

# Medição de Pontos por Função a Partir da Especificação de Requisitos

Helena Cristina A. B. Tavares, Ana Elizabete S. Carvalho, Jaelson F. B. Castro

Serpro – Empresa do Ministério da Fazenda, Universidade Federal de Pernambuco  
[helena-sristina.tavares@serpro.gov.br](mailto:helena-sristina.tavares@serpro.gov.br), [ana-elizabete.carvalho@serpro.gov.br](mailto:ana-elizabete.carvalho@serpro.gov.br), [jbc@cin.ufpe.br](mailto:jbc@cin.ufpe.br)

**Abstract.** Neste trabalho apresentaremos uma proposta para medição de Pontos por Função a partir da especificação de requisitos expressa em casos de uso, notação UML (*Unified Modeling Language*). Com esta medição torna-se disponível uma métrica confiável na fase de especificação de requisitos do processo de desenvolvimento de software. Esta proposta visa enfatizar a importância da especificação de requisitos para o trabalho de medição de sistemas, minimizando o esforço dos líderes de projeto, obtendo uma compreensão intuitiva do porte do projeto em termos de Pontos por Função de maneira simples e dinâmica. Verifica-se que caso de uso e pontos por função podem ser usados juntos efetivamente para alcançar sucesso no gerenciamento dos requisitos e do projeto.

## 1 Introdução

Com a globalização da economia e maior competitividade no mercado, as empresas tornam-se mais dependentes dos seus sistemas de informação. Construir esses sistemas em tempo hábil para serem úteis aos negócios e com qualidade adequada é o desafio que as organizações que desenvolvem software estão enfrentando.

Como Empresa de Tecnologia da Informação, o SERPRO atua num mercado em permanente ebulição, por isso iniciou um processo para identificação e implantação das melhores práticas para elevar o estágio de maturidade do processo de desenvolvimento de *software*.

Impulsionando este processo está a meta de atingir o nível 2 de maturidade segundo o modelo CMM (*Capability Maturity Model*) do SEI até o final de 2002, com a avaliação prevista para novembro, conforme descrito em [2]. E, os trabalhos iniciais verificaram que a falta de uma efetiva gerência de requisitos era um dos principais problemas a serem superados.

Dando início à estruturação interna de uma gerência de requisitos, o SERPRO participou do projeto *Plataforma Tecnológica em Engenharia de Requisitos - Estratégias para o Aumento da Qualidade no Desenvolvimento de Sistemas* [3], o qual teve como objetivo estabelecer bases para o aumento da qualidade dos processos de produção de

software, por meio da cooperação entre universidades e empresas, com ênfase na utilização da Engenharia de Requisitos. A plataforma foi extremamente importante para um melhor conhecimento dos problemas enfrentados pelas organizações que participaram do projeto e da possibilidade da cooperação com as instituições de pesquisa para minorar esses problemas [4].

Identificados os problemas, os processos descritos em [5] foram executados para a implantação da Gerência de Requisitos no SERPRO tendo em vista a melhoria do processo preconizada para o Nível 2 do CMM. Com a efetiva utilização e ênfase dada aos requisitos durante a fase inicial do processo de desenvolvimento de software, vislumbrou-se a oportunidade e a necessidade de extrair métricas de tamanho de software a partir dos requisitos descritos em casos de uso [6], o que serviria de subsídios para as demais áreas-chaves de processo do CMM.

Concentrando esforços para descrever requisitos de software com caso de uso e medindo os requisitos utilizando a métrica Pontos por Função, projetos podem ser melhor gerenciados e controlados. O dimensionamento de projeto na fase de requisitos é um tópico que vem sendo bastante considerado quando o enfoque é Qualidade do Processo de Desenvolvimento de Sistemas. Com base no tamanho de um projeto, pode-se derivar uma série de indicadores gerenciais que proporcione uma postura pró-ativa dos gestores de desenvolvimento sob diversos aspectos.

Nos idos de 1979, em busca de uma métrica melhor, A. J. Albrecht (IBM) considerou a utilização dos aspectos externos visíveis de um software para gerar uma nova métrica, conhecida como Pontos de Função e que foi um dos primeiros métodos a medir e prever o desenvolvimento de software com alguma precisão [7].

Considerando esses tópicos, [6] propôs a medição de Pontos por Função a partir da especificação de requisitos baseada em casos de uso. Esta proposta visa antecipar o trabalho de medição de sistemas minimizando o esforço dos líderes de projeto, obtendo uma compreensão intuitiva do porte do projeto em termos de Pontos por Função a partir dos requisitos elicitados.

Na próxima Seção, são descritos trabalhos relacionados. Na Seção 3, faremos um resumo da Métrica Análise de Pontos por Função. Na Seção 4, apresentaremos uma proposta de medição de Pontos por Função a partir da descrição dos requisitos expressos em notação UML e um estudo de caso para facilitar o entendimento da proposta. E, finalmente, a Seção 5 é composta das conclusões e trabalhos futuros.

## **2 Trabalhos Relacionados**

Recentemente, tem surgido grande número de extensões para o modelo de pontos por função. Vários autores têm proposto métricas para adaptar pontos de função a software orientado a objetos. Em [10] classes são tratadas como objeto, e serviços fornecidos a clientes como transações, enquanto em [11] cada classe é considerada como um arquivo interno, e mensagens enviadas através da fronteira do sistema são tratadas como

transações. Sneed [12] propôs *pontos de objeto* como medidas para estimar tamanho de software OO. *Pontos de objeto* são derivados da estrutura de classes, de mensagens e de processos ou casos de uso e ponderados pelos fatores de ajuste.

Vários trabalhos estão sendo publicados mostrando a utilização da métrica Análise de Pontos por Função na fase de Requisitos em projetos orientados a objetos. Entre eles, destacam-se: Function Point Usage in Object Oriented Environments [13], Use Cases and Function Points [14], Estimating Software Development Effort Based on Use Cases-Experiences from Industry [23] e Function Point Measurement from Java Programs [24]. Toro relata a futura utilização de métrica de tamanho de Software a partir dos requisitos utilizando a técnica de Pontos de Caso de Uso na ferramenta descrita em [15], confirmando assim a importância da extração das métricas de tamanho de software a partir dos requisitos.

Uma proposta inicial do IFPUG [16] trata classes como arquivos e métodos como transações. Fetcke [17] define regras para mapear um modelo de casos de uso [18] para os conceitos do manual de Práticas de contagem do IFPUG, mas nenhuma tentativa foi feita para relacionar os resultados a outras métricas, tais como, pontos de função tradicionais, linhas de código ou esforço [19].

Nos ambientes de desenvolvimento de software, atualmente, temos uma enumerável quantidade de métodos de desenvolvimento, tecnologias, plataformas, configuração e acesso centralizado/descentralizado. Indiferente a metodologias e ambientes, no entanto, os pré-requisitos chave para o desenvolvimento de sistema de qualidade são entender, documentar e priorizar os requisitos do sistema [1].

A métrica de Análise de Pontos por Função mede o tamanho do software pela quantificação de sua funcionalidade externa, baseada no projeto lógico ou a partir do modelo de dados; abrange a funcionalidade específica requerida pelo usuário. Essa funcionalidade diz “o que” será entregue ao usuário.

A abordagem de casos de uso captura requisitos da perspectivas de como o usuário realmente utilizará o sistema ao invés da perspectivas das características que o sistema deve incorporar. Isto significa que o mesmo documento pode ser utilizado para contar Pontos por Função (ou para medir o projeto). Uma vez que o mesmo documento é utilizado, então é assegurado um maior nível de precisão. É muito melhor comunicar à gerência que um projeto teve um crescimento de 300 a 900 FPs que declarar que o projeto teve muito crescimento.

