Software, versão 2.2.1, tradução da versão original em inglês disponível em [www.nesma.nl](http://www.nesma.nl).
3. FPA for Software Enhancement, versão 1.0, publicado pela NESMA.
4. Análise de Pontos de Função – Medição, Estimativas e Gerenciamento de Projetos de Software, de Carlos Eduardo Vasquez, Guilherme Siqueira Simões e Renato Machado Albert.
5. Roteiro de Métricas de Software do SISP (Sistema de Administração dos Recursos de Tecnologia da Informação do Governo Federal), versão 2.0.

Sites:

1. International Function Point Users Group (IFPUG): <http://www.ifpug.org>
2. Brazilian Function Point Users Group (BFPUG): <http://www.bfpug.com.br>
3. International Software Benchmarking Standards Group (ISBSG): <http://www.isbsg.org>
4. Netherlands Software Metrics Association (NESMA): <http://www.nesma.nl/sectie/home/>
5. Sistema de Administração dos Recursos de Tecnologia da Informação (SISP): <http://www.sisp.gov.br/>

## 2. FUNDAMENTOS GERAIS DA APF

### 2.1 OBJETIVOS DA APF: A QUE A TÉCNICA SE PROPÕE

A APF propõe-se a:

- *Medir a funcionalidade que o usuário solicita e recebe.* Para efeitos da contagem de pontos de função, funcionalidade refere-se à capacidade do software interagir com o usuário e armazenar os seus dados. Existem regras, procedimentos e práticas definidos no *Counting Practices Manual* (CPM) visando identificar as funções do software que oferecem essas capacidades e a ponderar numericamente cada uma delas conforme o seu tipo e complexidade. O termo função e funcionalidade podem ser considerados sinônimos nesse contexto.
- *Medir o desenvolvimento e a manutenção de software de forma independente da tecnologia utilizada para sua implementação.* As regras, procedimentos e práticas definidos no CPM, tanto para identificação das funções quanto para a determinação de sua complexidade, têm o princípio comum

de utilizar como referência a visão do usuário - a perspectiva do negócio em que o software está inserido.

Além destes objetivos, o processo de contagem de pontos de função deve ser:

- *Simples o suficiente para minimizar o trabalho adicional envolvido no processo de medição.* As regras e procedimentos de contagem observam esse princípio de simplicidade. O desenvolvimento de software envolve tanta complexidade que aqueles que realizam a contagem e que têm origem técnica acabam tendo mais dificuldade que aqueles que realizam a contagem e vêm de uma área usuária.
- *Uma medida consistente entre vários projetos e organizações.* Este objetivo do processo de contagem refere-se tanto à reprodutibilidade de uma medição quanto ao significado associado à medição por diferentes pessoas em diferentes projetos e organizações.

## 2.2 A REFERÊNCIA PARA MEDIÇÃO: O USUÁRIO E A VISÃO DO USUÁRIO

A identificação de qualquer tipo de funcionalidade é feita com base na fronteira da aplicação. Essa por sua vez é definida numa perspectiva do usuário e, portanto, é importante que o analista de pontos de função tenha sempre o conceito de usuário em foco ao realizar uma medição.

### 2.2.1 USUÁRIO

Usuário para APF significa:

- Qualquer pessoa que especifique requisitos funcionais; ou
- Qualquer pessoa, sistema ou componente que interaja com o sistema a qualquer momento.

Com base nessa definição, é correto dizer que para uma determinada aplicação a ser medida temos como seus usuários, por exemplo: os gestores do negócio que o software atenderá (*stakeholders*); outras aplicações relacionadas ou componentes de *hardware* que recebem ou enviam dados para ela; a pessoa física que o usa; seus usuários finais; agentes governamentais (exigências regulatórias do governo normalmente abrangem boa parte dos requisitos de um *software*) ou qualquer pessoa responsável pela definição formal dos requisitos.

O conceito de ator em um caso de uso é uma boa aproximação do conceito de usuário (que interage com o sistema) para a APF. Um ator não está restrito a ser uma pessoa física, podendo ser outra aplicação ou mesmo algum componente de *hardware*, desde que também interaja com o sistema. Se a contagem de pontos de função fosse baseada apenas no conceito de usuário como sendo uma pessoa física, não seria possível medir sistemas que não têm interface com o usuário final. No entanto, a técnica pode ser aplicada também nestas situações.

Levando em consideração essa amplitude do conceito de usuário, durante uma contagem de pontos de função, convém buscar no conjunto de usuários possíveis aquele(s) cuja visão melhor representa as funções que a aplicação fornece.

Por exemplo, o sistema *Detran MG* (<http://www.detran.mg.gov.br/>) tem como usuários os proprietários de veículos, proprietários de carteira de habilitação, interessados em ter habilitação, interessados em comprar determinado veículo etc. Basear a contagem dessa aplicação somente na visão dos usuários finais é ter uma visão limitada da aplicação. É fundamental levar em consideração também a visão do

usuário que especifica os requisitos e as regras de negócio, neste caso, funcionário do Detran e também a Secretaria da Fazenda, que define as regras de pagamentos referentes a ela.

## 2.2.2 VISÃO DO USUÁRIO

A visão do usuário é a descrição formal das necessidades de negócio do usuário em sua própria linguagem de negócio. Os desenvolvedores têm o papel de traduzir essa informação em uma linguagem técnica necessária ao provimento de uma solução de tecnologia. Ela é usada para contar pontos de função, desde que seja uma descrição das suas funções de negócio e também seja aprovada pelo usuário. A visão do usuário é representada por diversos artefatos, alguns deles gerados pelo desenvolvedor durante a especificação dos requisitos: documento de visão, especificação de requisitos, casos de uso, modelo de dados (conceitual ou modelo de classes).

Em termos mais objetivos, podemos dizer que a visão do usuário contrasta com uma visão técnica. Ao realizar a análise de pontos de função, não se deve assumir que uma tabela em banco de dados ou arquivo no sistema operacional, por exemplo, equivalha necessariamente a um arquivo na perspectiva do usuário. É possível, e até comum, existirem tabelas ou arquivos incluídos na solução tecnológica que não sejam individualmente contados como arquivos quando da análise de pontos de função, onde a visão do usuário é a referência. Analogamente, devem-se observar os diferentes formulários em papel usados pelos usuários em seus processos de negócio. Várias telas podem ser usadas para preencher um único formulário visando uma melhor apresentação, mas na visão do usuário existe apenas uma transação para o preenchimento do mesmo.

Entradas de dados para o sistema em que esses são recebidos por diferentes meios como arquivos de passagem de movimento ou digitação em tela e que independentemente do meio as entradas têm a mesma lógica de processamento, idênticas na visão do usuário, tipicamente correspondem a uma única transação na análise de pontos de função. O oposto também ocorre, um único relatório ou arquivo de movimento gerado pelo sistema pode corresponder a várias transações na visão do usuário.

Para facilitar essa distinção entre aquilo que é ou não percebido pela visão do usuário, pela visão dos processos de negócio do usuário, é conveniente classificar os tipos de requisitos dos sistemas e, consequentemente, dos projetos que os desenvolvem e os mantêm.

É importante observar que a Prodemge desenvolve também sistemas corporativos, que muitas vezes tem várias categorias de usuários, com visões de negócio distintas. Ela também desenvolve aplicações com arquitetura de serviço (*web services*) cujos usuários em potencial, muitas vezes, ainda não estão totalmente definidos. Nestes contextos, para definir a visão do usuário, é importante utilizar a definição do IFPUG para ponto de função:

*“Pontos de função medem o tamanho do software através da quantificação da funcionalidade provida ao usuário baseando-se somente em desenhos lógicos e especificações funcionais”.*

*(Function Points measure software size by quantifying the functionality provided to the user based solely on logical design and functional specifications).*

Fonte: [http://www.ifpug.org/?page\\_id=12](http://www.ifpug.org/?page_id=12).

Desta definição, podemos retirar a conclusão de que a técnica de APF IFPUG quantifica funcionalidades **disponibilizadas para o usuário**, não sendo estritamente necessário que elas tenham sido definidas por ele. Obviamente, as funcionalidades especificadas devem atender às necessidades do usuário e devem, sempre que possível, ser aprovadas por ele. Na prática, essa interpretação é necessária para



que a técnica de APF possa ser aplicada em contextos onde o usuário final não participa (ou participa apenas indiretamente) da especificação de requisitos.

Esta definição também esclarece que **a medição é baseada nas especificações funcionais**. Para a Prodemge, isto significa que a visão do usuário está modelada (ou traduzida) nas especificações de requisitos. Assim, é através da interpretação dos itens modelados que a visão do usuário deve ser determinada.

Por fim, como a Prodemge é muitas vezes a responsável por traduzir em requisitos as necessidades de vários clientes (ou usuários) com necessidades de negócio distintas, é do melhor interesse do Estado, especialmente nos casos de subcontratação total ou parcial do desenvolvimento, que o tamanho funcional represente da melhor forma possível a correlação com o esforço de implementação a ser desenvolvido pela Prodemge ou por uma eventual subcontratada. Portanto, nos casos em que a definição da visão do usuário trazer impactos nos custos do software, sem que haja justificativa em termos de esforço para a adoção de uma definição que implique no aumento do tamanho funcional, prevalecerá o entendimento de que a visão do usuário é aquela que pode ser determinada pelas funcionalidades a serem desenvolvidas e entregues, de acordo com a especificação de requisitos.

## 2.3 TIPOS DE REQUISITOS DE SOFTWARE

A norma ISO/IEC 1443 distingue três tipos de requisitos do usuário:



**Figura 1: Tipos de requisitos conforme norma ISO/IEC 1443**

Apesar de a APF se propor a medir apenas os requisitos funcionais, os outros tipos de requisitos também são importantes e devem sempre ser considerados numa estimativa de esforço ou custo. Assim, num contrato de desenvolvimento e manutenção de software por PF, o preço (R\$/PF) deve contemplar todo o cenário de requisitos não funcionais que deverão ser atendidos.

Com o objetivo de estabelecer uma referência para derivações de esforço, prazos e custos para aspectos do software não mensurados pela técnica de APF, a Prodemge adota uma tabela de itens não mensuráveis (ver seção 6).

A visão do usuário não é estática e na medida em que o ciclo de vida de uma aplicação passa pelos seus diversos estágios, a sua medição é alterada. *Contagens anteriores à especificação completa de requisitos funcionais são **estimativas** do tamanho medido nessa ocasião.*

### 2.3.1 EXEMPLOS DE CADA TIPO DE REQUISITO DE SOFTWARE

A seguir alguns exemplos de requisitos de software:

- Requisito Funcional: Caso de uso Gerir funcionário, Caso de uso Apurar frequência, Caso de uso Apurar resultado do concurso.
- Requisito de Qualidade:
  - Usabilidade: paginação, ordenação, calendário pop-up.
  - Desempenho: tempo de resposta para uma funcionalidade.
  - Confiabilidade: Segurança acesso, segurança dos dados.
- Requisito Técnico: versão do navegador em que o sistema irá executar.

De acordo com o padrão ISO revisado (ISO/IEC 14143-1:2007), base da versão 4.3 do Manual de Contagem, os requisitos de Qualidade e Técnicos são agora chamados de Requisitos não funcionais.

### 2.4 TIPOS DE ENTIDADES DE DADOS

A Análise de Pontos de Função trabalha com três tipos de dados:

- Dados do Negócio
- Dados de Referência
- Dados de Código

As primeiras duas categorias de entidades são normalmente identificadas para satisfazer os *Requisitos Funcionais do Usuário*, e como tais, estas entidades serão analisadas para contagem como arquivos lógicos.

A terceira categoria de dados, apresentada neste capítulo como “Dados de Código”, existe normalmente para satisfazer requisitos técnicos mais propriamente do que requisitos funcionais. As diferentes categorias de dados estão descritas a seguir para auxiliar na identificação.

#### 2.4.1 DADOS DE NEGÓCIO

Dados de Negócio podem também ser referenciados como Dados Centrais do Usuário (“*Core User Data*”) ou Objetos de Negócio. Estes tipos de dados refletem a informação necessária para ser guardada e recuperada pela área funcional tratada pela aplicação. Dados de Negócio normalmente representam um percentual significativo das entidades identificadas. A maior parte tem as seguintes características:

- Lógicas:
  - Obrigatório para a operação da área funcional do usuário;
  - Identificável pelo usuário (normalmente por um usuário do negócio);
  - Mantido pelo usuário (normalmente por um usuário do negócio);
  - Armazenam Dados Centrais do Usuário para auxiliar as transações do negócio
  - Muito dinâmico – operações normais do negócio fazem com que eles sejam regularmente referenciados, incluídos, alterados e excluídos rotineiramente.
  - Reportável.
- Físicas:
  - Têm campos chave e normalmente muitos atributos
  - Podem ter de zero a inumeráveis registros



- Exemplos:
  - Arquivo de Cliente, Arquivo de Nota Fiscal, Arquivo de Funcionário, Arquivo de Função.
  - Arquivo de Função, dentro do Sistema de Gerenciamento, deveriam incluir itens como:
    - Número da Função
    - Nome da Função
    - Nome da Divisão
    - Data de Início na Função, etc.

## **2.4.2 DADOS DE REFERÊNCIA**

Dados deste tipo são armazenados para auxiliar as regras de negócio para a manutenção dos Dados de Negócio. Ex.: em uma aplicação de Folha de Pagamento eles devem ser os dados armazenados de Taxas de Impostos Governamentais para cada faixa salarial e a data em que a taxa do imposto se tornou efetiva. Normalmente os Dados de Referência representam um pequeno percentual das entidades identificadas. A maior parte tem as seguintes características:

- Lógicas:
  - Obrigatório para a operação da área funcional do usuário
  - Identificável pelo usuário (normalmente por um usuário do negócio)
  - Normalmente mantido pelo usuário (normalmente por um usuário administrativo)
  - Normalmente criado quando a aplicação é instalada pela primeira vez e mantido intermitentemente
  - Armazena os dados para auxiliar nas atividades centrais do usuário
  - Pouco dinâmico – ocasionalmente altera em resposta a mudanças no ambiente das áreas funcionais, processos funcionais externos e/ou regras de negócio.
  - Transações processando Dados de Negócio frequentemente necessitam acessar os Dados de Referência
- Físicas:
  - Têm campos chave e muitos atributos.
  - Normalmente pelo menos um registro ou um número limitado de registros.
- Exemplos:
  - Tarifas da Função, Tarifas Abatidas, Tarifas dos Impostos, Dissídio.
  - Arquivo de Tarifas da Função – armazena informações sobre as tarifas a serem pagas para cada tipo de função e a habilidade exigida para aquele tipo de função
    - Tipo da Função
    - Situação, Tarifa do Cargo, Data Efetiva (1:n)
    - Descrição da habilidade da Função (1:n)

## **2.4.3 DADOS DE CÓDIGO**

O usuário nem sempre especifica diretamente os Dados de Códigos, às vezes chamados de Lista de Códigos ou Dados de Tradução. Em outros casos ele é identificado pelo desenvolvedor em resposta a um ou mais requisitos técnicos do usuário. Dados de Código fornece uma lista de valores válidos que um atributo descritivo pode ter. Tipicamente os atributos dos Dados de Códigos são Códigos, Descrição e/ou outros atributos padrão descrevendo o código. Ex.: abreviação comum, data inicial, data final, dados de trilha de auditoria etc.

Quando os códigos são usados em Dados do Negócio, é necessário ter um meio de tradução para converter o código em alguma coisa mais familiar ao usuário. A fim de satisfazer requisitos técnicos, os desenvolvedores muitas vezes criam uma ou mais tabelas contendo os Dados de Código. Logicamente,

o código e sua referida descrição têm o mesmo significado. Sem uma descrição, o código nem sempre pode ser entendido claramente.

A diferença chave entre Dados de Código e Dados de Referência é:

- Com Dados de Código, você pode substituir um pelo outro sem alteração do significado dos Dados do Negócio. Ex.: Código do Aeroporto X Nome do Aeroporto, Código da Cor X Descrição da Cor.
- Com Dados de Referência você não pode substituir (Ex.: Código do Imposto com a Alíquota do Imposto).

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

A maior parte dos *Dados de Código* tem as seguintes características:

- Lógicas:
  - Dados são obrigatórios para a área funcional, mas armazenado opcionalmente como um arquivo de dados.
  - Geralmente não identificado como parte dos requisitos funcionais; ele é normalmente identificado como parte do projeto para satisfazer requisitos técnicos;
  - Às vezes mantidos pelo usuário (normalmente por um usuário do suporte);
  - Armazenam dados para padronizar e facilitar atividades do negócio e transações do negócio;
  - Essencialmente estático – apenas alterado em resposta a mudanças na maneira que se opera o negócio;
  - Transações do negócio acessam Dados de Código para melhorar casos de entradas de dados, melhorar a consistência de dados, garantir integridade de dados, etc.;
  - Se reconhecido pelo usuário:
    - Às vezes é considerado como um grupo do mesmo conjunto de dados;
    - Pode ser mantido utilizando a mesma lógica de processamento
- Físicas:
  - Constitui-se de campos chave e normalmente um ou dois atributos apenas (código e descrição).
  - Tipicamente tem um número estável de registros.
  - Às vezes desnormalizado e armazenado em uma tabela física com outros Dados de Código.
  - Pode ser implementado de diferentes formas (ex.: em uma aplicação separada, dicionário de dados ou *hard-coded* dentro de um software).
- Exemplos:
  - Estado:
    - Código do Estado
    - Descrição do Estado
  - Tipo de Pagamento:
    - Código do Tipo de Pagamento
    - Descrição do Tipo de Pagamento

## 2.5 O QUE É O 'F' DA APF: O QUE É FUNÇÃO

A análise de pontos de função “quebra” o sistema em partes (denominadas funções ou funcionalidades) que permitem ao usuário - ou em outras palavras - que fornecem uma função ao usuário de:

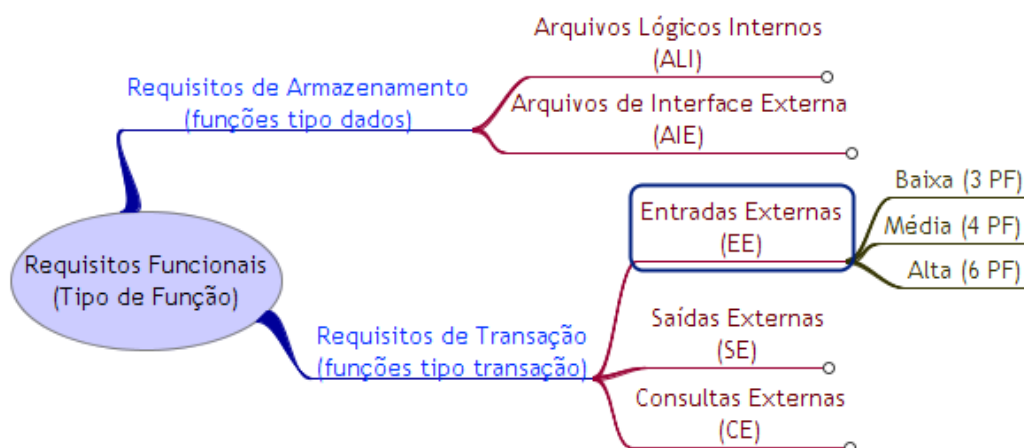
- Interação com o sistema (Requisitos de Transação);
- Armazenamento dos seus dados (Requisitos de Armazenamento).

As funcionalidades do sistema são, portanto, classificadas em: **funções tipo dados e tipo transação**. Cada um desses tipos é adicionalmente detalhado em tipos mais específicos. O diagrama abaixo ilustra essa hierarquia de classificação de funções:



**Figura 2:** Hierarquia de funções na APF

Depois de identificada, cada função deve ser enquadrada como um dos tipos de função apresentados acima e ter a sua complexidade funcional avaliada como baixa, média ou alta. Conforme seu tipo de função e a sua complexidade funcional, a funcionalidade recebe a sua contribuição à contagem do tamanho funcional do projeto ou aplicação. O diagrama abaixo ilustra, por exemplo, a contribuição das funções classificadas como entradas externas.



**Figura 3:** Exemplo de contribuição de função na APF

A forma como o usuário entende o *software*, como ele enxerga o *software* como uma unidade, é fundamental nesse processo. É a referência para a identificação de todos os tipos de função. No contexto das regras de contagem, essa referência é materializada na forma da fronteira da aplicação.

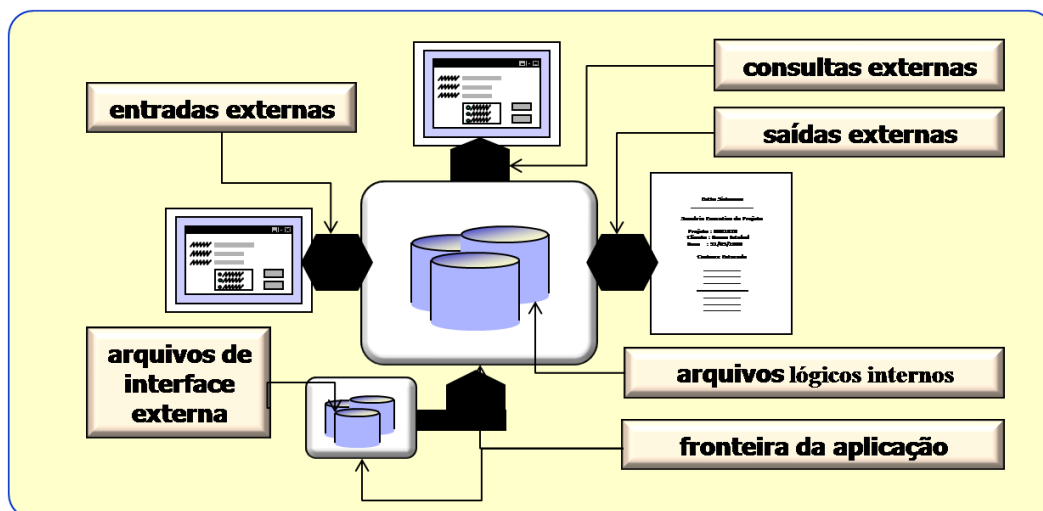


Figura 4: Funções de transações e dados x Fronteira da aplicação

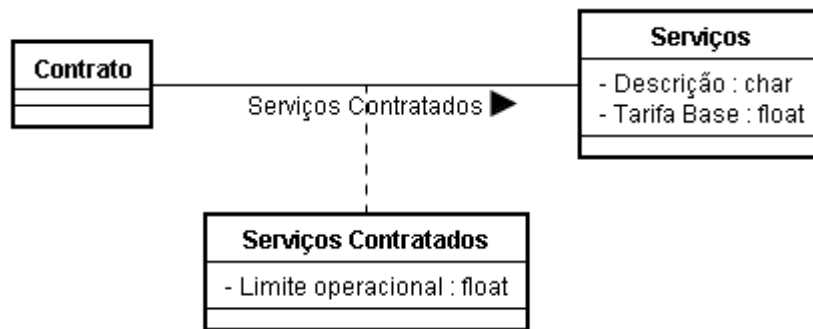
### 2.5.1 REQUISITOS DE ARMAZENAMENTO FUNCIONAL (ALI/AIE)

Um determinado requisito de armazenamento de dados será classificado como arquivo lógico interno (ALI) quando existir algum processo compreendido por essa fronteira da aplicação que os mantenham e como arquivo de interface externa (AIE) quando for também um arquivo lógico interno em outra aplicação, mas seja apenas lido por processos compreendidos pela fronteira da aplicação em análise.

As funções tipo dado são identificadas usando como critério a forma como o usuário agrupa os dados que o sistema armazena para ele. Fazer uma analogia com o conceito de **documento** ou **formulário** é uma boa forma de entender a lógica desse agrupamento na análise de pontos de função. Por exemplo, um documento como o pedido de compra possui uma série de dados relacionados entre si. Mesmo que esses dados sejam armazenados em três tabelas em um banco de dados relacional, ainda assim para a análise de pontos de função o seu conjunto como um todo é considerado um grupo logicamente relacionado de dados. As seguintes perguntas dão apoio na avaliação se um grupo de entidades deve ser contado como um único arquivo lógico:

- Do ponto de vista do negócio, apenas um documento ou formulário corresponde aos dados mantidos nas diferentes entidades em análise?
- Existe uma relação de dependência entre essas diferentes entidades? Se os dados de uma entidade forem retirados do sistema, os dados da outra ainda assim têm significado para o negócio?
- Usando a metáfora do arquivo como um móvel que guarda documentos, os dados dessas entidades seriam guardados no mesmo armário ou em armários distintos? Imagine como é (ou seria) o procedimento manual.

Observe que muitas vezes dois documentos de negócio independentes – ainda que inter-relacionados entre si – têm uma dependência introduzida pela modelagem dos dados. Por exemplo, a relação entre “serviços contratados” e “serviços”:



**Figura 5: Exemplo de requisitos de armazenamento funcional**

No negócio em questão existe o contrato (dado de negócio) e a tabela de serviços (dado de referência). Como resultado da modelagem de sistemas, será definida a entidade “serviços contratados” dependente de contrato e de serviços. Na análise de pontos de função, dois arquivos lógicos são identificados, porém os dados de “serviços contratados” são contabilizados apenas junto aos dados de “contrato”.

Outra diretriz geral para identificação dos arquivos lógicos é sempre pensar no **modelo conceitual** dos dados. Se este for um artefato disponível pelo processo de desenvolvimento, a contagem de ALI e AIE fica facilitada. Cada entidade no modelo conceitual corresponderá a um ALI ou AIE.

Por fim, em um diagrama de fluxo de dados, os arquivos lógicos corresponderiam aos depósitos de dados e as transações (vide próxima seção) corresponderiam aos fluxos de dados.

As definições para o Arquivo Lógico Interno e o Arquivo de Interface Externa são:

**Arquivo Lógico Interno (ALI):** *um grupo logicamente relacionado de dados ou informações de controle, identificável pelo usuário, mantido dentro da fronteira da aplicação sendo contada. Sua principal intenção é armazenar dados mantidos através de uma ou mais transações da aplicação sendo contada. Exemplo: tabelas de banco de dados atualizadas pela aplicação.*

**Arquivo de Interface Externa (AIE):** *um grupo logicamente relacionado de dados ou informações de controle, identificável pelo usuário, mantidos fora da fronteira da aplicação sendo contada. Sua principal intenção é armazenar dados referenciados através de uma ou mais transações da aplicação sendo contada. Exemplo: tabelas de banco de dados lidas pela aplicação, mas atualizadas por outra aplicação.*

### 2.5.1.1 REGRA DE COMPLEXIDADE FUNCIONAL PARA ARQUIVOS

O número de ALIs, AIEs e suas respectivas complexidades funcionais determinam a contribuição das funções de dados para a contagem de pontos de função. A complexidade funcional de cada ALI e AIE é determinada com base na quantidade de tipos de dados elementares (TD ou DER) e de tipos de registros elementares (TR ou RLR) associados ao mesmo.

As definições para tipo de dado elementar e tipo de registro elementar são:

**Tipo de Dado Elementar (TD ou DER):** *Um tipo de dado elementar é um campo único, reconhecido pelo usuário e não repetido.*

**Tipo de Registro Elementar (TR ou RLR):** é um subgrupo de dados reconhecido pelo usuário dentro de um ALI ou um AIE.

Existem dois tipos de subgrupos: **subgrupos opcionais** são aqueles que o usuário tem a opção de usar um ou nenhum dos subgrupos durante o processo elementar que inclui ou cria uma instância do dado; **subgrupos obrigatórios** são subgrupos onde o usuário deve usar pelo menos um.

## 2.5.2 REQUISITOS DE TRANSAÇÃO (EE/SE/CE)

Os requisitos de transação do sistema para serem mapeados para uma função tipo transação devem obrigatoriamente receber e/ou enviar dados ou informação de controle pela fronteira da aplicação. Para esse fim podemos entender a fronteira da aplicação como uma tela ou um relatório por exemplo. As transações representam os requisitos de processamento do usuário.

São classificadas em entradas externas (EE), consultas externas (CE) e saídas externas (SE). O critério fundamental para a sua identificação é atender ao conceito de **processo elementar: menor unidade de atividade significativa para o usuário, que seja completa em si mesma e deixe a aplicação em um estado consistente**. Considerando essas explicações seguem as definições de Entrada Externa, Saída Externa e Consulta Externa:

**Entrada Externa (EE):** é um processo elementar que processa dado ou informações de controle originadas de fora da fronteira da aplicação. Sua principal intenção é manter um ou mais arquivos lógicos internos e/ou alterar o comportamento do sistema. Exemplo: incluir cliente, alterar cliente, excluir cliente.

**Saída Externa (SE):** é um processo elementar que envia dado ou informações de controle para fora da fronteira da aplicação. Sua principal intenção é apresentar informação ao usuário através de lógica de processamento que não seja apenas uma simples recuperação de dados ou informações de controle. Seu processamento deve, obrigatoriamente, conter algum cálculo ou fórmula matemática, ou criar dados derivados, ou manter um arquivo lógico interno, ou alterar o comportamento do sistema. Exemplo: relatório de totais de faturamento por cliente.

**Consulta Externa (CE):** é um processo elementar cuja principal intenção é apresentar informação ao usuário pela simples recuperação de dados ou informações de controle de arquivos lógicos internos (ALI) ou arquivos de interface externa (AIE). Sua lógica de processamento não contém fórmula matemática ou cálculos, não cria dado derivado, não mantém arquivo lógico interno (ALI) durante o processamento nem modifica o comportamento do sistema.

### 2.5.2.1 REGRA DE COMPLEXIDADE FUNCIONAL PARA PROCESSOS ELEMENTARES

O número de processos elementares (EE, SE e CE) e suas relativas complexidades funcionais determinam a contribuição das funções de transação para a contagem de pontos de função. A complexidade funcional de um processo elementar é definida com base no número de tipos de arquivos referenciados (AR ou ALR) e tipos de dados elementares (TD ou DER).

Definições:



**Tipo de arquivo referenciado (AR ou ALR)** é um arquivo lógico interno lido ou mantido por uma função de transação ou um arquivo de interface externa lido por uma função de transação

**Tipo de dado elementar (TD ou DER)** é um campo único, reconhecido pelo usuário e não repetido.

## 2.6 JUNTANDO AS PONTAS: O PROCESSO DE MEDIÇÃO

### 2.6.1 PROPÓSITO DA CONTAGEM

Ao realizar uma medição de pontos de função, a primeira providência a se tomar é descobrir o propósito daquele trabalho. É descobrir **qual a pergunta de um problema de negócio se quer responder** com aquela contagem. Esse propósito é o que definirá qual o tipo e o escopo da contagem, assim como influenciará no posicionamento da fronteira da aplicação.