Casos de uso estão se tornando um método largamente utilizado para descrever os requisitos aparentemente visíveis de uma aplicação de software. Pontos por função é um método para medir software a partir de uma perspectiva de requisitos e casos de uso é um método para desenvolver requisitos. Ambos, pontos por função e casos de uso, tentam ser independentes da tecnologia utilizada para implementar a solução de software. Casos de uso são utilizados para validar um “design” proposto e para garantir que está de acordo com todos os requisitos. Novamente, um benefício de um caso de uso é tentar definir requisitos bem no início do ciclo de vida. Se casos de uso são definidos no início e pontos por função são aplicados, então as estimativas do projeto são muito mais precisas. Uma

vez que Pontos por Função possuem grandes bases históricas, é possível comparar a produtividade de utilizar métodos de casos de uso com outros métodos.

Ao saber o tamanho do projeto no início do ciclo de vida, melhores estimativas de tempo e custo podem ser desenvolvidas. Os casos de uso são mantidos atualizados, os Pontos por Função são mantidos atualizados, as estimativas do projeto podem ser facilmente atualizadas e os desacordos explicados.

O aumento do foco nos requisitos resulta em um software de maior qualidade. Casos de Uso e Pontos por Função podem ser usados juntos efetivamente para alcançar sucesso no gerenciamento dos requisitos e do projeto [8].

O principal aspecto deste trabalho é a utilização das informações contidas nos casos de uso para as regras de contagem de pontos por função tradicionais, o que permitiu a comparação de tamanho com os sistemas pontuados anteriormente que foram definidos a partir da especificação estruturada.

### 3 Análise de Pontos por Função (APF)

De acordo com a técnica Análise de Pontos por Função, uma aplicação de software, vista sob a ótica do usuário, é um conjunto de funções ou atividades do negócio que o beneficiam na realização de suas tarefas [20]. O manual do IFPUG classifica os seguintes tipos de elementos funcionais[21]:

1. Entrada Externa – EI (*External Input*) – transações lógicas onde dados entram na aplicação e mantém dados internos.
2. Saída Externa – EO (*External Output*) – transações lógicas onde dados saem da aplicação para fornecer informações para usuários da aplicação.
3. Consulta Externa – EQ (*External Query*) – transações lógicas onde uma entrada solicita uma resposta da aplicação.
4. Arquivos Lógicos Internos – ILF (*Internal Logical File*) – grupo lógico de dados mantido pela aplicação.
5. Arquivos de Interface Externa – EIF (*External Logical File*) – grupo lógico de dados referenciado pela aplicação, mas mantido por outra aplicação.

O manual do IFPUG fornece tabelas e diretrizes para determinar a complexidade de cada elemento funcional [7]. A complexidade dos ILF e EIF é baseada número de registros lógicos – RET (Record Element Types) e no número de itens de dados referenciados – DETs (Data Element Types) e a complexidade das transações é baseada no número de Arquivos Referenciados – FTR (File Types Referenced) e no número de Itens de Dados Referenciados – DET (Data Element Type).

Os elementos funcionais identificados são totalizados para calcular obtenção dos Pontos por Função não Ajustados. Então é calculado é calculado a partir de 14 (quatorze) características gerais dos projetos, que permitem uma avaliação geral da funcionalidade da aplicação: Comunicação de Dados, Processamento Distribuído, Atualização de Dados Online, Entrada de Dados Online, Volume de Transações, Eficiência do Usuário Final,

Complexidade do Processamento, Facilidade de Implantação, Multiplicidade de Locais, Facilidade de Mudanças, Facilidade Operacional, Desempenho, Utilização do equipamento, Reutilização de Código. A cada característica será atribuído um peso de 0(zero) a 5(cinco), de acordo com o Nível de Influência na aplicação. O Nível de Influência Geral é obtido pelo somatório do nível de influência de cada característica e o Fator de Ajuste é obtido pela expressão:

$$Fator\ de\ ajuste = 0,65 + (Nível\ de\ Influência\ Geral * 0,01) \quad (1)$$

O total de Pontos por Função da aplicação será encontrado mediante a multiplicação do número de Pontos por Função não ajustados pelo Fator de Ajuste:

$$Pfs\ Ajustados = Pfs\ Não-Ajustados * Fator\ De\ Ajuste \quad (2)$$

Maiores detalhes sobre a técnica podem ser obtidos em [7].

#### **4 Proposta de Medição de Pontos por Função a partir da Descrição de Requisitos expressos em Notação UML**

A medição de software através de Pontos por Função é uma ferramenta muito útil para a gerência. Observamos também que a especificações de requisitos por meio de casos de uso está sendo cada vez mais adotada e que se faz necessário realizar medições de Pontos por Função a partir das descrições dos requisitos expressos em notação UML para antecipar e facilitar o trabalho de gerenciamento.

Neste artigo apresentaremos uma descrição resumida de uma proposta de medição utilizando Análise de Pontos por Função a partir de um subconjunto da notação UML que foi apresentada em detalhes em [6].

Esta proposta de medição será discutida nas fases relacionadas a seguir:

- Apresentar estudo de caso no intuito de facilitar o entendimento da proposta;(4.1)
- Identificar e descrever os Diagramas de Casos de Uso da forma especificada nesta proposta; (4.2)
- Elaborar o Diagrama de Classe da aplicação a ser medida; (4.3)
- Especificar a fronteira da aplicação; (4.4)
- Identificar as Funções de Dados a partir do Diagrama de Classes e do Diagrama de Casos de Uso; (4.5)
- Identificar as Funções Transacionais a partir do Diagrama de Casos de Uso; (4.6)

##### **4.1 Estudo de Caso**

Para melhor entendimento das fases desta proposta, será apresentada a experiência de implantação da proposta de medição de Pontos por Função a partir da especificação de requisitos baseada em casos de uso em um sistema desenvolvido no SERPRO. O