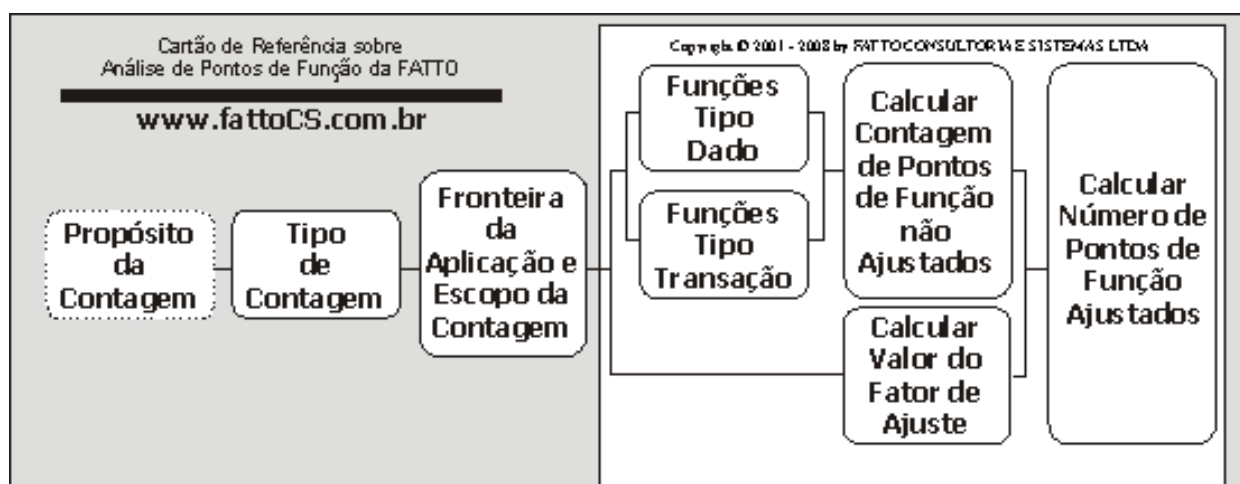


Figura 6: Processo de medição

### 2.6.2 TIPO DE CONTAGEM E ESCOPO DA CONTAGEM

O passo seguinte é determinar o tipo daquela contagem. São três os tipos de contagem: Projeto de Desenvolvimento, Projeto de Melhoria e Aplicação (contagem da base instalada, ou baseline são sinônimos). A determinação do tipo de contagem basicamente tem impacto em quais funcionalidades da aplicação (ou de conversão de dados) serão incluídas em determinada contagem - o escopo da contagem.

Por exemplo, na contagem de um projeto de desenvolvimento, todas as funcionalidades construídas ou customizadas serão incluídas no escopo junto com toda funcionalidade de conversão de dados necessária à implantação do sistema. Já na contagem de um projeto de melhoria, além dessas funcionalidades de conversão de dados, as funcionalidades incluídas, alteradas ou excluídas pelo projeto também estão no escopo da contagem. O projeto de melhoria visa medir aquelas manutenções classificadas como manutenção adaptativa. A contagem da aplicação não inclui em seu escopo as funções de conversão de dados, mas dependendo do propósito pode incluir todas as funcionalidades da aplicação ou apenas aquelas em uso quando da contagem.

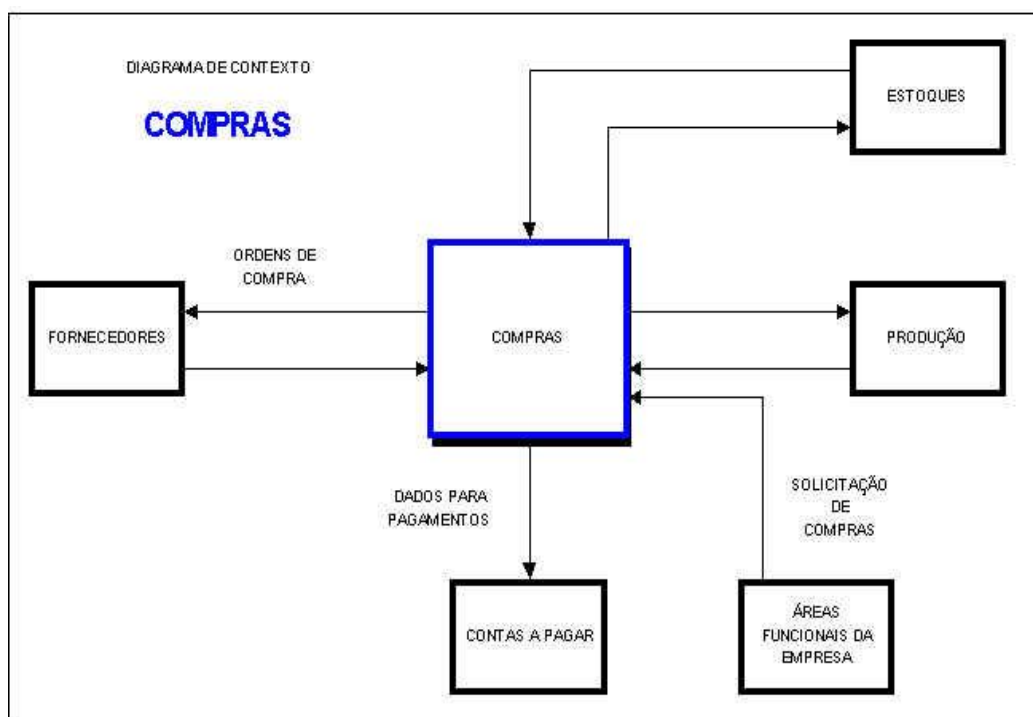
### 2.6.3 FRONTEIRA DA APLICAÇÃO

É uma premissa fundamental para a contagem de pontos de função que se estabeleça o limite entre a aplicação e o que é externo a ela. Esse processo é a definição da fronteira da aplicação. Um erro no estabelecimento da fronteira leva todo o trabalho posterior da contagem a ser inutilizado. Ela é definida com base no ponto de vista de negócio, não em considerações técnicas (por exemplo, arquitetura do sistema). Geralmente a fronteira da aplicação costuma estar bem explícita, não havendo dificuldade para sua definição.

Um artefato que representa o conceito da fronteira da aplicação é o diagrama de contexto (diagrama de caso de uso, DFD nível zero, DFD de nível mais alto). Ele representa todo o sistema como um único processo e é composto por fluxos de dados (no DFD) ou relacionamentos entre atores e casos de uso (Diagrama de caso de uso) que mostram as interfaces entre o sistema e as entidades externas. O diagrama é uma forma de representar o objeto do estudo, o projeto, e sua relação com o ambiente.

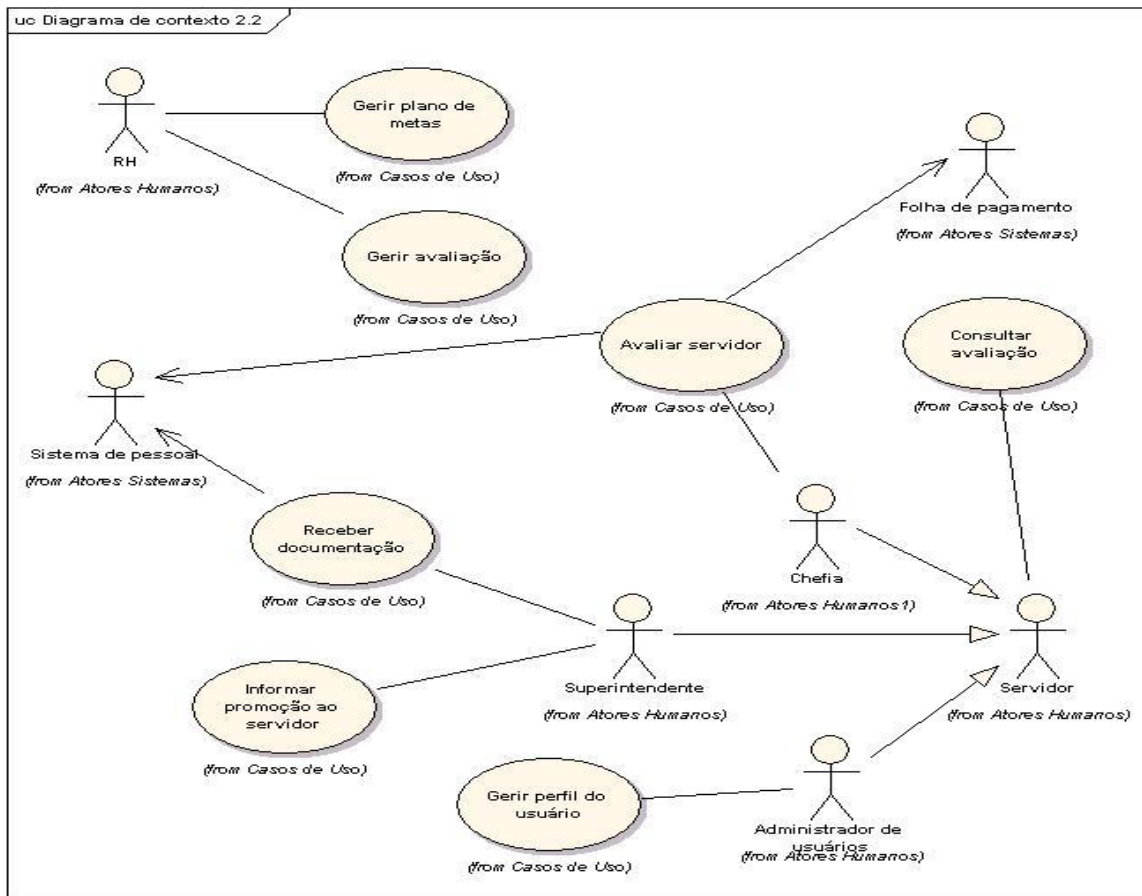
Um diagrama de contexto permite identificar os limites dos processos, as áreas envolvidas com o processo e os relacionamentos com outros processos e elementos externos à empresa (ex.: clientes, fornecedores).

- **Exemplo 1**



**Figura 7: Exemplo de Diagrama de Contexto – Análise Essencial**

- **Exemplo 2**



**Figura 8: Exemplo de Diagrama de Contexto - UML**

Um erro muito comum é pensar na fronteira com base na implementação física do sistema, como por exemplo, separar as fronteiras pelo número de camadas do sistema, ou pela quantidade de programas executáveis. Esta não deve ser a diretriz.

### 2.6.3.1 DIRETRIZES PARA ESTABELECEER A FRONTEIRA DA APLICAÇÃO

Há algumas dicas que podem ajudar na identificação da fronteira:

- Obter uma documentação do fluxo de dados no sistema e traçar uma linha em volta para destacar quais partes são internas e externas à aplicação;
- Verificar como os grupos de dados são mantidos;
- Identificar áreas funcionais pela atribuição de propriedade de certos objetos de análise, como entidades e processos;
- Comparar os critérios utilizados em outras métricas como esforço, custo, duração, defeitos. As fronteiras para a APF e as outras métricas deveriam ser as mesmas;
- Verificar como a aplicação é gerenciada; se é desenvolvida ou mantida na sua totalidade por uma equipe única ou equipes distintas;
- Verificar se o software possui ordens de serviços específicas e independentes;
- Se há usuários distintos especificando requisitos para cada parte do software.

O conceito de fronteira é tão importante que o ideal é que se estabeleça de antemão um inventário das aplicações da organização. A ideia é apenas identificar e listar todas elas (o

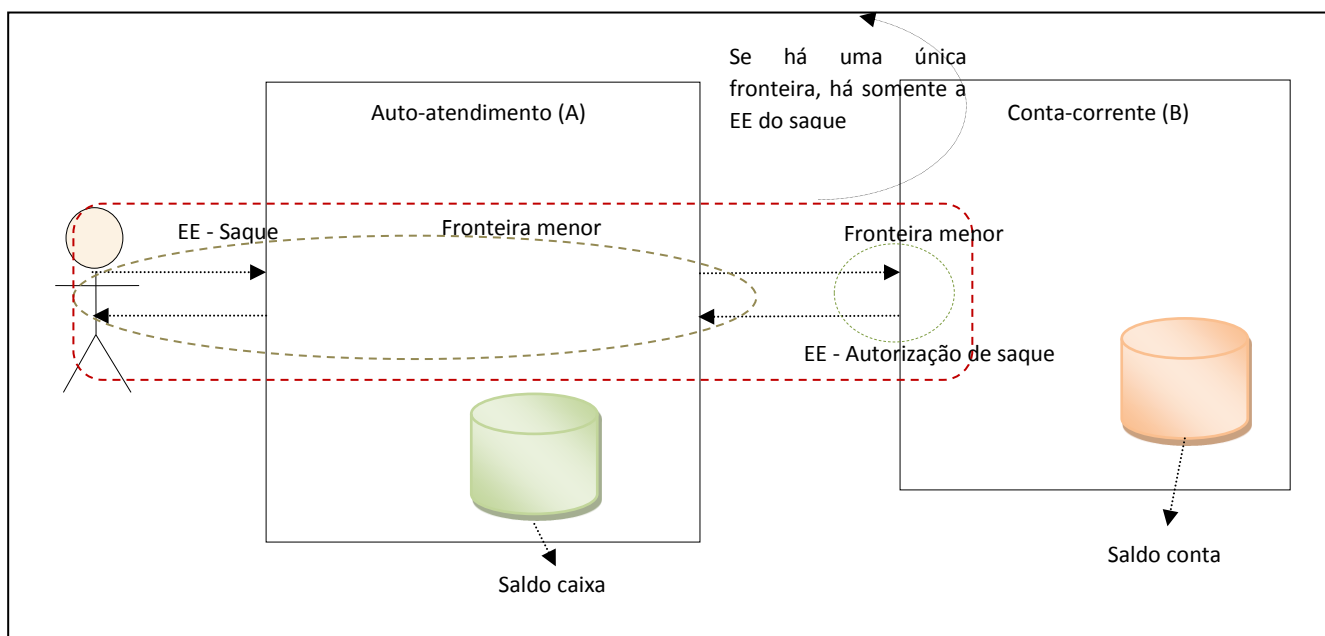
inventário das funções de cada aplicação também é interessante). Todas as contagens de pontos de função realizadas devem usar como base as fronteiras estabelecidas previamente. A fronteira é um conceito que dificilmente muda entre as contagens de pontos de função. Ou seja, as diversas medições dos projetos de melhoria na aplicação, terão como base a mesma fronteira.

### 2.6.3.2 PROBLEMAS NO ESTABELECIMENTO INCORRETO DA FRONTEIRA DA APLICAÇÃO

No caso de uma fronteira posicionada de maneira incorreta, por exemplo, um sistema com dois componentes, mas cada um deles sendo tratado como uma aplicação na APF, o somatório dos PFs de cada componente será bem superior ao total de PFs que haveria se houvesse uma única fronteira para os dois componentes.

- **Exemplo**

Se considerarmos o sistema de autoatendimento (**Erro! Fonte de referência não encontrada.**) e o sistema de conta-corrente como sistemas diferentes (duas fronteiras de aplicação), uma operação de saque contaria como duas entradas: a 1ª no Autoatendimento que referencia 1 AIE e a outra no Conta-corrente. Se for somente uma aplicação, tem-se um único processo elementar que acessa 1 ALI.



**Figura 9: Exemplo de Fronteira da Aplicação**

Pistas de posicionamento incorreto da fronteira:

- Aplicação com número excessivo de AIEs de outro sistema. Isto indica alto grau de acoplamento com o outro sistema, indicando que provavelmente a “aplicação” é apenas um módulo do outro sistema.
- Manutenções conjuntas: quando um sistema sempre sofre manutenção junto com outro, é também uma indicação de que ele não tem “força” para ser uma aplicação independente.