### Medição de Pontos por Função a Partir de Especificação de Requisitos 283

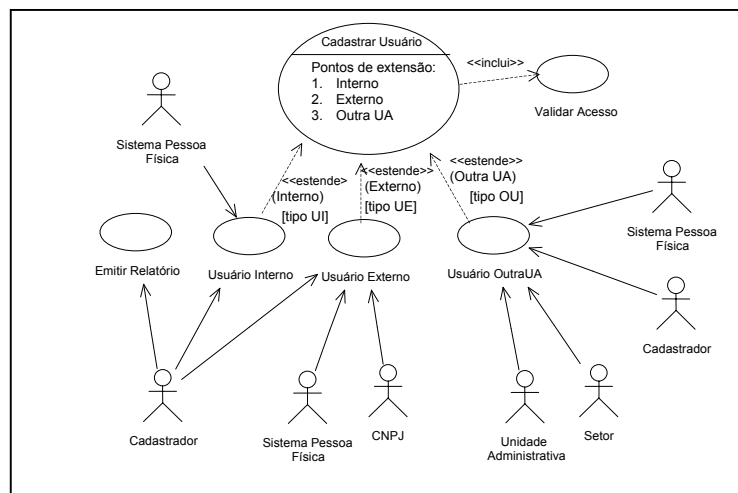
SERPRO, Serviço Federal de Processamento de Dados, é uma empresa de informática vinculada ao Ministério da Fazenda, cuja função principal é a execução de serviços de tratamento de informações e processamento de dados para o governo federal. A Empresa está sediada em Brasília, com representações regionais em 10 capitais (Belém, Belo Horizonte, Curitiba, Fortaleza, Porto Alegre, Recife, Rio de Janeiro, Salvador, São Paulo e a própria capital federal) e um quadro de 8.774 funcionários. A Empresa encontra-se estruturada em Unidades de Gestão (UG) responsáveis por um segmento da Administração Pública. Cada segmento atende a pelo menos um órgão federal com características e necessidades próprias), por intermédio de projeções da UG em cada uma das unidades organizacionais. A natureza diversa desses clientes e a descentralização do desenvolvimento em suas diversas representações exigem do SERPRO a manutenção de um complexo parque de desenvolvimento e o envolvimento direto com as mais diferentes plataformas tecnológicas. Neste cenário, a adoção de práticas para padronização, organização e controle do processo de software torna-se fundamental para a perfeita integração entre as unidades de negócio. Dentre as Superintendências, a SUNAT (Superintendência de Negócios - Administração Tributária) iniciou, em 1998, o processo de implantação de um plano de medições definindo como métrica a ser adotada a Análise de Pontos por Função (APF). A proposta apresentada neste artigo está sendo implementada em 120 projetos desenvolvidos por cerca de 700 desenvolvedores. O Sistema Senha, que será usado como exemplo, foi desenvolvido, primeiramente, utilizando-se métodos tradicionais, e, a medição deste projeto foi realizada com a métrica Análise de Pontos por Função. O total de Pontos por Função encontrado foi 186,60. Considerando a necessidade da Empresa de iniciar a migração de suas aplicações para um novo ambiente e visando obter uma maior produtividade no desenvolvimento, casos de uso passaram a ser utilizados para documentar requisitos. O Sistema Senha foi um dos primeiros projetos a serem especificados com este paradigma e a ser aplicada a proposta apresentada neste artigo. O resultado obtido nessa nova contagem foi 186,40 Pontos por Função o que implica numa diferença mínima de pontuação.

A próxima Seção relata, passo a passo, o processo de medição de algumas funcionalidades do Sistema Senha que tem como objetivo fazer o controle de acesso dos usuários às Aplicações de uma empresa, onde ocorrem interações entre o cadastrador, que inicia a ação, e o processo de inclusão de usuário. O Sistema Senha tem como finalidade cadastrar os usuários, com as respectivas permissões de acesso às transações do Sistema Integrado de Informações - SII, garantindo que apenas usuários habilitados possam acessá-las. O Sistema de Controle de Acesso – SENHA controla, verifica e gerencia os usuários, os perfis e os relacionamentos entre estes e as transações; possibilita ao projeto SII a segurança necessária e imprescindível ao ambiente e às informações; e está baseado nos seguintes conceitos:

- USUÁRIO - Pessoa física cadastrada no Sistema Integrado de Informações (SII) e habilitada para acesso às informações.
- CADASTRADOR - Usuário do Sistema Senha que exerce as funções de inclusão e habilitação de outros cadastradores, usuários e perfis do SII.

- TRANSAÇÃO - É uma operação do sistema, que executa alguma ação (inclusão, alteração, exclusão, consulta) sobre as informações do banco de dados.
- TRANSAÇÃO DE USO COMUM - Transação disponível para qualquer usuário do SII.
- PERFIL - Conjunto de transações que definem a abrangência de atuação de um cadastrador ou usuário.

Será detalhada a descrição dos Casos de Uso, na Figura 1, apenas das funções de Cadastrar Usuário, Validar Acesso e Emitir Relatório de Usuário no intuito de melhor explicar a proposta. Vale ressaltar que o detalhamento de todas as funções e a contabilização total de todo o Sistema de Controle de Acesso - Senha está disponível em [6].



**Fig. 1.** Diagrama de Casos de Uso

De acordo com os requisitos, são identificados para cadastramento três tipos de usuários com características distintas: Interno, Externo e Outra UA (Unidade Administrativa). No caso de usuário Externo deve ser informado se é uma Pessoa Física ou Jurídica (CNPJ). No caso de usuário de Outra UA (Unidade Administrativa) deve ser informada a UA e o Setor. Outro requisito da aplicação é se o usuário está autorizado a acessar a aplicação. Deverá ser emitida relação com os usuários cadastrados.

O diagrama de Caso de Uso da Figura 1 será descrito conforme orientação dada na seção a seguir.

#### 4.2 Diagramas de Caso de Uso (*Use Case*)

O desenvolvimento de Diagramas de Caso de Uso requer que a equipe de desenvolvimento de software dedique mais tempo nos estágios iniciais do plano de desenvolvimento e garanta que os requisitos sejam completos, rastreados e bem documentados.

**Tabela 1.** Descrição de Caso de Uso Cadastrar Usuário

| <i>Cadastrar Usuário – Entrada Externa</i>   |
|--|
| <p><b>Fluxo Principal de Eventos:</b><br/>                     O Use Case inicia o sistema e exibe para o cadastrador a tela para cadastrar um usuário no Sistema SENHA.<br/>                     O cadastrador deve informar o CPF e a senha do cadastrador<br/>                     Inclui Validar Acesso<br/>                     O cadastrador deve informar o Tipo do Usuário<br/>                     Ponto de extensão [Tipo UI] ← &lt;interno&gt;<br/>                     Ponto de extensão [Tipo UE] ← &lt;externo&gt;<br/>                     Ponto de extensão [Tipo UO] ← &lt;Outra UA&gt;<br/>                     Fazer a opção na tela: finalizar o use case ou inicializar</p> |
| <p><b>Fluxo Excepcional de Eventos</b> – O cadastrador pode cancelar a transação em qualquer momento.</p>  |
| <i>Cadastrar Usuário Interno – Ponto de Extensão</i>   |
| <p>Para inclusão de <b>Usuário Interno</b>, &lt;&lt;ATUALIZAR&gt;&gt;:<br/>                     CPF,<br/>                     SEQUENCIAL,<br/>                     DATA CRIAÇÃO,<br/>                     DATA EXTINÇÃO,<br/>                     DATA REGISTRO,<br/>                     DISPONIBILIDADE,<br/>                     DATA VALIDADE SENHA,<br/>                     SENHA ATUAL,<br/>                     TENTATIVAS INVÁLIDAS,<br/>                     ATIVO,<br/>                     TIPO.<br/>                     Através da interação com o ator PESSOA FÍSICA, que é o Sistema Cadastro de Pessoa Física, obteremos o nome do usuário.</p>                                 |
| <p><b>Fluxo Excepcional de Eventos</b> – No Sistema Pessoa Física o CPF do usuário não é válido, o USE CASE informa e reinicia não permitindo a inclusão do usuário.</p>   |
| <p><b>Fluxo Excepcional de Eventos</b> – O cadastrador pode cancelar a transação em qualquer momento.</p>  |

Aplicar Pontos por Função aumenta a qualidade do caso de uso. A aplicação da APF ajuda a determinar se o caso de uso está num nível de detalhe apropriado. Isto é, se é capaz de descrever como dados são passados do ator para dentro da fronteira ou como dados fluem de dentro da fronteira da aplicação para o ator, então está num nível adequado de detalhe. Por outro lado, se não é possível descrever este nível de funcionalidade, então o caso de uso necessita de mais detalhes. A idéia básica é que se é possível contar Pontos por Função, o caso de uso está num nível de detalhe apropriado. Depende de quão grande é o seu caso de uso é bom estabelecer um limite superior de tamanho de seu caso de uso de forma que seu tamanho seja controlado. Normalmente, o



tamanho máximo de um caso de uso não deveria ser maior que 50 Pontos por Função. Uma vez que um caso de uso torna-se muito grande, torna-se difícil de ser entendido.

Sugerem-se os seguintes passos para se descrever Casos de Uso de sistemas:

- Na descrição do Use Case deve-se definir claramente as informações que poderão ser atualizadas, utilizando para isso a expressão <<ATUALIZAR>>. Deve-se utilizar a expressão supracitada quando pudermos identificar que se trata de uma Entrada Externa (EI) com base nas regras do [7].
- Na descrição do Use Case deve-se definir, também, as consultas que serão realizadas utilizando a expressão <<CONSULTAR>> especificando-se quais os parâmetros de entrada e as saídas que devem ser exibidas. Deve-se utilizar a expressão supracitada quando pudermos identificar que se trata de uma Consulta Externa (EQ) com base nas regras do [7].
- Na descrição do Use Case deve-se definir quais as informações que serão apresentadas utilizando a expressão <<SAÍDAS>>. Deve-se utilizar a expressão supracitada quando pudermos identificar que se trata de uma Saída Externa (EO) com base nas regras do [7].
- Descrever, também, as informações que serão fornecidas pelo Use Case, que são consideradas como Tipo de Elemento de Dado (DET) com base nas regras do [7]. Por exemplo, uma informação de total que não seja armazenada.

**Tabela 2.** Descrição de Caso de Uso Emitir Relatório

| <i>Emitir Relatório – Saída Externa (EO)</i>  |
|---|
| <p><b>Fluxo Principal de Eventos:</b> O Use Case é iniciado quando o cadastrador solicita a emissão da Relação de Usuários. Serão exibidas as seguintes &lt;&lt;SAÍDAS&gt;&gt;:</p> <p>CPF,<br/>           NOME CPF,<br/>           DATA CRIAÇÃO,<br/>           DATA EXTINÇÃO,<br/>           DATA REGISTRO,<br/>           DISPONIBILIDADE,<br/>           ATIVO,<br/>           TIPO,<br/>           CNPJ,<br/>           NOME CNPJ,<br/>           UNIDADE ADMINISTRATIVA EXERCÍCIO,<br/>           SETOR EXERCÍCIO,<br/>           DESCRIÇÃO SETOR EXERCÍCIO,<br/>           UNIDADE ADMINISTRATIVA LOTAÇÃO,<br/>           DESCRIÇÃO UNIDADE ADMINISTRATIVA LOTAÇÃO.</p> <p><b>Fluxo Excepcional de Eventos</b> – O cadastrador pode cancelar a opção de incluir transação em qualquer momento.</p> |

Na Tabela 1, na Tabela 2 e na Tabela 3 são apresentadas as descrições dos diagramas de use case Cadastrar Usuário, Validar Usuário e Emitir Relatório respectivamente.

O importante neste ponto é quantificar o tamanho destes requisitos esboçados usando unidades de tamanho de software objetivos tais como Pontos por Função. Quantificar o

## Medição de Pontos por Função a Partir de Especificação de Requisitos 287

tamanho dos requisitos do usuário é crítico para o gerenciamento e controle do projeto, como visto anteriormente.

**Tabela 3.** Descrição de Caso de Uso Validar Acesso

| <i>Validar Acesso - Consulta Externa (EQ)</i>  |
|--|
| <p><b>Fluxo Principal de Eventos:</b><br/>O Use Case é iniciado pelo cadastrador e será usado para &lt;&lt;CONSULTAR&gt;&gt; se o cadastrador está autorizado a acessar a aplicação.<br/>Para esta validação o usuário deverá informar o CPF e a Senha. Caso sejam satisfeitas estas condições, o Senha liberará aos usuários, as transações vinculadas aos perfis a ele associado na unidade em questão.</p> <p><b>Fluxo Excepcional de Eventos</b> – O cadastrador pode cancelar a opção de incluir transação em qualquer momento pressionando o BOTÃO CANCELAR, isto reinicializará o USE CASE.</p> <p><b>Fluxo Excepcional de Eventos</b> – Caso o cadastrador não esteja autorizado a acessar esta transação específica, não será permitido o acesso para execução desta transação.</p> |

### 4.3 Diagrama de Classes

A partir das informações obtidas durante a elicitação e documentação de requisitos utilizando casos de uso, esta proposta sugere que seja elaborada uma versão inicial do diagrama de classes de forma a melhor visualizar as informações de funções de dados para a contagem. O Diagrama de Classes deve ser elaborado de acordo com a UML. Como visto no início deste artigo, estaremos utilizando um subconjunto da notação UML, ou seja, estaremos tratando no Diagrama de Classes dos relacionamentos: associação, agregação/composição e herança.

Vale ressaltar que na nossa Empresa temos todo o modelo de dados mapeado, ou seja, no início de qualquer especificação de requisitos de um projeto já temos o Diagrama de Classes. Caso não houvesse o Diagrama de Classes no momento da especificação dos requisitos poderíamos mapear as funções de dados da métrica Pontos por Função a partir dos casos de uso. Na Figura 2, apresentamos o Diagrama de Classes do Sistema Senha que de acordo com os requisitos, são identificadas três classes principais: Usuário, Perfil e Transação. Como podem existir três tipos de usuários, com características distintas, estes estão definidos num relacionamento de herança. Como as associações Usuário-Perfil e Perfil-Transação possuem propriedades próprias, são definidas as classes associação Perfil-Usuário e Transação-Perfil. Algumas classes relacionam-se a classes de outros sistemas: a superclasse Usuário com a classe Pessoa Física (independente do tipo de usuário) e, para os tipos específicos, a classe Usuário Externo relaciona-se à classe Estabelecimento, e a classe Usuário OutraUA, às classes Unidade Administrativa e Setor. A classe transação está associada à classe Aplicação a qual pertence a outro sistema.

#### 4.4 Conceitos de Fronteiras

O ponto de vista do usuário é essencial em Análise de Pontos por Função para determinar quais partes da aplicação contribuem para a funcionalidade fornecida. O conceito de fronteira de contagem é a abstração de alto nível de uma aplicação que determina o que será medido. Antes que qualquer medição se inicie, o objeto do processo de medição deve ser especificado.