## 2.7 CÁLCULO DOS PONTOS DE FUNÇÃO

Uma vez definidas essas premissas, o analista de pontos de função pode realizar em qualquer ordem uma das três atividades:

- Medir as funções tipo dados;
- Medir as funções tipo transação; e/ou
- Determinar o valor do fator de ajuste.

As duas primeiras consistem na identificação, na determinação da complexidade funcional e no cálculo da contribuição de cada função no escopo da determinação do tamanho funcional.

### 2.7.1 CONTAGEM DAS FUNÇÕES TIPO DADO E DAS FUNÇÕES TIPO TRANSAÇÃO

Nesse passo são identificadas todas as funções tipo dado (ALI/AIE) e funções tipo transação (EE/SE/CE) do sistema. As complexidades são determinadas e associadas. A cada complexidade existe uma quantidade de pontos de função correspondente (contribuição ao tamanho funcional). Vide tabela abaixo:

Complexidade Funcional				Contribuição			
<b>ALI e AIE</b>	TR \ TD	<20	20-50	>50			
	1	Baixa	Baixa	<b>Média</b>	Arquivo Lógico Interno	7 PF	10 PF
	2-5	Baixa	<b>Média</b>	Alta	Arq. de Interf. Externa	<b>5 PF</b>	<b>7 PF</b>
	>5	<b>Média</b>	Alta	Alta	Entrada Externa	3 PF	4 PF
<b>EE</b>	AR \ TD	<5	5-15	>15	Saída Externa	<b>4 PF</b>	<b>5 PF</b>
	<2	Baixa	Baixa	<b>Média</b>	Consulta Externa	3 PF	4 PF
	2	Baixa	<b>Média</b>	Alta			
	>2	<b>Média</b>	Alta	Alta			
<b>SE e CE*</b>	AR \ TD	<6	6-19	>19			
	<2*	Baixa	Baixa	<b>Média</b>			
	2-3	Baixa	<b>Média</b>	Alta			
	>3	<b>Média</b>	Alta	Alta			

\* A CE deve referenciar ao menos 1 ALI ou AIE

**Tabela 1: Complexidade Funcional das Funções de Dados e de Transação**

Por exemplo, se o processo de contagem identifica um determinado grupo de dados como um arquivo lógico interno (ALI) e determina que o mesmo tem 5 tipos de registros elementares (TR ou RLR) e 47 tipos de dados elementares (TD ou DER), então, de acordo com a primeira tabela acima, esse ALI é de complexidade média. Por conseguinte, pela tabela de Contribuição, ele contribui com 10 pontos de função para o tamanho funcional da aplicação.

Da mesma forma, se o processo de contagem identifica um determinado processo elementar como uma saída externa (SE), com 4 arquivos referenciados (AR ou ALR) e 16 tipos de dados elementares (TD ou DER) referenciados, então, de acordo com a última tabela acima à esquerda, esse processo elementar é

de complexidade alta. Por conseguinte, pela tabela de Contribuição, ele contribui com 7 pontos de função para o tamanho funcional da aplicação.

## **2.7.2 VALOR DO FATOR DE AJUSTE**

Pelo CPM, o cálculo do valor do fator de ajuste (VAF) definido pelo IFPUG é feito pela ponderação de 14 características gerais de sistema quanto ao seu nível de influência no projeto ou na aplicação: de nenhuma influência até uma grande influência. O manual de práticas de contagem oferece uma série de orientações específicas para cada característica geral de sistema visando uma maior uniformidade na determinação do respectivo nível de influência. A técnica define uma fórmula tal que o valor do fator de ajuste varia entre 0,65 e 1,35, ou seja, produz uma variação de +/- 35% no tamanho do sistema.

O valor do fator de ajuste pode ser sempre calculado. Entretanto, em diversas situações ele não é aplicado para ajustar a contagem por diversas razões, entre elas a aderência estrita aos padrões definidos pela ISO/IEC para mensuração de software (a norma ISO/IEC 10926 só reconhece a APF como uma medida de software válida se não for aplicado o fator de ajuste), razões de mercado e também para que a contagem esteja em conformidade com as definições incluídas em editais ou contratos.

**IMPORTANTE:** Na Prodemge adotamos 1 (um) como valor do fator de ajuste, o que equivale a dizer que não se utiliza fator de ajuste.

## **3. METODOLOGIA DA NESMA PARA CONTAGEM DE MELHORIA**

Seis passos são necessários para determinar o escopo e o tamanho (em pontos de função de melhoria) para projetos de melhoria:

1. Identificar as funções de dados e de transações dentro do escopo do projeto de melhoria e determinar seu tamanho funcional.
2. Determinar as funções de dados e de transação a serem adicionadas
3. Determinar as funções de dados e de transação a serem excluídas.
4. Determinar as funções de dados a serem modificadas e determinar o fator de impacto
5. Determinar as funções de transação a serem modificadas e determinar o fator de impacto
6. Calcular os Pontos de Função de Melhoria.

### **1 – Identificar as funções dentro do escopo do projeto de melhoria**

A proposta de melhoria, a documentação funcional do sistema atual e os pontos de função do sistema existente são usados para identificar as funções de transação e de dados que fazem parte do escopo do projeto de melhoria. Uma análise de ponto de função do sistema existente é um pré-requisito essencial pois todas as funções existentes que são afetadas direta ou indiretamente pela melhoria contribuem para o tamanho em pontos de função da melhoria. Se, por alguma razão, uma análise de pontos de função de um sistema existente não existe, então ela deve ser feita para identificar ao menos as funções afetadas pela melhoria.

O tamanho do sistema existente, ou a parte impactada pelo projeto de melhoria, é expresso em pontos de função padrão,  $\Sigma$ PFbase.

### **2 - Determinar as funções de dados e de transações a serem adicionadas**



A proposta de melhoria deve especificar as funções de dados e de transação a serem adicionadas à aplicação. A partir dessa proposta, é possível calcular o tamanho das funções adicionadas aplicando o método de Análise de Pontos de Função (APF). O tamanho total da funcionalidade adicionada é expresso em  $\Sigma PF_{adicionados}$ . **O fator de impacto para funções adicionadas é sempre 1.** Assim, o número de pontos de função de melhoria para cada função adicionada é determinado da seguinte maneira:

$$PF_{Madicionados} = PF_{adicionados}$$

Isso significa, por exemplo, que 3 pontos de função adicionados resultarão em 3 pontos de função de melhoria.

### 3 - Determinar as funções de dados e de transações a serem excluídas

As funções (de dados ou de transações) que serão excluídas do sistema existente são identificadas na proposta de melhoria e o número de pontos de função que elas representam é determinado. O tamanho das funções excluídas é expresso como

$$\Sigma PF_{excluídos}..$$

O fator de impacto para as funções excluídas é igual a 0,4. O número de pontos de função de melhoria para uma função excluída é determinado assim:

$PF_{Mexcluídos} = PF_{excluídos} \times 0,40$  Isso significa, por exemplo, que 6 pontos de função excluídos resultarão em 2,4 pontos de função de melhoria.

### 4 - Determinar as funções de dados a serem modificadas e o fator de impacto

Uma função de dados pode ser um arquivo lógico interno (ALI) ou um arquivo de interface externa (AIE). Cada tipo de função de dados é avaliado para identificar:

- Funções de dados modificadas internamente: TDs são adicionados, excluídos ou modificados, e
- Funções de dados que mudam de tipo, mas não são alterados internamente (isto é, um ALI passa a ser um AIE ou vice-versa).

Determine quais funções de dados serão modificadas e quantos pontos de função cada função de dados apresenta depois da modificação, aplicando as regras de APF padrão. O tamanho do ponto de função das funções de dados modificadas é expresso como  $\Sigma PF_{modificados}$ .

Para funções de dados *alteradas internamente* o fator de impacto é calculado pelo percentual de TDs alterados. O percentual de alteração é definido pela razão entre os TDs alterados e o número original de TDs.

$$\text{Percentual de mudança de TDs} = \frac{\text{Número de TDs adicionados/alterados/excluídos}}{\text{Número de TDs originais}} \times 100$$

O fator de impacto ( $I_{alterado}$ ) é obtido na Tabela 2 utilizando o percentual de alteração no número de TDs.

% mudança de TDs	<= 33%	<= 67%	<= 100%	> 100%
Fator de impacto	0,25	0,50	0,75	1,00

Tabela 2: Fator de impacto das funções de dados

Se a função de dados *muda de tipo* (por exemplo, um AIE passa a ser um ALI), o valor de 0,40 é usado como fator de impacto. Entretanto, no caso de uma alteração no tipo, é necessário verificar se existe também uma alteração interna do Arquivo Lógico (alteração de TDs). Se o número de TDs altera, assim como o tipo, o fator de impacto devido à mudança no número de TDs deve ser determinado. O valor do fator de impacto devido à mudança no tipo é comparado com a da mudança no número de TDs e o maior valor é usado no cálculo do tamanho dos pontos de função da melhoria (alterado)

O número dos pontos de função para cada função de dados alterada é determinado assim:

$$PF_{\text{modificados}} = PF_{\text{modificados}} \times I_{\text{modificados}}$$

O número dos pontos de função de melhoria originado da alteração nas funções de dados depende, portanto, da extensão dessa alteração na função de dados.

Se um AIE ou um ALI é *dividido* em duas (ou mais) funções de dados, uma função de dados excluída e duas (ou mais) funções de dados são contadas.

Se um AIE e um ALI são *combinados*, duas funções de dados são contadas como excluídas e uma como adicionada.

## 5 - Determinar as funções de transação que serão modificadas e o fator de impacto

As funções de transação modificadas devem ser identificadas e o tamanho de cada transação após a mudança (melhoria) é determinado.

Uma função de transação é considerada alterada se ela é alterada de alguma forma, mas mantém o mesmo nome e a mesma intenção antes e depois da melhoria. Para determinar o tamanho funcional de uma função transacional após a alteração, as mesmas diretrizes de contagem para sistemas novos são usadas, aplicando as regras padrão de APF. O número de pontos de função para cada função transacional é expresso como  $PF_{\text{modificados}}$ .

Uma função de transação pode ser afetada pelas alterações nas funções de dados. Todas as funções de transação especificadas na proposta de melhoria e aquelas afetadas pelas alterações nas funções de dados são incluídas no escopo da análise. Isso significa que a transação é contada quando uma das seguintes condições é satisfeita:

- A transação é identificada na proposta de melhoria ou
- A transação passa por uma alteração em consequência de outras alterações definidas na proposta de melhoria.

Geralmente, a transação deve ser contada se o usuário pode identificar que a transação foi alterada. Isso significa que ao menos um dos seguintes critérios foi atendido:

- Uma transação é afetada por um TD que foi adicionado, alterado ou excluído;  
Uma transação é afetada por um Arquivo Lógico (ALI ou AIE) adicionado, alterado ou excluído;
- A interface do usuário é alterada funcionalmente (por exemplo, a disposição de uma tela ou de um relatório);
- Uma lógica de negócio que suporta uma transação é alterada (por exemplo, regras de edição ou cálculos executados nos dados da transação);
- Uma alteração cosmética visível nas interfaces de usuário é feita, por exemplo:
- Dados estáticos são modificados ou movidos em um relatório ou outra mídia,

Um cabeçalho é substituído ou modificado em um relatório ou em uma tela.

Uma alteração no nome de um TD não é considerada como alteração de transação. A natureza de um TD não altera se somente o nome é alterado.

Existem quatro passos para calcular o tamanho em pontos de função de melhoria de alterações numa transação:

- i. Identificar os TDs e os ARs utilizados pela transação;
- ii. Determinar o percentual de TDs e ARs alterados como resultado da melhoria;
- iii. Determinar o fator de impacto para a transação;
- iv. Calcular o número de pontos de função da melhoria.

#### *i - Identificar os TDs e os ARs utilizados pela transação*

O tamanho em pontos de função de melhoria de uma função transacional alterada é calculado a partir do tamanho em pontos de função da função *depois da alteração* e do fator de impacto da alteração. O fator de impacto é determinado pelo percentual de alteração no número de TDs e ARs usados pela transação.

Se uma alteração é somente cosmética, o número de TDs e ARs alterados é nulo. O impacto desse tipo de mudança é considerado mínimo e o valor do fator de impacto (0,25) reflete relativamente um impacto baixo. Entretanto, a alteração deve ser incluída no escopo do projeto de melhoria.

#### *ii – Determinar o percentual de TDs e ARs alterados como resultado da melhoria*

O fator de impacto é determinado pelo percentual de alterações no número de TDs e ARs usados pela transação, comparado com o número original dos mesmos TDs e ARs.

%Mudança de TDs =  $\frac{\text{Número de TDs adicionados/alterados/excluídos}}{\text{Número de TDs na transação original}} \times 100$

Número de TDs na transação original

%Mudança de ARs =  $\frac{\text{Número de ARs adicionados/alterados/excluídos}}{\text{Número de ARs na transação original}} \times 100$

Número de ARs na transação original

É possível que as alterações excedam 100% quando TDs e ARs são adicionados à transação.

#### *iii – Determinar o fator de impacto da transação*

O fator de impacto ( $I_{\text{alterado}}$ ) para a transação é determinado pelo percentual de alteração no número de TDs e ARs pela Tabela 3.

Mudança	Percentual de TDs		
Percentual de ARs	<= 67%	<= 100%	> 100%
<= 33%	0,25	0,50	0,75
<= 67%	0,50	0,75	1,00
<= 100%	0,75	1,00	1,25
> 100%	1,00	1,25	1,50

**Tabela 3: Fatores de Impacto para funções de transação modificadas**

Se o fator de impacto é 1,00 ou maior, você deve avaliar se melhorar a transação ainda tem sentido.

*iv – Calcular o número de pontos de função da melhoria*

O tamanho dos pontos de função de melhoria de uma única função transacional é calculado da seguinte maneira:

$$PF_{\text{modificados}} = PF_{\text{modificados}} \times I_{\text{modificados}}$$

## **6 – Calcular o tamanho do projeto de melhoria**

O tamanho do projeto de melhoria é a soma do número de pontos de função de melhoria para todas as funções de dados e transações afetadas.

$$PF_{\text{total}} = \Sigma PF_{\text{adicionados}} + \Sigma PF_{\text{excluídos}} + \Sigma PF_{\text{modificados}}$$

*Nota:* Atividades de conversão não são incluídas neste guia porque podem se manifestar de várias formas diferentes. A conversão pode ser, por exemplo, traduzir o código fonte para uma linguagem nova ou atualizada, transferir um sistema para um ambiente operacional totalmente diferente ou modificar o armazenamento de dados físicos para acomodar a introdução de um novo sistema gerenciador de banco de dados. Geralmente não são evidentes quais formas de conversão devem ser consideradas manutenção.

Quando as melhorias são feitas, às vezes é necessário desenvolver um sistema específico de conversão. Esse sistema de conversão pode ser considerado um novo desenvolvimento e seu tamanho funcional pode ser determinado usando o método APF padrão.

### **➤ CALCULAR O TAMANHO DO SISTEMA DEPOIS DA MELHORIA**

O tamanho funcional do sistema pode ser alterado como resultado da melhoria. O tamanho após a melhoria pode ser calculado pela análise de toda a aplicação novamente ou aplicando as alterações a partir da análise dos pontos de função originais. Os passos a seguir são:

1. Calcular o tamanho em pontos de função da aplicação antes das alterações ( $PF_{\text{base}}$ ) utilizando o método APF padrão.
2. Identificar as funções de dados e de transação excluídas da aplicação existente e determinar seu tamanho em pontos de função ( $PF_{\text{excluídos}}$ )

3. Determinar as funções de dados e de transação alteradas. Calcular o número de pontos de função que elas representam antes e depois da melhoria ( $PF_{depois}$  e  $PF_{antes}$ ), utilizando o método APF padrão.
4. Determinar as funções de dados e de transação adicionadas ao sistema e calcular quantos pontos de função elas representam ( $PF_{adicionados}$ )
5. Calcular o tamanho do sistema após a melhoria ( $PF_{novo}$ )

O tamanho do sistema em pontos de função após a melhoria é:

$$PF_{novo} = PF_{base} + \Sigma PF_{adicionados} + (\Sigma PF_{após-mudança} - \Sigma PF_{antes-mudança}) - \Sigma PF_{excluídos}$$

*Nota: O Fator de Impacto não interfere na determinação do tamanho do sistema após a melhoria.*

## 4. PROCESSOS DE ANÁLISE DE PONTOS DE FUNÇÃO DA PRODEMGE

### 4.1 PROCEDIMENTO DE CONTAGEM

Ao longo do nosso processo de desenvolvimento de software utilizamos procedimentos diferentes para lidar com a análise de pontos de função. Utilizamos a **Contagem Estimativa** definida pela NESMA (*Netherlands Software Metrics Users Association*) para fazermos estimativa do tamanho do software a ser desenvolvido quando na fase de concepção/iniciação. Isso se deve ao fato de que nesse ponto é possível somente estimar o tamanho do sistema, pois ainda não temos todos os elementos necessários à contagem estabelecida pelo IFPUG (*International Function Point Users Group*).

A **Contagem Estimativa** consiste em identificar todas as funcionalidades do software de acordo com as regras do IFPUG e, adota como convenção, pela falta de dados mais detalhados, considerar como de complexidade baixa as funções de dados (Arquivo Lógico Interno e Arquivo de Interface Externa) e de complexidade média as funções de transação (Entrada Externa, Saída Externa e Consulta Externa).

A **Contagem** do IFPUG é utilizada ao final da especificação, seja de módulos, partes ou do sistema todo, quando então se tem todos os elementos necessários e quando essa especificação será enviada para as fábricas para construção. Consiste em usar o método descrito no CPM (*Counting Practices Manual*) para calcular o tamanho do sistema tendo como base todos os arquivos (ALIs e AIEs), telas, relatórios (EEs, SEs e CEs), com seus campos respectivos especificados para o sistema (TDs e TRs).

No caso de alteração de escopo e alterações de requisitos durante o desenvolvimento do sistema, será utilizado o processo de **Análise de Pontos de Função para Melhoria de Software**, da NESMA, descrito seção "Metodologia da NESMA para contagem de melhoria", neste documento.

#### 4.1.1 TIPOS DE CONTAGEM

Na Prodemge é mais comum a ocorrência de dois tipos de contagem, desenvolvimento e melhoria. Existe também a contagem da aplicação. Ela é importante no sentido de conhecer os ativos da empresa e como base de conhecimento para comparações e soluções de problemas.

- **Contagem de projeto de desenvolvimento** – contagem cujo escopo consiste de todas as funcionalidades incluídas mais a funcionalidade de conversão de dados, se houver.
- **Contagem de projeto de melhoria** – o escopo consiste das funcionalidades incluídas, alteradas e excluídas, além das funcionalidades de conversão de dados.
- **Contagem da aplicação** – refere-se à contagem das funcionalidades da aplicação instalada. Dependendo do propósito pode incluir todas as funcionalidades da aplicação ou apenas aquelas que servirão de contagem de base para os projetos de melhoria.

#### **4.1.2 MARCOS DE CONTAGEM**

- **Concepção/Iniciação** – Estimativa feita com base no documento que contém a visão do produto de software. Tem por objetivo subsidiar os demais documentos desta fase: cronograma inicial, planejamento inicial e proposta comercial.
- **Elaboração** – Contagem baseada na especificação de requisitos de software. Seu objetivo principal é ser à base da negociação com a fábrica que fará a construção. Assim, pode ser do sistema todo, de cada módulo ou Ordens de Codificação (OC) a serem encaminhados às fábricas.
- **Alteração de escopo** – Pode acontecer a qualquer momento durante as fases seguintes à Concepção/Iniciação. Contagem feita nas funcionalidades incluídas, excluídas e/ou alteradas na alteração de escopo solicitada pelo cliente ou mesmo percebida e demandada pelo analista. Feita pela técnica da NESMA para melhoria de software.
- **Transição** – Contagem feita ao final da transição, após a implantação do sistema, na aplicação instalada.

#### **4.1.3 REGRAS DE REGISTRO DE TIPO DE CONTAGEM, ESCOPO E FRONTEIRA**

- **Tipo de contagem e escopo** – Os registros do tipo de fronteira e do escopo da contagem são feitos na planilha de Contagem de Pontos de Função.
- **Fronteira** – A fronteira será definida pelo Diagrama de Casos de Uso.
- Os documentos deverão ser arquivados no sistema de controle de versão, conforme orientação do processo de desenvolvimento quanto à estrutura da documentação.

#### **4.1.4 INSUMOS DE CONTAGEM**

- **Estimativa** – Documento que contém a visão do produto de software.
- **Contagem IFPUG** – Especificação dos casos de uso com suas interfaces de usuários e diagrama de classes. A especificação pode ser de todo o sistema ou de parte dele, dependendo do escopo de contagem.
- **Contagem de melhoria** – Especificação alterada, documento de alteração de escopo e Contagem da Aplicação que servirá de base para a contagem das alterações e exclusões de funcionalidades.



Não existindo a Contagem da Aplicação será necessária a contagem de base das funcionalidades afetadas antes da melhoria.

- **Contagem da aplicação** – Especificação de requisitos final, depois de todas as alterações de escopo documentadas. Pode ser a própria aplicação.

#### 4.1.5 PRODUTOS GERADOS

Planilha correspondente ao tipo de contagem contendo todas as funções da contagem e o seu valor total.

#### 4.1.6 PADRÕES DE REGISTRO DE CONTAGEM

O padrão de registro da contagem de pontos de função na Prodemge está consolidado na Planilha de Contagem de Pontos de Função e na Planilha de Estimativa de Pontos de Função, ambas disponíveis no site dedicado ao processo de desenvolvimento da Prodemge.

### 4.2 MODULARIZAÇÃO E PARTICIONAMENTO DE ENTREGAS

#### *Em casos de modularização, como lidar com os ALI e AIEs?*

Existem muitos sistemas que são especificados, aprovados e negociados por módulos. Nesse caso, como lidar com ALIs e AIEs? Uma das bases principais da Análise de Pontos de Função é a fronteira da aplicação. O fato de estar sendo especificado por módulos não muda a fronteira da aplicação. Estamos dentro da mesma aplicação. Assim, um ALI ou AIE que foi identificado e contado em um determinado módulo, não será contado novamente no módulo seguinte, onde apareceu pela segunda vez. Da mesma forma, um ALI que foi contado em um determinado módulo não será considerado um AIE no módulo seguinte, “porque estaria fora da fronteira da aplicação” (nesse caso, o módulo que está sendo contado). Eles podem ser colocados nas funções de dados da contagem de uma entrega, porém valor 0 (zero), o que seria interessante apenas para deixar explícito o uso da função de dado por aquelas transações consideradas naquela entrega.

Caso ocorram modificações em ALIs ou AIEs entre solicitações, eles não serão contados novamente, mas pode haver um ajuste na contagem nas seguintes situações:

- a) Caso o acréscimo de tipos elementares de dados (TD) ou tipos de registros elementares (TR) modifique a complexidade do ALI/AIE. Nesse caso, a diferença de complexidade, se positiva, deverá ser remunerada.
- b) Quando houver modificações em ALIs/AIEs utilizados por funcionalidades já implementadas, se essa alteração exigir a modificação dessas funcionalidades, a alteração será tratada como um projeto de melhoria e as alterações tanto nas funções de transação quanto nas funções de dados serão mensuradas e remuneradas de acordo com a regra da NESMA.
- c) Nos casos em que as alterações não provocarem nenhum impacto nas funcionalidades já implementadas (desconsiderando-se o impacto causado pelo aumento da complexidade dos ALIs/AIEs já tratado no item a), nenhum ajuste será necessário.

***Em casos de desenvolvimento com entregas por pacotes de casos de uso, como lidar com as funções de dados?***

Muitos sistemas são desenvolvidos por fábricas externas e entregues em pacotes de casos de uso (ordens de codificação). Nesses casos, como distribuir o tamanho funcional das funções de dados entre as entregas? Existem várias soluções possíveis, dentre as quais destacamos as seguintes:

1. Distribuir o tamanho total das funções de dados entre as entregas de forma proporcional ao tamanho das funções de transação de cada pacote, isto é, se o sistema vai ser entregue em "n" pacotes, o tamanho em PF do pacote "i" (PF(i)) será o somatório do tamanho em PF das funções de transação dos casos de uso desse pacote (PFtrans(i)) mais uma fração do tamanho total das funções de dados do sistema (PFdados) proporcional a divisão do tamanho das funções de transação do pacote pelo tamanho das funções de transação de todo o sistema (PFtrans):  $PF(i) = PFtrans(i) + PFdados * PFtrans(i)/PFtrans$ .
2. Distribuir o tamanho total das funções de dados uniformemente entre as entregas, ou seja, se o sistema vai ser construído em "n" pacotes, o tamanho em PF de cada pacote será o somatório das funções de transação dos casos de uso do pacote em questão mais 1/n do tamanho total das funções de dados.
3. Distribuir o tamanho das funções de dados na medida em que elas passam a ser utilizadas pelas funções de transação de cada pacote. Ou seja, a cada pacote são incorporadas as funções de dados utilizadas pelas funções de transação daquele pacote e que ainda não haviam sido contabilizadas. Esta é a alternativa utilizada na Prodemge.

## **5. MANUTENÇÃO E MELHORIA EM SOFTWARE**

### **5.1 TIPOS DE MANUTENÇÃO DE SOFTWARE**

Segundo o Manual de Práticas de Contagem do IFPUG, as manutenções se classificam em:

**Adaptativa:** Inclui modificações para atender novos requisitos de negócio, requisitos de negócio em processo de mudança, ou para adicionar funcionalidade não presente em uma versão anterior. Pode também incluir modificações necessárias ao atendimento de requisitos técnicos. É iniciada por solicitações de negócio para adicionar, modificar e/ou excluir funcionalidades de negócio. É sinônimo do conceito de uma "melhoria" como definido pelo IFPUG.

**Corretiva:** Inclui as modificações referentes ao reparo de defeitos. Não envolve mudanças às funcionalidades do negócio, mas garante que a funcionalidade previamente entregue execute como solicitado. O esforço associado com esta atividade deveria ser atribuído ao projeto de desenvolvimento ou de melhoria original que foi responsável pela introdução do defeito.

**Perfectiva:** Mudanças no hardware ou software executadas para prevenir defeitos futuros ou falhas. Por exemplo, reestruturar programas ou dados para aumentar a facilidade de manutenção e para prevenir defeitos. Podem incluir modificações para atualização de plataforma de suporte ou software de sistema, otimização de desempenho e outras atividades afins à manutenção de acordos de nível de serviço (SLA). Não existem modificações na funcionalidade de negócio associada com este trabalho.

Mais tarde, o mesmo manual afirma que:

*"A APF não deve ser utilizada para medir trabalho de manutenção perfectiva ou corretiva. (...) Se uma versão [de software mantido] contém um misto de requisitos de manutenção adaptativa, corretiva e/ou perfectiva, deve ser tomado cuidado na separação do esforço de trabalho, uma vez que as últimas duas categorias não contribuem com Pontos de Função para o negócio."*

Ou seja, para o IFPUG, somente manutenções adaptativas são passíveis de mensuração utilizando a técnica de APF. A *Netherlands Software Metrics Association* (NESMA) definiu uma metodologia específica para tratamento desse tipo de manutenção, baseada na técnica do IFPUG e publicada no documento *FPA for Software Enhancement*, disponível na Internet no site da NESMA (<http://www.nesma.nl/sectie/home/>). Essa metodologia está descrita na seção "Metodologia da NESMA para contagem de melhoria" deste documento.

Na Prodemge, as manutenções adaptativas (IFPUG) ou melhorias em software (NESMA) devem ser mensuradas de acordo com as regras da NESMA e as definições estabelecidas neste guia sobre itens não-mensuráveis.

Manutenções corretivas e perfectivas devem ser tratadas conforme definido na seção de itens não-mensuráveis.

## 5.2 CORREÇÕES E ALTERAÇÕES DE REQUISITOS DURANTE O CICLO DE DESENVOLVIMENTO

### 5.2.1 CONTEXTO

No contexto de subcontratação da fase de Construção de aplicações pela Prodemge, o principal objetivo da medição do tamanho funcional de um sistema é o dimensionamento do esforço necessário à sua implementação. Esse dimensionamento é o instrumento de derivação do esforço a ser realizado pela fábrica e determina a remuneração que ela deve receber por seus serviços.

Normalmente a construção de um sistema é dividida em etapas constituídas por ordens de codificação, contendo um grupo de funcionalidades (casos de uso) a serem implementadas. Cada Ordem de Codificação corresponde uma determinada quantidade de pontos de função. A especificação de requisitos de software é a referência para a contagem de pontos de função do módulo ou conjunto de casos de uso. Portanto, se a especificação de requisitos dessas funcionalidades for alterada de alguma forma, a contagem de pontos de função também pode sofrer alterações. Em princípio, para que o tamanho em pontos de função de um conjunto de funcionalidades pudesse ser calculado de forma correta e definitiva, ele teria que ser feito após a estabilização completa do documento de especificação de requisitos. Porém, isso nem sempre ocorre na prática. Se o sistema está sendo desenvolvido pela própria empresa, os impactos das alterações podem ser tratados internamente. Entretanto, nos casos em que a construção é responsabilidade de um fornecedor externo, por exemplo, devem existir regras específicas para tratar das questões de dimensionamento de esforço, pois isto afeta diretamente a remuneração dos fornecedores.

Pelo IFPUG, sempre que uma função de dados ou um processo elementar são alterados, eles devem ser contados novamente. Isso é válido, evidentemente, para projetos de manutenção adaptativa ou melhoria. Porém, normalmente considera-se que o sistema entrou na fase de manutenção somente depois que ele é disponibilizado em ambiente de produção. Assim, a adesão estrita à regra de contagem de melhoria durante a etapa de desenvolvimento de um sistema é inadequada, visto que toda especificação de requisitos pode conter pequenos erros, omissões ou outros tipos de problemas. Mesmo que alguns casos possam ser tecnicamente considerados alterações nos requisitos, o esforço de correção não deve ser considerado equivalente, para fins de pagamento, ao esforço necessário à implementação das funcionalidades afetadas.

É importante observar que as regras do IFPUG, por si só, são insuficientes para avaliações de esforço no caso de atividades de manutenção adaptativa. O "Manual de Práticas de Contagem de Pontos de Função" trata desse assunto na seção "Projetos de Melhoria e Atividades de Manutenção". No início desse capítulo, lê-se:

*Este capítulo fornece diretrizes adicionais na identificação e contagem das mudanças funcionais em aplicações instaladas. Não é abordado o relacionamento entre o tamanho funcional da melhoria e o esforço necessário para implementar a melhoria. Para obtenção de perspectivas adicionais sobre este tópico, o leitor pode recorrer à publicação da NESMA "Function Point Analysis for Software Enhancement" [NESMA 2001], que pode ser obtida através do site [www.nesma.org](http://www.nesma.org).*

Ou seja, as regras do IFPUG tratam apenas do cálculo do tamanho funcional de uma aplicação ou de um projeto de melhoria num contexto de alterações em aplicações instaladas. A afirmação transcrita acima deixa claro que as medidas de tamanho em ponto de função calculadas segundo a metodologia do IFPUG, no caso de atividades de manutenção adaptativa durante o ciclo de desenvolvimento, não necessariamente guardam correlação com o esforço necessário para realizar essas atividades.