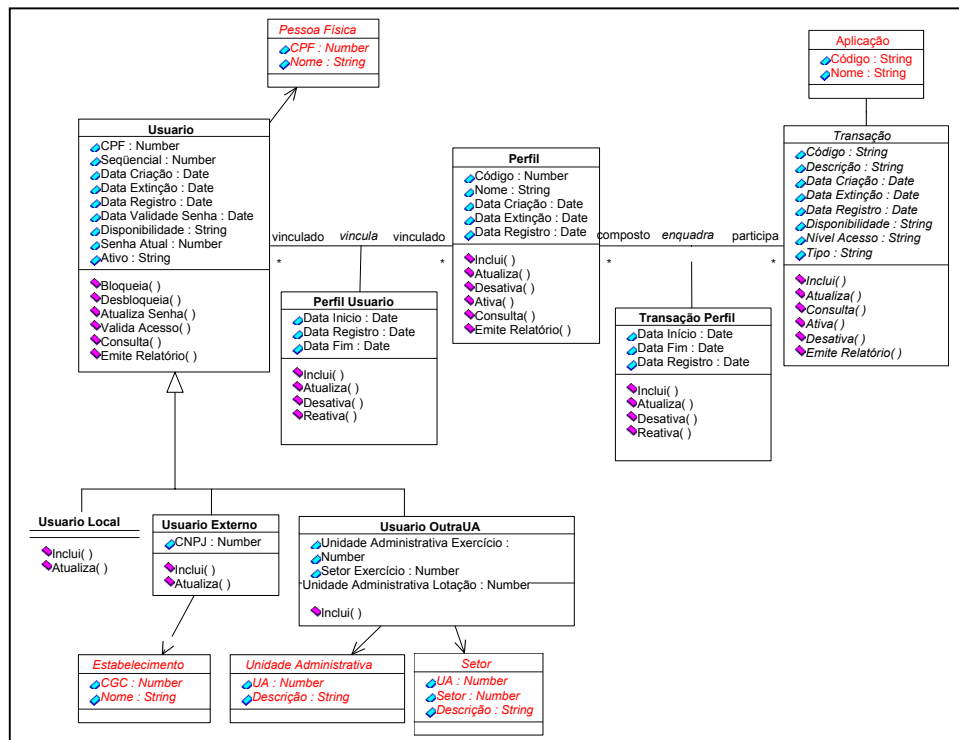


Fig. 2. Diagrama de Classes

A fronteira de contagem de Pontos por Função indica a fronteira entre a aplicação medida e as aplicações externas ou o domínio do usuário e é sempre dependente do propósito e do ponto de vista da contagem.

Uma vez que o conceito de atores corresponderá a usuários ou aplicações externas em Análise de Pontos por Função, cada tipo de usuário da aplicação será um ator. Similarmente, toda aplicação que se comunica com a aplicação considerada também deve aparecer como um ator. Desta forma, o conjunto de atores nos fornece uma visão completa dos usuários e das aplicações externas.

## Medição de Pontos por Função a Partir de Especificação de Requisitos 289

As regras seguintes foram formuladas para garantir um mapeamento consistente e coerente entre o diagrama Use Case e os procedimentos de medição de Pontos por Função. Regras propostas para Fronteiras:

- I. Considere cada ator humano como um usuário do sistema
- II. Considere cada ator não-humano, o qual não é um sistema desenvolvido apenas para fornecer funcionalidades para o sistema medido, como uma aplicação externa.

A documentação requerida para este passo é o diagrama de Casos de Uso exibindo atores e Casos de Uso em um alto nível.

Após a identificação da fronteira da aplicação, a elaboração do diagrama de classe e diagrama de Caso de Uso devidamente descrito poderemos, então, iniciar o processo de medição.

A contagem de Pontos por Função poderá mostrar, através dos questionamentos necessários ao mapeamento para identificação das funções de dados e funções transacionais, se existe falta de clareza nos requisitos, porque Pontos por Função vincula os dados armazenados (objetos) a sua manipulação e uso dentro das funções do software.

### 4.5 Identificar as Funções de Dados

Um passo fundamental da medição de Pontos por Função é a identificação das funções de dados: os Arquivos Lógicos Internos (ILFs) e Arquivos de Interface Externa (EIFs). Cada função de dado deve ser classificada de acordo com sua complexidade funcional relativa, que é baseada no número de registros lógicos, Tipo de Elemento de Registro (RET), e no número de itens de dados referenciados, Tipos de Elementos de Dados (DETs).

**Tabela 4.** Arquivos Lógicos Internos e Arquivos de Interface Externa

| ILFs  | EIFs  |
|---|---|
| USUARIO   | PESSOA FISICA   |
| PERFIL  | ESTABELECIMENTO                                       |
| TRANSACAO   | SETOR UA  |
| PERFIL POR USUARIO                                | APLICACAO   |
| TRANSACAO POR PERFIL                              | UNIDADE ADMINISTRATIVA                                |
| <b>Total Arquivos Lógicos Internos (ILFs) = 5</b> | <b>Total Arquivos de Interface Externa (EIFs) = 5</b> |

Como estamos utilizando uma prévia do diagrama de classe e o diagrama de Casos de Uso nas fases iniciais do desenvolvimento, eles serão a principal fonte de informação para identificação dos Arquivos Lógicos Internos (ILFs), Arquivos de Interface Externa (EIFs), Tipos de Elementos de Dados (DETs) e Tipo de Elemento de Registro (RETs).

Regras de mapeamento proposta para classes:

- III. Selecione toda classe como um arquivo lógico. As exceções são aquelas que fazem parte de um relacionamento de agregação/composição.
- IV. Atributos de classes são candidatos a Tipos de Elementos de Dados (DETs) para arquivos e para a transações pelas quais são lidos e/ou mantidos.

- V. Candidatos para Tipos de Elementos de Registros (RETs) são determinados pelos subgrupos dos arquivos e pelas regras de relacionamento de associação, agregação/composição e de herança.

**Tabela 5.** Grupos de elementos de dados

| <b>ILF ou EIF (Classes)</b> | <b>RETs</b>  |
|-----------------------------|--|
| USUARIO                     | USUARIO+USUARIOiINTERNO<br>USUARIO+USUARIO EXTERNO<br>USUARIO+USUARIO OUTROS |
| PERFIL                      | PERFIL   |
| TRANSACAO                   | TRANSACAO  |
| PERFIL POR USUARIO          | PERFIL POR USUARIO   |
| TRANSACAO POR PERFIL        | TRANSACAO POR PERFIL   |
| PESSOA FISICA               | PESSOA FISICA  |
| ESTABELECIMENTO             | ESTABELECIMENTO  |
| SETOR UA                    | SETOR UA   |
| APLICACAO                   | APLICACAO  |
| UNIDADE ADMINISTRATIVA      | UNIDADE ADMINISTRATIVA   |

Na figura 2 foi apresentado o Diagrama de classe do Sistema SENHA que tem como objetivo o controle de acesso aos aplicativos da empresa. Conforme a regra III as classes Usuário, Transação, Perfil, Perfil Usuário, Transação Perfil, Aplicação, Pessoa Física, Estabelecimento, Unidade Administrativa e Setor são candidatas a arquivos lógicos. A partir dos arquivos lógicos identificados, poderemos usar as regras de contagem do [7], para identificamos os Arquivos Lógicos Internos (ILFs) e Arquivos de Interface Externa (EIFs) do Sistema SENHA, como apresentado na Tabela 4.

**Tabela 6.** Tipos de elementos de dados

| <b>USUARIO</b>                                       |
|--|
| <b>Atributos da classe</b>                           |
| CPF  |
| SEQUENCIAL   |
| DATA CRIAÇÃO   |
| DATA EXTINÇÃO  |
| DATA REGISTRO  |
| DISPONIBILIDADE                                      |
| DATA VALIDADE SENHA                                  |
| SENHA ATUAL  |
| TENTATIVAS INVALIDAS                                 |
| ATIVO  |
| TIPO   |
| CNPJ   |
| UNIDADE ADMINISTRATIVA EXERCÍCIO                     |
| SETOR EXERCÍCIO                                      |
| UNIDADE ADMINISTRATIVA LOTAÇÃO                       |
| <b>Total Tipos de Elementos de Dados (DETs) = 14</b> |

## Medição de Pontos por Função a Partir de Especificação de Requisitos 291

De posse dos Arquivos Lógicos Internos (ILFs) e Arquivos de Interface Externa (EIFs) é necessário determinar suas complexidades onde precisaremos identificar os Tipos de Elementos de Registros (RETs) e os Tipos de Elementos de Dados (DETs):

Contagem de Tipos de Elementos de Registros (RETs):

Baseados nas regras de contagem do IFPUG, identificamos no diagrama de classe da figura 2 os RETs (grupos de elementos de dados reconhecidos pelo usuário) do Sistema SENHA, conforme mostrado na Tabela 5.

Contagem de Tipos de Elementos de Dados (DETs) (Data Element Types):

Baseados nas regras de contagem do IFPUG, identificamos os DETs (Tipos de elementos de dados) reconhecidos pelo cliente da classe Usuário do Sistema SENHA conforme mostrado na Tabela 6.

Adicionalmente será necessário identificar como um diagrama de classes que possua relacionamentos de associação, agregação, composição e herança poderá ser mapeado para as funções de dados: Arquivos Lógicos Internos (ILFs) e Arquivos de Interface Externa (EIFs) e seus Tipos de Elementos de Registros (RETs) e os Tipos de Elementos de Dados (DETs). Maiores detalhes podem ser obtidos em [6].

**Associação.** Regras de mapeamento proposta para relacionamentos de associação:

VI. Selecione a classe associação como um arquivo lógico, considerando, para a contagem dos Tipos de Elementos de Dados (DETs), os atributos da classe associação e os atributos que são chaves primárias das classes associadas.

VII. selecione as classes associadas como arquivos lógicos.

A classe Transação-Perfil é uma classe associação. Esta será contabilizada como um Arquivo Lógico Interno (ILF), considerando como Tipos de Elementos de Dados (DETs) os atributos Data-Início, Data-Fim e Data-Registro, Código-Perfil (chave da classe associada Perfil), Código-Transação (chave da classe associada Transação).

**Agregação.** Regras de mapeamento propostas para relacionamentos de agregação/composição:

VIII. Uma classe que agregada a outra classe, é uma candidata a um Tipo de Elemento de Registro (RET) para o arquivo relacionado à classe agregada e não a um Arquivo Lógico Interno.

**Herança.** Regras de mapeamento proposta para relacionamentos de herança:

IX. Uma classe abstrata não é uma candidata a um arquivo lógico. É uma candidata a um Tipo de Elemento de Registro (RET) em cada classe que herda suas propriedades.

X. Subclasses de uma classe concreta são candidatas a arquivos lógicos ou a um Tipo de Elemento de Registro (RET) daquela classe. Se as subclasses não são contadas como arquivos lógicos, são subgrupos opcionais do arquivo relacionado à superclasse.

A classe Usuário é considerada como Arquivo Lógico Interno (ILF) e as subclasses Usuário Local, Usuário Externo e Usuário Outra UA como RETs da classe Usuário, pois estas são subgrupos opcionais da classe Usuário.

**Candidatos adicionais a arquivos.** Alguns dados que são considerados por convenção do Pontos por Função, como arquivos internos/externos podem não ser representados em

um diagrama de classes, embora aquela funcionalidade seja requerida pelo usuário. Mensagens de erro ou textos de help, por exemplo, podem ser um requisito e necessitam de uma representação de acordo com as regras de Pontos por Função. Estas informações entretanto não são modeladas normalmente como classes, mas são contempladas nos diagramas de Casos de Uso.

- XI. Se Casos de Uso fizer uso implícito de arquivos lógicos que não são representados no diagrama de classes estes arquivos devem ser incluídos no conjunto de arquivos.

As documentações requeridas para este passo são as descrições do Diagrama de Casos de Uso e o Diagrama de Classes.

#### 4.6 Identificar as Funções Transacionais

Uma das etapas na contagem de Pontos por Função é a identificação das funções transacionais, que podem ser de três tipos: Entrada Externa (EI), Saída Externa (EO) ou Consulta Externa (EQ). Nesta seção, será investigado como, a partir da documentação inicial, as funções transacionais são identificadas.

Os Diagramas de Casos de Uso da UML correspondem a transações na técnica Análise de Pontos por Função. Portanto, um conjunto de Casos de Uso serão os principais candidatos a funções transacionais. Um Caso de Uso pode ser contado como uma ou mais transações, dependendo das tarefas que desenvolve.

Será possível escolher as candidatas a transação a partir do Diagrama de Casos de Uso que se comunicam diretamente com usuários ou aplicações externas.

Proposta para mapeamento de Funções Transacionais:

- XII. Selecione todo Caso de Uso que possui uma relação direta com um ator. Este Caso de Uso será um candidato a uma ou várias transações.
- XIII. Selecione todo Caso de Uso que estende um Caso de Uso selecionado acima como um candidato. A extensão pode incluir interação com um usuário ou uma aplicação externa.
- XIV. Cada classe mantida e/ou lida por um Caso de Uso conta como um Tipo de Arquivo referenciado (FTR) para a transação associada, se e somente se, a classe foi identificada como um arquivo.

Deve-se notar que as direções das setas no modelo de Casos de Uso não indica o tipo de transação.

A documentação requerida para este passo é o diagrama de Casos de Uso exibindo atores e Casos de Uso de alto nível. Além disso, a fronteira da aplicação identificada, é necessária como entrada.

Identificaremos os Entradas Externas (EIs), Consultas Externas (EQs) e Saídas Externas (EOs) a partir dos Diagramas de Casos de Uso e suas descrições levando-se em consideração as regras XII, XIII e XIV e as regras de contagem de Pontos por Função do IFPUG. Vale salientar que na descrição dos Diagramas de Casos de Uso deve-se ressaltar a opção de atualizar com a expressão <<ATUALIZA>>, a opção de consultar com a

## Medição de Pontos por Função a Partir de Especificação de Requisitos 293

expressão <<CONSULTA>> e a opção de exibir informações com a expressão <<SAIDAS>>.

Definição da complexidade das Entradas Externas (EIs), Consultas Externas (EQs) e Saídas Externas (EOs)

A complexidade das funções transacionais é baseada na determinação de Tipos de Elementos de Dados (DETs) e Tipos de Arquivos Referenciados (FTRs). Esta informação deve ser extraída da documentação detalhada dos Diagramas de Casos de Uso e do Diagrama de Classes.

### 4.6.1 Contagem de FTRs (File Types Referenced) para Entradas Externas (EIs)

Cada Entrada Externa deve ser classificada de acordo com sua complexidade funcional relativa, que é baseada no número de arquivos referenciados (FTRs) e no número de Itens de dados referenciados (DETs).

**Tabela 7.** Número de arquivos referenciados das EIs

| Entradas Externas (EI) | ILF apenas mantido | ILF ou EIF apenas referenciado  | ILF mantido ou referenciado | Total |
|------------------------|--------------------|---|-----------------------------|-------|
| Cadastra Usuário       | Usuário            | Pessoa Física<br>Estabelecimento<br>Setor UA<br>Unidade- Administrativa |                             | 5     |

O número de arquivos referenciados é o somatório do número de Arquivos Lógicos Internos e de arquivos de Interface Externa atualizados ou consultados na Entrada Externa, conforme apresentado na Tabela 7, o exemplo relativo ao diagrama de Casos de Uso descrito na Figura 1.