Mais tarde, o mesmo manual afirma que:

*A APF não deve ser utilizada para medir trabalho de manutenção perfectiva ou corretiva. (...) Se uma versão [de software mantido] contém um misto de requisitos de manutenção adaptativa, corretiva e/ou perfectiva, deve ser tomado cuidado na separação do esforço de trabalho, uma vez que as últimas duas categorias não contribuem com Pontos de Função para o negócio.*

Por conseguinte, para tratar dessa questão de forma mais adequada, seguindo as diretrizes do IFPUG, a Prodemge adota as regras definidas pela NESMA (*Netherlands Software Metrics Association*) em seu artigo "Análise de Pontos de Função para Projetos de Melhoria" (*FPA for Software Enhancement*) para o dimensionamento de atividades melhoria durante o ciclo de desenvolvimento, assim como para manutenção adaptativa, adicionadas das definições estabelecidas neste guia sobre itens não-mensuráveis.

## 5.2.2 DEFINIÇÕES

- Entende-se por **inclusão de novas funcionalidades** as alterações de escopo que modifiquem a abrangência do projeto, como a inclusão de novos casos de uso após a definição e aprovação do escopo inicial pela fábrica de software. Essas alterações de escopo podem gerar impactos no cronograma e na política de alocação de recursos da contratada.
- São consideradas **alterações de requisitos** as alterações decorrentes de modificações ou aprimoramento nos requisitos do usuário e também as modificações resultantes de omissões na especificação fornecida pela Prodemge, após a entrega da mesma para implementação. Esse tipo de alteração é considerada uma manutenção adaptativa e deve ser mensurada de acordo com a técnica da NESMA para mensuração de projetos de melhoria.
- Entende-se por **manutenção corretiva** as modificações referentes ao reparo de defeitos. Este tipo de manutenção não envolve mudanças nas funcionalidades do negócio, mas garante que a funcionalidade previamente entregue execute como solicitado. O esforço associado com essa atividade é do responsável pela introdução do defeito. No caso de subcontratação, esses defeitos devem ser corrigidos pelo fornecedor, no período de garantia, em conformidade com as cláusulas de qualidade estabelecidas em contrato e sem ônus para a Prodemge.

### 5.2.3 CRITÉRIO PARA TRATAMENTO DE INCLUSÃO DE NOVAS FUNCIONALIDADES

As **inclusões de novas funcionalidades** são consideradas melhorias e devem ser feitas através de solicitações específicas e mensuradas de acordo com a técnica da NESMA para projetos de melhoria. Observe-se que a regra da NESMA, nesses casos, é equivalente à técnica padrão de pontos de função (IFPUG). Caso haja impactos significativos no cronograma ou impossibilidade da fábrica de atender dentro das condições pré-estabelecidas, um novo acordo será negociado.

### 5.2.4 CRITÉRIOS PARA TRATAMENTO DE ALTERAÇÕES DE REQUISITOS DURANTE O DESENVOLVIMENTO

#### 5.2.4.1 REGRAS GERAIS

1. As **alterações de requisitos** são mensuradas de acordo com a regra da NESMA e são remuneradas de acordo com o definido a seguir.
2. A fábrica externa responsável pela construção deve buscar o completo entendimento das especificações de caso de uso fornecidas pela Prodemge dentro do período de construção de cada entrega. Ou seja, no período que começa com o recebimento da especificação até o momento da entrega do produto, a fábrica deve buscar a elucidação de qualquer dúvida ou questão cujo não esclarecimento possa prejudicar a implementação correta do sistema. Assim, caso a fábrica detecte informações contraditórias ou omissões na especificação, essas inconsistências deverão ser registradas como "Dúvida", "Pendência" ou "Erro" na ferramenta de gestão de defeitos. Esses tipos de problemas, se resolvidos pela Prodemge durante o período de implementação da entrega, não são considerados para efeito de remuneração.

São incluídos nessa categoria erros como os seguintes:

- Artefatos, modelos, instruções ou especificações faltantes, quando essa omissão pode ser facilmente detectada a partir da análise das informações disponíveis;
  - Referência a classes, interfaces, regras, tabelas ou atributos inexistentes nos modelos entregues;
  - Incompatibilidade de tipo, tamanho ou obrigatoriedade entre as interfaces, a classe e o esquema do banco;
  - Inconsistências nas regras de validação de atributos (especialmente validação cruzada do tipo "o atributo "x" é obrigatório se..." ) nos casos em que a análise da especificação (regras de negócio) permita determinar a existência de um erro ou incompatibilidade entre os requisitos fornecidos;
  - Incompatibilidade entre as regras de negócio e aspectos das interfaces ou do modelo de dados;
  - Como regra geral, a fábrica deve reportar os problemas encontrados o mais cedo possível, e não iniciar a implementação antes de buscar a elucidação e correção de quaisquer erros que aparecem como contradições ou omissões na especificação e que impeçam a implementação correta da funcionalidade na ausência da informação faltante.
3. Caso a alteração de requisitos resulte em retrabalho significativo para a fábrica, os prazos de entrega podem ser renegociados em proporção equivalente ao esforço calculado para implementar alterações, desde que respeitadas as datas limites do cronograma. Se isso não for possível, será negociada outra solução. As datas de inclusão das questões na ferramenta de gestão de defeitos e as de suas respectivas resoluções devem ser avaliadas para determinar a responsabilidade das partes.
  4. Deve ser feita a verificação da especificação de requisitos de software pelo processo de qualidade interno antes de seu repasse o fornecedor.

#### 5.2.4.2 REGRAS PARA PAGAMENTO

1. As alterações em código fonte solicitadas às fábricas serão consolidadas por Ordem de Codificação para efeito de cálculo do tamanho dessas alterações. Esse cálculo será feito após a entrega do código, depois que a entrega for homologada e aceita pela Prodemge. Nesse momento, o tamanho funcional de todas as alterações pedidas para a Ordem de Codificação será determinado utilizando a técnica da NESMA e a remuneração devida será estabelecida.
2. As alterações de requisitos decorrentes de modificações ou aprimoramento nos requisitos pelo usuário após o início da implementação serão pagas pela Prodemge, de acordo com a regra da NESMA, com periodicidade a ser definida de acordo com o projeto.
3. Caso o fornecedor já tenha desenvolvido a funcionalidade no momento da solicitação de alteração, a Ordem de Codificação deverá ser finalizada parcialmente com a entrega dos artefatos correspondentes e será feita uma nova Ordem de Codificação incluindo as alterações solicitadas.
4. Caso o fornecedor ainda não tenha desenvolvido a funcionalidade no momento da solicitação de alteração, esta não será remunerada, porém poderão ser feitos ajustes na contagem de pontos de função.

#### 5.2.5 PROCEDIMENTOS RELACIONADOS AO REGISTRO E ACOMPANHAMENTO DE DEFEITOS

A ferramenta para registro e acompanhamento de resolução de defeitos e problemas é o Rational Team Concert – RTC. Cada demanda deverá ter suas dúvidas, pendências e defeitos tratados nesta ferramenta.

### 6. ITENS NÃO-MENSURÁVEIS

Em cenários de subcontratação, o principal objetivo da medição do tamanho funcional de um sistema é o dimensionamento do esforço necessário para sua implementação, de onde se deriva a remuneração da fábrica. Entretanto, em muitos casos onde há esforço, este não é devidamente mensurado pela técnica de APF. Exemplos são a implementação de tabelas do tipo "code data", alterações cosméticas na interface de usuário, saídas diversas para um mesmo processo elementar (múltiplos meios) etc.

Durante o desenvolvimento de um módulo de software, a remuneração do esforço das fábricas necessário à implementação de itens não mensuráveis é conseguida através do ajuste do preço estabelecido para o ponto de função, que normalmente inclui uma margem de segurança – baseada em valores médios de sistemas já implementados - destinada a remunerar o esforço realizado incluindo esses itens. No caso dos contratos baseados em ponto de função, os fornecedores fazem esse ajuste quando estabelecem o preço de seus serviços.

Entretanto, para possibilitar a solicitação de manutenção adaptativa durante o ciclo de desenvolvimento seguida da devida remuneração do esforço realizado, a Prodemge adota a tabela de itens não mensuráveis descrita a seguir, a ser aplicada nos casos em que as alterações não estejam sendo (ou não possam ser) mensuradas pela técnica de APF ou NESMA.



## 6.1 TABELA DE ITENS NÃO-MENSURÁVEIS

	Especificação do item	Unidade medida	Qtde. PF
1	<p><b>Layout</b> - contempla as alterações referentes aos layouts de telas, mudança de posição de campos em telas, relatórios ou layout de arquivos, sem que haja alteração em elementos de dados, arquivos referenciados ou informações de controle. Contempla também inclusão, alteração ou exclusão de logotipo da empresa, divisão de telas e/ou relatórios, sem que tenha havido mudança na funcionalidade.</p> <p><b>Nota: nos casos em que a utilização dos componentes do framework jCompany permita que a alteração solicitada seja feita através da edição de um único artefato da aplicação, somente um elemento será contado.</b></p>	Quantidade de elementos	0,04
2	<p><b>Campos e Variáveis</b> - contempla a inclusão, alteração ou exclusão de campos e variáveis em programas e tabelas, sem que tenha havido mudança na funcionalidade. Contempla também padronização de nomenclatura de campos e variáveis.</p>	Quantidade de dados/campos	0,08
3	<p><b>Mensagens</b> - contempla a necessidade de alterações de mensagens de retorno ao usuário, desde que não façam parte de um ALI ou AIE.</p> <p><b>Nota: nos casos em que a utilização dos componentes do framework jCompany permita que a alteração solicitada seja feita em através da edição de um único artefato da aplicação, somente um elemento será contado.</b></p>	Quantidade de mensagens	0,04
4	<p><b>Menus</b> - contempla a necessidade de adição ou reestruturação de menus de navegação estáticos; ajuda (help estático); criação, alteração ou exclusão de páginas estáticas.</p>	Quantidade de itens alterados	
	► Alteração e/ou criação de item de menu		0,04
	► Exclusão de item de menu		0,01
5	<p><b>Dados Hard Coded</b> - contempla a necessidade de inclusão, alteração ou exclusão de dados pertencentes a listas (combo box) ou tabelas físicas.</p>	Quantidade de dados/campos	0,04
6	<p><b>Parâmetros de processamento</b> - contempla a necessidade de alteração dos valores dos parâmetros, sem que a lógica de processamento tenha sido alterada.</p> <p>(Exemplo: ajustar filtro para recuperar dados entre 0 e 50 ao invés de valores entre 10 e 50).</p>	Quantidade de parâmetros	0,01
7	<p><b>Code Table</b> - contempla a necessidade de criação, alteração e exclusão de tabelas do tipo <i>code table</i> e respectivas funcionalidades, desde que as mesmas já estejam em produção.</p>		
	► Inclusão de tabela e funcionalidades (Create/Read/Update/Delete - CRUD)		2,7

	Especificação do item	Unidade medida	Qtde. PF
	▶ Alteração da tabela e funcionalidades (CRUD)		1,3
	▶ Excluir tabela e funcionalidades (CRUD)		0,6
	▶ Inclusão de funcionalidade		0,3
	▶ Alteração de funcionalidade		0,3
	▶ Exclusão de funcionalidade		0,3

**Tabela 4: Itens não-mensuráveis**

## 6.2 MANUTENÇÃO COSMÉTICA

São consideradas manutenções cosméticas as demandas associadas a 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.

Para manutenção cosmética adotar-se-á a tabela de itens não-mensuráveis.

## 6.3 MANUTENÇÃO CORRETIVA

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

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

Quando o sistema estiver fora da garantia, deverá ser estimado e calculado o tamanho do projeto de manutenção corretiva considerando um fator de impacto de 50% sobre a quantidade apurada de pontos de função das funcionalidades corrigidas.

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

## 6.4 MANUTENÇÃO EM FUNCIONALIDADES SEM ALTERAÇÃO DE REQUISITOS FUNCIONAIS

São consideradas nesta categoria as demandas de manutenção associadas a solicitações que envolvem aspectos não funcionais, ou seja, sem alteração em requisitos funcionais. Seguem alguns exemplos:

- Aumentar a quantidade de linhas por página em um relatório;
- Colocar paginação em um relatório;
- 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;
- Replicação de base de dados ou criação de base temporária para resolver problemas de desempenho ou segurança;
- 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.

Nestes casos, a aferição do tamanho em pontos de função do projeto de manutenção deverá considerar um fator de impacto de 50% sobre a quantidade apurada de pontos de função das funcionalidades afetadas.

Observa-se que as manutenções perfectivas se enquadram neste contexto.

## 7. CONTAGEM DE PONTOS DE FUNÇÃO COM MÚLTIPLAS MÍDIAS

A contagem de pontos de função 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*. O IFPUG reconhece ambas as abordagens para a aplicação das regras definidas no CPM.

Seguindo o disposto no Roteiro de Métricas de Software do SISP (Governo Federal), as estimativas e contagens 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 e extraídos do texto do SISP:

- **Canal:** também se refere à mídia. Múltiplos canais são sinônimos 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.
- **Multimídia:** 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 do 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 facilita o entendimento de outros cenários envolvendo múltiplas mídias. Este roteiro pode ser atualizado considerando a publicação de novas diretrizes do IFPUG e novos cenários que emergirão nas contagens dos projetos a serem desenvolvidos.

### 7.1 CENÁRIO 1: MESMOS DADOS APRESENTADOS EM TELA E IMPRESSOS

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

Adota-se a abordagem *Single instance*, considerando que dados idênticos sendo apresentados em tela e relatório impresso devem ser contados como uma única função.

Caso as lógicas de processamento da consulta em tela e do relatório em papel sejam distintas, o processo elementar não é único e, portanto, a funcionalidade será contada duas vezes (*Multiple instance*). Neste caso, duas funções são contadas: apresentação de dados em tela e apresentação de dados impressos.

### 7.2 CENÁRIO 2: MESMOS DADOS DE SAÍDA COMO DADOS EM ARQUIVO E RELATÓRIO IMPRESSO

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

Deve-se utilizar a abordagem *Single instance* considerando que os dados impressos e os dados apresentados no arquivo de saída sejam idênticos e que a ferramenta de desenvolvimento apoie a geração dessas múltiplas saídas. Assim, apenas uma funcionalidade será incluída na contagem de pontos de função.

Caso as lógicas de processamento da geração do arquivo de saída e do relatório em papel sejam distintas, o processo elementar não é único e, portanto, a funcionalidade será contada duas vezes. 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*.

### 7.3 CENÁRIO 3: MESMOS DADOS DE ENTRADA BATCH E ON-LINE

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

Adota-se a abordagem *Multiple instance*, que conta duas funcionalidades: a entrada de dados batch e a entrada de dados on-line. Geralmente, a lógica de processamento utilizada nas validações em modo batch é diferente da lógica de processamento das validações nas entradas de dados on-line.

## 7.4 CENÁRIO 4: MÚLTIPLOS CANAIS DE ENTREGA DA MESMA FUNCIONALIDADE

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

Deve-se utilizar 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.

Caso o projeto seja claro o suficiente para dizer que o desenvolvimento é o mesmo, deverá ser utilizada a abordagem *Single instance*.

## 7.5 CENÁRIO 5: RELATÓRIO EM MÚLTIPLOS FORMATOS

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

Nestes casos, considera-se a ferramenta utilizada na geração dos relatórios.