**Tabela 8.** Contagem dos DETs das EIs

| Cadastra Usuário                 |
|----------------------------------|
| CPF                              |
| SEQUENCIAL                       |
| DATA CRIACAO                     |
| DATA EXTINCAO                    |
| DATA REGISTRO                    |
| DISPONIBILIDADE                  |
| DATA VALIDADE SENHA              |
| SENHA ATUAL                      |
| ATIVO                            |
| TIPO                             |
| CNPJ                             |
| UNIDADE ADMINISTRATIVA EXERCÍCIO |
| SETOR EXERCÍCIO                  |
| UNIDADE ADMINISTRATIVA LOTAÇÃO   |
| <b>Total = 14</b>                |



Contagem de Tipos de Elementos de Dados (DETs) para Entradas Externas (EIs):

O número de itens de dados referenciados é o total de campos identificados pelo usuário que são atualizados em um Arquivo Lógico Interno por uma Entrada Externa.

Para contagem dos Tipos de Elementos de Dados (DETs), iremos considerar as informações do diagrama de classes da Figura 2 e a descrição do use case da Tabela 1, conforme mostrado na Tabela 8.

De acordo com o número de itens de dados (DETs) e de arquivos referenciados (FTRs), classificam-se as Entradas Externas em simples, médias ou complexas.

Para o exemplo acima teremos 5 Tipos de Arquivos Referenciados (FTRs) e 16 Tipos de Elementos de Dados (DETs), onde 14 foram obtidos a partir dos Tipos de Arquivos Referenciados (FTRs) e 2 são decorrentes de Itens de dados adicionais( teclas de função e mensagens de erro) o que implica em uma complexidade ALTA, ou seja, corresponde a 6 Pontos por Função para a Entrada Externa Cadastra Usuário.

#### 4.6.2 Contagem de FTRs para Saídas Externas (EOs)

Cada Saída Externa deve ser classificada de acordo com sua complexidade funcional relativa, que é baseada no número de arquivos referenciados e no número de itens de dados referenciados.

**Tabela 9.** Contagem dos DETs das EOs

| EO<br>relatorio_usuario          |
|----------------------------------|
| CPF                              |
| NOME                             |
| SEQUENCIAL                       |
| DATA CRIACAO                     |
| DATA EXTINCAO                    |
| DATA REGISTRO                    |
| TIPO                             |
| DISPONIBILIDADE                  |
| VALIDADE SENHA                   |
| SENHA ATUAL                      |
| CNPJ                             |
| NOME                             |
| UNIDADE ADMINISTRATIVA EXERCÍCIO |
| SETOR EXERCÍCIO                  |
| UNIDADE ADMINISTRATIVA LOTAÇÃO   |
| <b>Total = 7</b>                 |

O número de arquivos referenciados é o somatório do número de Arquivos Lógicos Internos e de Arquivos de Interface Externa consultados pela Saída Externa, conforme apresentado na Tabela 10, o exemplo relativo ao diagrama descrito na Figura 1.

Contagem de Tipos de Elementos de Dados (DETs) para Saídas Externas (EOs):

O número de itens de dados referenciados é o total de campos identificados pelo usuário que aparecem na Saída Externa.

## Medição de Pontos por Função a Partir de Especificação de Requisitos 295

Para contagem dos Tipos de Elementos de Dados (DETs) de Saídas Externas (EOs), iremos considerar as informações do diagrama de classes da Figura 2, e a descrição do use case da Tabela 2, conforme mostrado na Tabela 9.

De acordo com o número de itens de dados (DETs) e de arquivos referenciados (FTRs), classificam-se as Saídas Externas em simples, médias ou complexas.

**Tabela 10.** Número de arquivos referenciados das EOs

| Saída Externas (EI) | ILF apenas mantido | ILF ou EIF apenas referenciado | ILF mantido ou referenciado | Total |
|---------------------|--------------------|--------------------------------|-----------------------------|-------|
| Relatório           |                    | Usuário                        | Pessoa Física               | 3     |
| Usuário             |                    |                                | Estabelecimento             |       |

Para o exemplo acima teremos 3 Tipos de Arquivos referenciados (FTRs) e 7 Tipos de Elementos de Dados (DETs) o que implica em uma complexidade MÉDIA, ou seja, corresponde a 5 Pontos por Função para a Saída Externa Emite Relatório de Usuário.

### 4.6.3 Contagem de FTRs (File Types Referenced) para Consultas Externas (EQs)

Cada Consulta Externa deve ser classificada de acordo com sua complexidade funcional relativa, que é baseada no número de arquivos referenciados e no número de itens de dados referenciados.

**Tabela 11.** Contagem de FTRs

| Consulta Externa (EI) | ILF ou EIF apenas referenciado | Total |
|-----------------------|--------------------------------|-------|
| Valida Acesso         | Usuário                        | 1     |

O número de arquivos referenciados é o somatório do número de Arquivos Lógicos Internos e de Arquivos de Interface Externa acessados na Consulta Externa, conforme apresentado na Tabela 11, o exemplo relativo ao diagrama de caso de uso descrito na Figura 1.

**Table 12.** Contagem de DETs

|                      |
|----------------------|
| <b>EQ</b>            |
| <b>valida acesso</b> |
| CPF                  |
| SENHA                |
| VALIDACAO            |
| <b>Total = 3</b>     |

Contagem de Tipos de Elementos de Dados (DETs) para Consultas Externas (EQs):

O número de itens de dados referenciados é o número de parâmetros informados para obtenção do resultado da Consulta Externa mais os itens de saída, contando os repetidos apenas uma vez.

Para contagem dos Tipos de Elementos de Dados (DETs), iremos considerar as informações do diagrama de classe da Figura 2 e a descrição do caso de uso da tabela 3, conforme mostrado na Tabela 12.

De acordo com o número de itens de dados e de arquivos referenciados, classifica-se a Consulta Externa em simples, média ou complexa. Para este exemplo teremos 1 Tipo de Arquivo referenciado (FTR) e 3 Tipos de Elementos de Dados (DETs) o que implica em uma complexidade BAIXA, ou seja, corresponde a 3 Pontos por Função para a Consulta Externa Valida Acesso.

#### 4.6.4 Contribuição dos Entradas Externas (EIs), Consultas Externas (EQs) e Saídas Externas (EOs)

Conclui-se a contagem das funções transacionais fazendo um balanço da contagem dos Entradas Externas (EIs), Consultas Externas (EQs) e Saídas Externas (EOs) conforme Tabela 13.

Tabela 13. Contribuição Funções Transacionais

| Tipo de Função          | Complexidade Funcional | Total de Pontos de Função |
|-------------------------|------------------------|---------------------------|
| Entradas Externas (EI)  | 0 Baixa x 3 = 0        | 6                         |
|                         | 0 Média x 4 = 0        |                           |
|                         | 1 Alta x 6 = 6         |                           |
| Consultas Externas (EQ) | 1 Baixa x 3 = 3        | 3                         |
|                         | 0 Média x 4 = 0        |                           |
|                         | 0 Alta x 6 = 0         |                           |
| Saídas Externas (EO)    | 0 Baixa x 4 = 0        | 5                         |
|                         | 1 Média x 5 = 5        |                           |
|                         | 0 Alta x 7 = 0         |                           |

Depois de calculado o total de Pontos por Função não ajustados com base no total nas funções de dados e nas funções transacionais (ILFs + EIFs + EIs + EQs + EOs) , poderemos calcular o fator de ajuste para o total dos Pontos por Função e o total de Pontos por Função da aplicação conforme apresentado na Seção 3.

## 5 Conclusões e Trabalhos Futuros

Mostramos por meio do estudo de caso a possibilidade de Medição de Pontos por Função a partir de Especificação de Requisitos. O diagrama de Casos de Uso em conjunto com o diagrama de classes fornecem uma direção para a contagem de Pontos por Função. É fundamental termos disponível uma métrica confiável para as primeiras fases do processo de desenvolvimento de software. A especificação de requisitos com a notação UML e a Medição de Pontos por Função podem ser usados em conjunto, efetivamente, para alcançar sucesso no controle e gerenciamento dos projetos.