Se a equipe de desenvolvimento precisar desenvolver o relatório nos dois formatos na ferramenta em questão, serão contadas duas funcionalidades, porque a lógica de processamento de análise de condições para verificar quais são aplicáveis é identificada.

No entanto, se a ferramenta de desenvolvimento suportar um gerador de relatórios que o usuário visualize o relatório em tela e o gerador permita ao usuário imprimir o relatório, salvar em html ou salvar no formato de valores separados por vírgula, então se contará apenas uma vez, observando que a funcionalidade será da ferramenta e não da aplicação.

## 8. ANÁLISE DE CENÁRIOS (CASOS DE CONTAGEM)

### 8.1 CENÁRIO 1 – COMUNICAÇÃO ENTRE SISTEMAS UTILIZANDO TECNOLOGIAS ESPECÍFICAS

#### 8.1.1 CONCEITO

Uma aplicação precisa usar dados mantidos por outra.

#### 8.1.2 ABORDAGEM

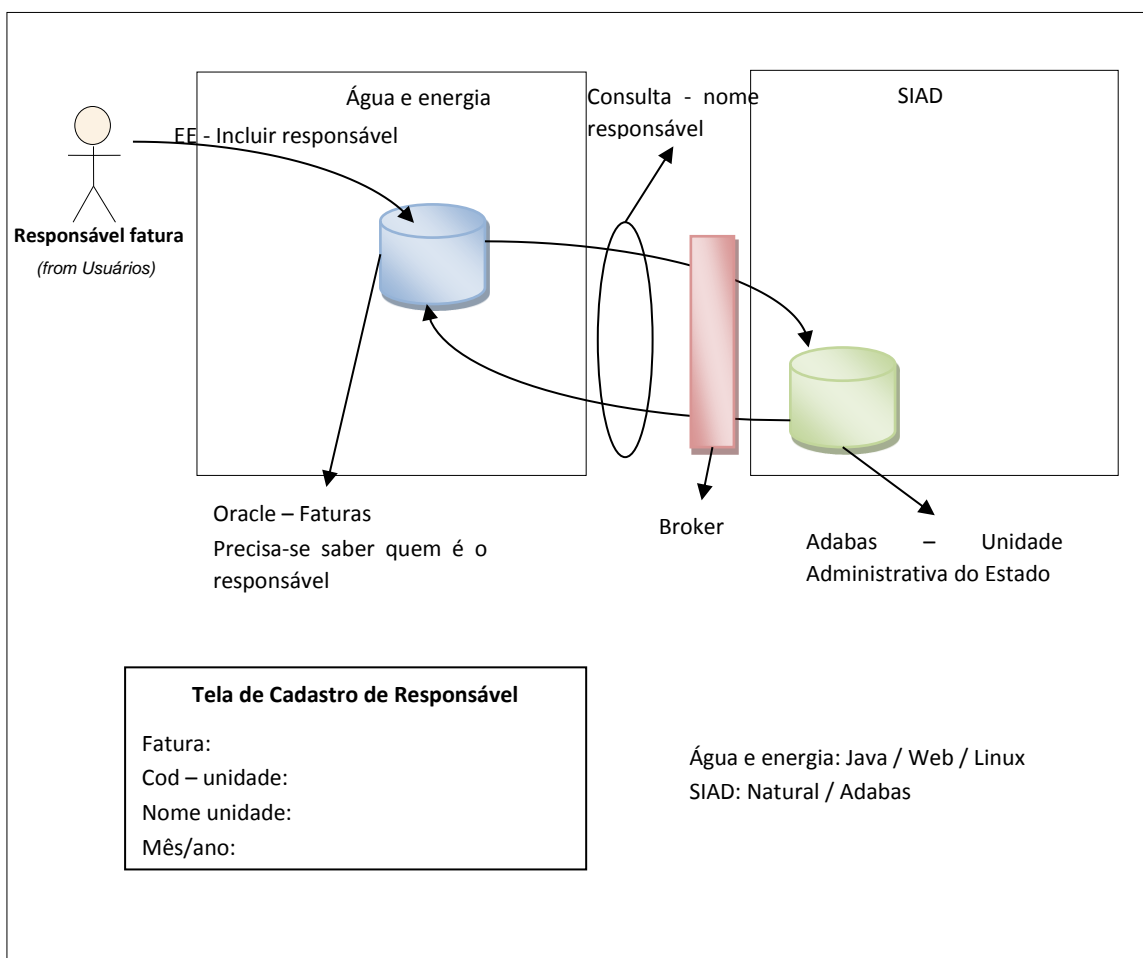
Se um sistema (A) tem como requisito acessar dados mantidos em outro sistema (B) há um AIE presente na contagem dos PFs de A independente da forma como o acesso é feito (seja via tabela do banco de dados, seja via web-service, seja via broker ou qualquer outro middleware)

#### 8.1.3 EXEMPLOS

- Exemplo 1

Situação

Quando há envio ou recebimento de dados de/para outro sistema. Ambos estão na mesma rede e em plataformas diferentes.



**Figura 10: Exemplo de comunicação entre sistemas utilizando tecnologia específica**

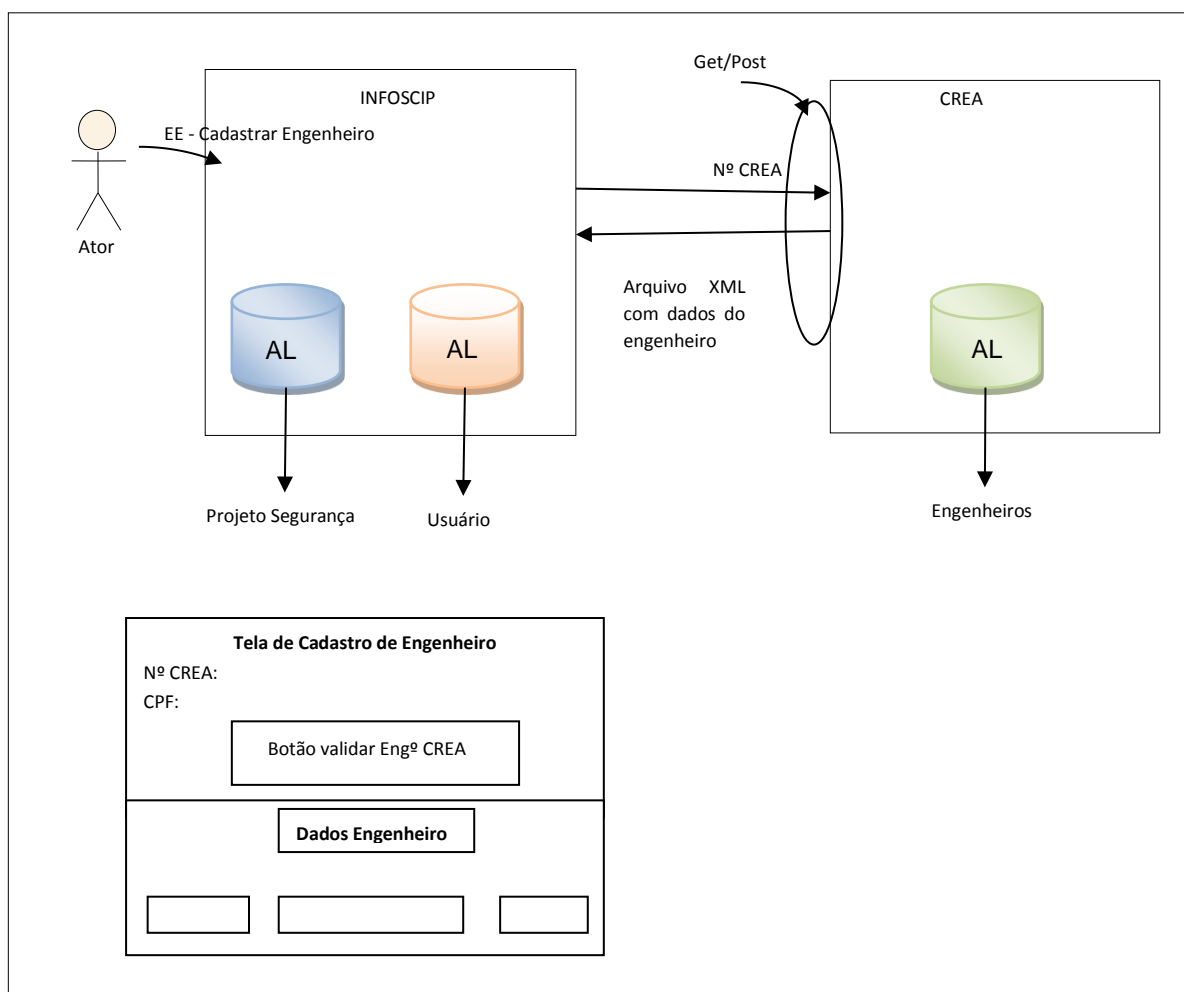
#### Análise

- Água e Energia tem o requisito de acessar dados de Unidade Administrativa mantida pelo SIAD;
- Unidade Administrativa é uma AIE para Água e Energia;
- O Broker é apenas tecnologia para acessar os dados do SIAD, não afeta a contagem;
- Todas as transações de Água e Energia que precisam de dados de Unidade Administrativa vão considerar este AIE como um AR.

#### • Exemplo 2

#### Situação

Quando há envio ou recebimento de dados de/para outro sistema. Os sistemas não estão na mesma rede e na mesma plataforma.



**Figura 11: Exemplo de comunicação entre sistemas utilizando tecnologia específica**

### Análise

- Para o INFOSCIP: Entrada Externa - Cadastrar Engenheiro
- O Cadastro do Engenheiro do CREA é um AIE
- Somente são contados como TD do AIE os campos utilizados (só para reforçar)
- Consultar dados do Engenheiro do CREA **não é um processo elementar**. Isso faz parte do processo de entrada (válido também para o exemplo 1).
- O cadastro de Engenheiros é um AR na transação de Cadastrar Engenheiro.
- Se na hora em que os dados forem apresentados o engenheiro desistir de se cadastrar, ainda assim não é um processo elementar. Se isso tiver sentido por si só então o sistema terá essa consulta separadamente.

Mesmo que houvesse essa consulta independente, a transação de cadastro de engenheiro continuaria sendo contada do mesmo jeito, com os mesmos TDs e ARs.



## 8.2 CENÁRIO 2 – CONSULTA / VALIDAÇÃO E EXIBIÇÃO DE DADOS DURANTE UMA TRANSAÇÃO

### 8.2.1 CONCEITO

Um processo elementar é composto de vários passos (ou regras de negócio) e que cada uma delas isoladamente não pode ser contada como uma transação.

### 8.2.2 ABORDAGEM

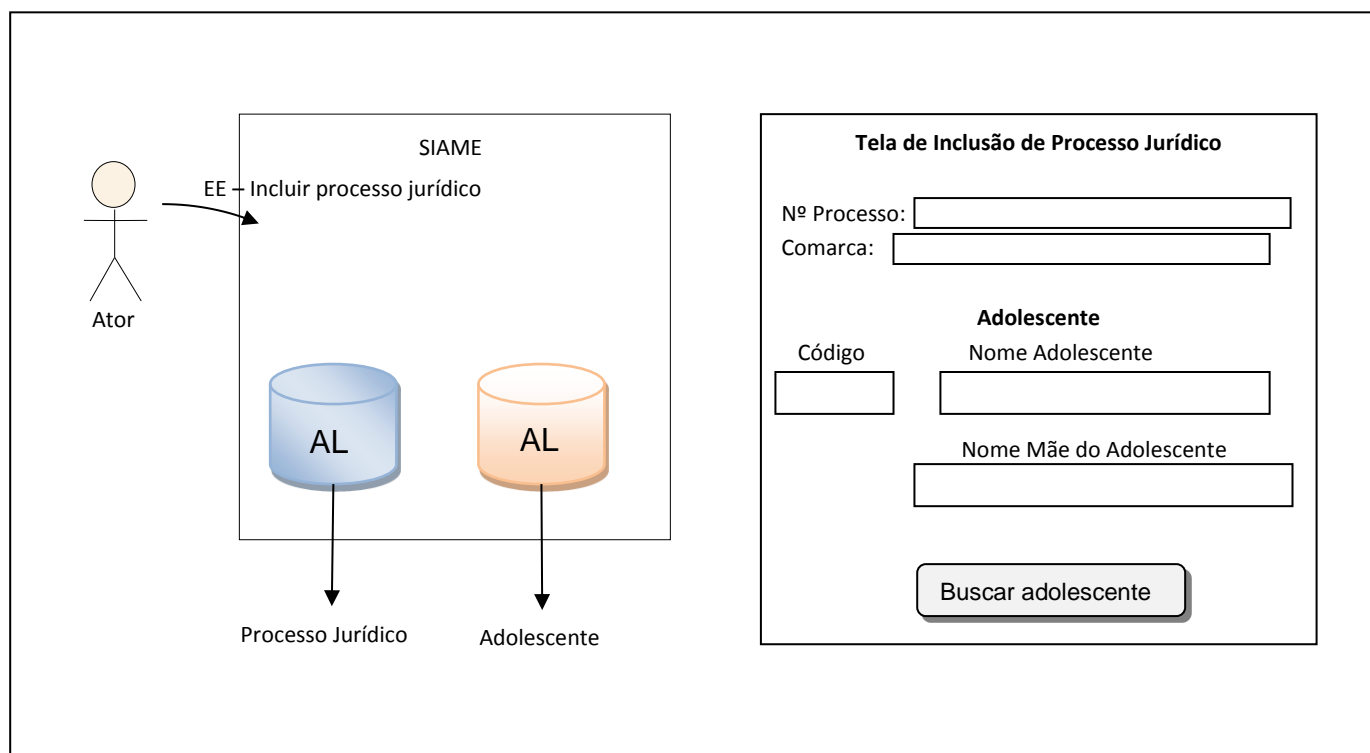
Se durante o processamento de uma transação dados são recuperados de algum arquivo lógico (por exemplo, consultados no banco de dados) e apresentados ao usuário não implica que este passo sozinho seja um processo elementar.

### 8.2.3 EXEMPLO

#### Situação

Durante uma inclusão de dados, há uma consulta para complementar os dados da inclusão ou dar clareza ao processo.

Exemplo:



**Figura 12: Exemplo de consulta / validação e exibição de dados durante uma transação**

#### Análise (Abordagem)

Consultar e exibir o nome do adolescente e da mãe do adolescente não é um processo elementar. Faz parte do processo Incluir Processo Jurídico.

### 8.3 CENÁRIO 3 – VISÃO DO USUÁRIO

#### 8.3.1 CONCEITO

A definição do IFPUG para ponto de função é a seguinte:

*"Pontos de função medem o tamanho do software através da quantificação da **funcionalidade provida ao usuário baseando-se** somente em desenhos lógicos e **especificações funcionais**."*

*Fonte: <http://www.ifpug.org/about/faqs.htm>.*

Desta definição, podemos retirar a conclusão de que a técnica de APF IFPUG quantifica funcionalidades **disponibilizadas para o usuário**, não sendo estritamente necessário que elas tenham sido definidas por ele. Obviamente, as funcionalidades especificadas devem atender às necessidades do usuário e devem, sempre que possível, ser aprovadas por ele. Esta definição também esclarece que **a medição é baseada nas especificações funcionais**. Para a Prodemge, isto significa que a visão do usuário está modelada (ou traduzida) nas especificações de requisitos.