Análise de Pontos por Função podem ser facilmente aplicados ao método de casos de uso, um dos problemas com pontos por função tem sido um conjunto de requisitos incompleto ou inconsistente, adotando ambos, método de caso de uso e Análise de Pontos por Função, a qualidade do documento de requisitos pode ser melhorada, a medição é melhorada e a estimativa como um todo é melhorada.

No SERPRO, a SUNAT (Superintendência de Negócios - Administração Tributária) iniciou, em 1998, o processo de implantação de um plano de medições definindo como métrica a ser adotada a Análise de Pontos por Função (APF), baseando-se no padrão internacional estabelecido pelo IFPUG (International Function Points Users Group). Quando utilizada em combinação com outras medidas, a Análise de Pontos por Função (APF) pode ser utilizada para determinar, por exemplo, o nível de produtividade da equipe, o esforço requerido para o desenvolvimento do sistema, o custo, a taxa de produção e de manutenção ou até a qualidade do sistema do ponto de vista do usuário. Com a compatibilização da Contagem de Pontos por Função no paradigma Tradicional e Orientada a Objeto, aproveitamos a cultura existente na Empresa em Pontos por Função.

A métrica de Análise de Pontos por Função já está sendo amplamente utilizada por todos os Pólos de Desenvolvimento da SUNAT, no Brasil, além de outras Superintendências. Foram pontuados 100% dos projetos da SUNAT. Com a familiarização dos líderes de projeto e criação da base de histórico (*baseline*), nessa primeira fase, está sendo possível avaliar o acervo de projetos, além de gerar subsídios para a tomada de decisões. Para tal foi utilizada uma ferramenta chamada Estimativa[21], desenvolvida pelo SERPRO, para ajudar na contagem e armazenamento de dados históricos.

Podemos verificar que com a medição dos Pontos por Função a partir da especificação dos requisitos obtivemos, na fase inicial do projeto, uma estimativa do total de Pontos por Função do projeto (186,40), praticamente igual a medição do projeto desenvolvido com o paradigma tradicional (186,60). Vale salientar que o projeto foi desenvolvido simultaneamente por duas equipes distintas, onde uma utilizou a especificação de requisitos tradicional e a outra a especificação de requisitos orientada a objeto. Na conclusão do projeto o total de pontos por função obtidos foi de 201. Esta experiência foi realizada para podermos verificar a validade da aplicação da proposta de medição apresentada em [6].

Durante o processo de implantação das práticas necessárias para atingir o Nível 2 do CMM, utilizamos esta Proposta de Medição em mais 120 projetos da organização e verificamos a exatidão das estimativas realizadas nas pontuações dos diversos projetos na fase de requisitos.

## Bibliografia

1. Greer, D., Bustard, D.W., Sunazuka, T.: Prioritisation of System Changes using Cost-Benefit and Risk Assessments. Fourth IEEE International Symposium on Requirements Engineering RE'99. Limerick, Ireland. (1999) 180--187
2. Tavares, H., Paim, F., Carvalho, A.: Implantando CMM Nível 2: A Estratégia SERPRO. I Simpósio Brasileiro de Qualidade de Software SBQS. Gramado, Brasil. (2002)
3. Leite, J., Castro, J., Pinheiro, F.: Plataforma Tecnológica em Engenharia de Requisitos – Estratégias para o Aumento da Qualidade no Desenvolvimento de Sistemas. <http://www.cic.unb.br/~facp/per/perhome.html>

4. Carvalho, A. E., Tavares, H. C., Castro, J.: Uma Estratégia para Implantação de uma Gerência de Requisitos visando a Melhoria dos Processos de Software. WER01. Buenos Aires, Argentina. (2001)
5. Carvalho, A. E.: Uma Estratégia para Implantação de uma Gerência de Requisitos visando a Melhoria dos Processos de Software. Dissertação de Mestrado. UFPE, Brasil. (2001)
6. Tavares, H. C.: Medição de Pontos por Função a partir da Especificação Orientada a Objeto. Dissertação de Mestrado. UFPE, Brasil. (1999)
7. IFPUG: Medição de Pontos por Função a partir da Especificação Orientada a Objeto. Function Point Counting Practices Manual, Versão 4.1.1. International Function Point Users Group <http://www.ifpug.org> (2000)
8. Dekkers, C.: Use Cases And Function Points -- Where's The Fit? Quality Plus Technologies, inc. (2001)
9. Booch, G., Rumbaugh, J., Jacobson, I.: The Unified Modeling Language – User Guide. Addison Wesley. (1999)
10. Schooneveldt, M.: Measuring the size of object oriented system. In Proc. 2nd Australian Conference on Software Metrics. Australian Software Metrics Association. (1995)
11. Whitmire, S.: Applying function points to object-oriented software. In Software Engineering Productivity Handbook. McGraw-Hill. (1993) 229--244
12. Sneed, H.: Estimating the Costs of Object-Oriented Software. In Proceedings of Software Cost Estimation Seminar. (1995)
13. Couturier, G.: FP Usage in OO Technology Environment. International Function Point. New Orleans, LA, USA. (1999)
14. Longstreet, D.: Use Cases And Function Points. Inc. (2001)
15. Toro, A. D., Cortés, A. R., Corchuelo, R., Toro, M.: Supporting Requirements Verification Using XSLT. IEEE Intl. Symposium on Requirements Engineering (RE02). Essen, Germany. (2002)
16. IFPUG: Function Point Counting Practices: Case Study 3 – Object Oriented Analysis. Object Oriented Design (Draft). International Function Point Users Group. (1995)
17. Fetcke, T., Abran, A., Nguyen, T. H.: Mapping the OO Jacobon approach to function point analysis. In Proc. IFPUG 1997. Spring Conference. (1997) 134--142
18. Jacobson, I., Christerson, M., Jonsson, P., Overgaard, G.: Object Oriented Software Engineering: A Use Case Driven Approach. Addison-Wesley. (1992)
19. Antoniol, G., Calzolari, F., Cristoforetti, L., Fiutem, R., Caldiera, G.: Adapting Function Points to Object Oriented Information Systems. 10th Conference on Advanced Information Systems Engineering [CAISE-98]. Pisa, Italia. (1998) 8--12
20. Castro, J. F.: Introdução à Medição de Software através de Ponto de Função. XII Simpósio Brasileiro de Engenharia de Software. Maringá, PR – Brasil. (1998)
21. Hastings, T.: Adapting Function Points to contemporary software systems: A review of proposals. Second Australian Conference on Software Metrics. University of New South Wales, Sydney, Australia. (1995)
22. Candéas, A., Lopes, C.: ESTIMATIVA: Uma Ferramenta para Agilizar o Dimensionamento de Projetos no SERPRO com base na metodologia de Análise de Pontos por Função. XIII Simpósio Brasileiro de Engenharia de Software - SBES'99. Florianópolis/SC, Brasil. (1999)
23. Anda, B., Dreiem, H., Jorgensen M.: Estimation Software Development Effort Based on Use Cases-Experiences from Industry. Fourth International Conference on the Unified Modeling Language. Toronto, Canada, september (2001)
24. Kusumoto, S., Imagawa, M., Morimoto, S.: Function Point Measurement from Java Programs. 24 International Conference on Software Engineering. Orlando, Flórida. (2002)