#### 8.3.2 ABORDAGEM

Como a Prodemge é a responsável por traduzir em requisitos as necessidades de vários clientes (ou usuários) com necessidades de negócio distintas, é do melhor interesse do Estado, especialmente nos casos de subcontratação total ou parcial do desenvolvimento, que o tamanho funcional represente da melhor forma possível a correlação com o esforço de implementação a ser desenvolvido pela Prodemge ou por uma eventual subcontratada. Portanto, nos casos em que a definição da visão do usuário trazer impactos nos custos do software, sem que haja justificativa em termos de esforço para a adoção de uma definição que implique no aumento do tamanho funcional, prevalecerá o entendimento de que a visão do usuário é aquela que pode ser determinada pelas funcionalidades a serem desenvolvidas e entregues, de acordo com a especificação de requisitos.

### 8.4 CENÁRIO 4 – CONTAGEM DE MELHORIA UTILIZANDO A TÉCNICA DA NESMA

#### 8.4.1 CONCEITO

A contagem de melhoria é baseada num conjunto de as regras definidas pela NESMA (*Netherlands Software Metrics Association*) em seu artigo "Análise de Pontos de Função para Projetos de Melhoria" (*FPA for Software Enhancement*) para o dimensionamento de em projetos de melhoria (ou manutenção adaptativa, de acordo com o IFPUG).

Seis passos são necessários para determinar o escopo e o tamanho funcional para projetos de melhoria:

1. Identificar as funções de dados e de transações dentro do escopo do projeto de melhoria.
2. Determinar o tamanho de melhoria das funções de dados e de transação a serem adicionadas.
3. Determinar o tamanho de melhoria das funções de dados e de transação a serem excluídas.
4. Determinar o tamanho de melhoria das funções de dados a serem alteradas.
5. Determinar o tamanho de melhoria das funções de transação a serem alteradas.
6. Calcular o tamanho funcional do projeto de melhoria.

#### 8.4.2 ABORDAGEM

A Prodemge adota as regras da NESMA para o dimensionamento de atividades de manutenção durante o ciclo de desenvolvimento. São consideradas neste contexto as alterações de requisitos as alterações

decorrentes de modificações ou aprimoramento nos requisitos do usuário e também as modificações resultantes de omissões na especificação fornecida pela Prodemge, após a entrega da mesma para implementação. As alterações mensuradas são remuneradas em duas situações: (1) quando a alteração acontecer após a entrega do código pela fábrica ou (2) quando a alteração, analisada segundo o critério da NESMA, revelar-se não trivial, isto é, gerar impacto superior a 0,25 nas funções de dados ou de transação.

As alterações de requisitos decorrentes de erros, omissões ou contradições na especificação cujo fator de impacto (NESMA) seja maior que 0,25 nas funções de dados ou nas funções de transação também serão pagas.

As alterações no código fonte porventura necessárias em função de problemas reportados pela Prodemge e que decorrem de erros/omissões detectadas na especificação após a entrega para homologação são consolidadas por Ordem de Codificação para efeito de cálculo de esforço. Esse cálculo será feito após a entrega do código, depois que a entrega for homologada e aceita pela Prodemge. Nesse momento, o tamanho funcional de todas as alterações pedidas para a Ordem de Codificação será determinado utilizando a técnica da NESMA e a remuneração devida será estabelecida.

### 8.4.3 EXEMPLOS

- **Exemplo 1 – Modificação de ALI**

Situação – Um ALI de 1 TR e 37 TDs é expandido pela adição de 14 novos TDs.

Questão – Qual fator de impacto deve ser usado e quantos pontos de função de melhoria essa alteração gera?

Resposta – O tamanho do ALI depois da alteração é de 10 PF (ALI com média complexidade). O impacto da alteração expresso como um percentual de TDs é  $14/37 \times 100 = 37,8\%$ . Pela tabela 1, uma alteração de 37,8% traz um fator de impacto de 0,5. O tamanho do ponto de função de melhoria é  $10 \times 0,5 = 5$  EPF

- **Exemplo 2 – Transformação de AIE em ALI**

Situação – O Sistema de informação A usa um AIE mantido pelo Sistema de informação B. Houve uma decisão de que, no futuro, a manutenção dessa função será feita pelo Sistema de Informação A (a estrutura da função de dados não foi alterada).

Questão 1 – Qual fator de impacto deve ser usado?

Resposta – Um AIE é importado para o sistema A e convertido para ALI. O fator de impacto 0,4 é usado quando um AIE é alterado para um ALI ou vice-versa.

Questão 2 – Quantos pontos de função de melhoria resultam da alteração dessa função de dados?

Resposta – Assumindo que a função de dados é um ALI de baixa complexidade, seu tamanho após a alteração é 7 PF. O fator de impacto é 0,4. Então, a alteração resulta em  $7 \times 0,4 = 2,8$  EPF.

- **Exemplo 3 – Transformação de AIE em ALI, com alterações nos campos**

Situação – O Sistema de informação A usa um AIE que é mantido pelo Sistema B e contém 45 TDs. Houve uma decisão de que o Sistema A irá manter a função de dados e, como consequência, 25 TDs serão removidos dessa função de dados.

Questão 1 – Qual fator de impacto deve ser utilizado?

Resposta – O fator de impacto 0,4 é utilizado quando um AIE é alterado para um ALI ou vice-versa. Entretanto, nesse caso, houve uma alteração estrutural na função de dados que deve também ser considerada na contagem. O percentual de alteração é:

$$25/45 \times 100\% = 55,6\%$$

Pela tabela 1, uma alteração de 55,6% tem um fator de impacto de 0,5. Isso é maior que o fator de impacto pela mudança de tipo (de AIE para ALI). Assim, o maior valor, 0,5, deverá ser usado.

Questão 2 – Quantos pontos de função de melhoria resultam da alteração da função de dados?

Resposta – Assumindo que o arquivo é um ALI de baixa complexidade, o tamanho da função de dados após a alteração é de 7 PF. O fator de impacto é 0,5 e assim a medida funcional da alteração é  $7 \times 0,5 = 3,5$  EPF.

- **Exemplo 4 – Divisão de AIE**

Situação – Um AIE é dividido em 2.

Questão – Como contar essa alteração?

Resposta – Conte um AIE excluído e dois AIEs adicionados. Uma análise mais profunda deve ser feita para contar todas as funções de transação afetadas pela alteração na função de dados.

- **Exemplo 5 – Adição de AIE**

Situação – Um novo AIE foi adicionado ao sistema.

Questão – Como contar essa alteração?

Resposta – Conte um AIE adicionado (fator de impacto = 1) e leve em consideração as funções de transação que utilizam essa função de dados. Essas transações devem ser reavaliadas como parte do projeto de melhoria.

- **Exemplo 6 – Alteração de TDs**

Situação – 3 TDs numa função de dados usados por uma transação são alteradas. 2 dos TDs alterados serão utilizados pela transação.

Questão – Como contar o impacto da alteração na transação?

Resposta – Uma função de dados alterada e dois TDs alterados são contados para determinar o fator de impacto para a transação alterada.

- **Exemplo 7 – Alteração de TDs**

Situação – 3 TDs em uma função de dados (ALI ou AIE) usado por uma transação são alterados. Um dos TDs será usado pela transação.

Questão – Precisa-se de cálculo para essa alteração?

Resposta – A transação tem de ser incluída no escopo da melhoria.

A função de dados é um AR para a transação e resultará em uma alteração lógica no processo subjacente à função de transação.

- **Exemplo 8 – Alteração do nome de TDs**

Situação – O nome de um TD é alterado em uma função de dados usado por uma transação. O TD é usado na transação.

Questão – Precisa-se de cálculo para essa alteração?

Resposta – A transação e a função de dados não são contadas.

- **Exemplo 9 – Adição, exclusão e alteração de TDs**

Situação – Dois TDs são adicionados a uma função de dados, 1 TD é excluído e 3 TDs alterados.

Questão – Quantos TDs serão contados nessa alteração?

Resposta – O número de TDs contados são  $2 + 1 + 3 = 6$  TDs

- **Exemplo 10 – Alteração em relatório**

Situação – Um relatório (incluindo o total calculado) mostrando 16 TDs terá 3 novos TDs adicionados a ele, 3 TDs serão alterados e 2 TDs serão excluídos. O número de TDs afetados é  $3 + 3 + 2 = 8$ . A transação de saída usa 2 ARs. Os novos TDs pertencem a um novo AR (a ser adicionado). Somente um AR é afetado pela alteração e exclusão de TDs.

Questão 1 – Como determinar o percentual de alteração nos TDs e ARs?

Resposta – Meça o número de TDs alterados na função de transação em relação ao número original de TDs da transação (16).

Consequentemente o cálculo é  $8/16 \times 100 = 50\%$

Meça o número de ARs alterados na transação relativos ao número original de ARs (2).

O cálculo, nesse caso, é  $2/2 \times 100 = 100\%$

Questão 2 – Qual o tamanho da alteração da transação?

Resposta – Após a melhoria o número de TDs é 17 e são 3 ARs. A complexidade da função permanece média e seu número de pontos de função não foi alterado (5 PF). O fator de impacto é obtido na tabela

2. O percentual de alteração nos TDs (50% - primeira coluna) e ARs (100% - terceira linha) dão um fator de impacto de 0,75.

O tamanho da melhoria é

$$EFP_{\text{alterado}} = 5 \times 0,75 = 3,8 \text{ EFP.}$$

## 8.5 CENÁRIO 5 – UTILIZAÇÃO DA TABELA DE ITENS NÃO-MENSURÁVEIS

### 8.5.1 CONCEITO

O propósito da tabela de itens não mensuráveis é prover um mecanismo de remuneração de atividades de manutenção quando o esforço necessário não puder ser devidamente mensurado utilizada a técnica de APF. Exemplos dessa situação são a implementação de tabelas do tipo "code data", alterações cosméticas na interface de usuário, saídas diversas para um mesmo processo elementar (múltiplos meios) etc.

Essa tabela não deve ser aplicada durante o desenvolvimento, exceto nos casos em que houver alteração na especificação de requisitos após a entrega da primeira versão do produto para homologação, nem nos casos em que a manutenção puder ser mensurada utilizando-se a técnica padrão de APF do IFPUG ou a técnica de mensuração de melhorias da NESMA.

### 8.5.2 ABORDAGEM

Utiliza-se a tabela de itens não mensuráveis para remunerar o esforço de manutenção, quando este não for passível de mensuração pelas técnicas de APF padrão.

### 8.5.3 EXEMPLO

Após a entrega do produto para homologação, o analista de requisitos decide solicitar as seguintes modificações na aplicação.

1. Mudança da posição de um botão em 23 telas;
2. Mudança do rótulo do botão "Excluir" para "Desativar" em 23 telas;
3. Inclusão de 4 novas mensagens de usuário, alteração 7 mensagens e alteração do texto da mensagem padrão do framework para registro incluído (jCompany);
4. Inclusão de 3 novos itens de menu;
5. Exclusão de um item de menu;
6. Alteração de lista de opções da Enumeration "Estado Civil" para incluir a opção "União Estável";
7. Inclusão de mais uma tabela "Code table" e funcionalidades associadas.

Utilizando-se a tabela de itens não mensuráveis (item 6.1), podemos encontrar a equivalência em ponto de função para o esforço a ser remunerado nesta situação. Teremos:

	Especificação do item	Qtde.itens	Qtde. PF	Total PF
1	<b>Layout:</b> mudança da posição do botão "Excluir" em 23 telas.	23	0,04	0,92
2	<b>Layout:</b> mudança do rótulo do botão "Excluir" para "Desativar"	1	0,04	0,04



3	<b>Mensagens:</b> alterações de mensagens	12	0,04	0,48
4	<b>Menu:</b> inclusão de 3 novos itens	3	0,04	0,12
5	<b>Menu:</b> exclusão de um item	1	0,01	0,01
6	<b>Dados Hard Coded:</b> alteração de lista de opções de Enumeration.	1	0,04	0,04
7	<b>Code Table:</b> inclusão de uma tabela e funcionalidades CRUD associadas	1	2,7	2,7
	<b>Total</b>			<b>4,31</b>

Ou seja, esse conjunto de manutenções especificado deve ser remunerado em 4,31 PF.

## **8.6 CENÁRIO 6 – CONTAGEM ESTIMATIVA DA NESMA**

### **8.6.1 CONCEITO**

A contagem estimativa da NESMA utiliza a técnica padrão do IFPUG para identificar as funções de dados e transação e estabelece que a complexidade das funções de dados deve ser considerada baixa e a das funções de transação, média.

### **8.6.2 ABORDAGEM**

A contagem estimativa da NESMA é utilizada ao final da fase de Concepção/Iniciação, utilizando as informações obtidas durante o levantamento inicial de requisitos, consolidadas no documento de visão do produto. O propósito da contagem estimativa é fornecer uma base para o dimensionamento de esforço, custo e prazo do desenvolvimento. Essa estimativa é utilizada pela área de negócios para elaborar a proposta comercial.

### **8.6.3 EXEMPLO**

Uma aplicação mantém dados de Cliente e Produto e referencia dados de Fornecedor.

Então temos:

- ALI: Cliente e Produto
- AIE: Fornecedor

Requisitos do usuário:

- O usuário quer adicionar, alterar e excluir dados de Cliente, quer consultar Cliente e também deseja quatro relatórios diferentes sobre Cliente com dados calculados.
- O usuário quer adicionar, alterar e excluir dados de Produto, quer consultar Produto e deseja um relatório sobre Produto com dados calculados
- O usuário quer consultar Fornecedor através do número do Fornecedor e requisita também um relatório sobre Fornecedor com totalização.

<b>Funções de Dados e</b>	<b>Tipo</b>	<b>Complexidade</b>	<b>Tamanho funcional</b>
---------------------------	-------------	---------------------	--------------------------

<b>Funções de Transação</b>	<b>de Função</b>	<b>(por default)</b>	
Cliente	ALI	Baixa	7
Produto	ALI	Baixa	7
Fornecedor	AIE	Baixa	5
Incluir Cliente	EE	Média	4
Atualizar Cliente	EE	Média	4
Excluir Cliente	EE	Média	4
Consultar Cliente	CE	Média	4
Relatório 1 sobre Cliente	SE	Média	5
Relatório 2 sobre Cliente	SE	Média	5
Relatório 3 sobre Cliente	SE	Média	5
Relatório 4 sobre Cliente	SE	Média	5
Incluir Produto	EE	Média	4
Atualizar Produto	EE	Média	4
Excluir Produto	EE	Média	4
Consultar Produto	CE	Média	4
Relatório sobre Produto	SE	Média	5
Consultar Fornecedor	CE	Média	4
Relatório sobre Fornecedor	SE	Média	5
<b><i>Tamanho funcional estimado</i></b>			<b><i>85 PF</i></b>

## **8.7 CENÁRIO 7 – CONTAGEM DE ENVIO DE EMAIL**

### **8.7.1 ABORDAGEM**

O envio de e-mail só será cobrado quando for um processo elementar. Tal processo será considerado uma consulta externa (CE), pois não há qualquer tipo de processamento nele.

Quando o envio de e-mail fizer parte de alguma função de transação não será cobrado.

### **8.7.2 EXEMPLOS**

- **Exemplo 1**

Um e-mail é enviado automaticamente, uma vez por semana, para alertar o usuário sobre determinado assunto. **Será contado.**

- **Exemplo 2**

O usuário entra no sistema para enviar e-mail sobre uma determinada situação. Por que entrar no sistema? Porque ele tem características próprias como formulário, assinatura, etc. **Será contado.**

- **Exemplo 3**

O usuário está executando uma determinada função no sistema e ao final é necessário enviar um e-mail comunicando o resultado. Este envio faz parte da função de transação que está sendo executada e **não será contado.**

***Nota: Demais cenários que não constarem neste guia serão tratados oportunamente entre as partes interessadas.***