

# **Estimativa de Esforço para Projetos de Software em uma Fábrica de Software Orientada a Produto**

**Keilla Wanderley Costa, Hermano Perreli de Moura**

Centro de Informática Universidade Federal de Pernambuco (UFPE)  
Caixa Postal 7851 – 50.732-970 – Recife – PE – Brasil  
[keilla@nlink.com.br](mailto:keilla@nlink.com.br), [hermano@cin.ufpe.br](mailto:hermano@cin.ufpe.br)

## **Resumo**

Este artigo aborda as dificuldades encontradas quanto à estimativa de esforço para projetos de software em uma fábrica de software orientada a produto que possui, como produto, um sistema de Gestão Hospitalar já consolidado. Baseado na sua realidade, o artigo sugere a técnica de Análise de Pontos de Função (APF) como uma ferramenta adequada para auxiliar a área de TI; inicialmente a APF será utilizada para estimar projetos de software, mas seu uso pode ser estendido para medir a produtividade da equipe, a qualidade do produto e ainda permitir a criação de um ambiente padrão na área de TI que permitirá estabelecer uma comparação de seu desempenho com o do mercado.

## **Abstract**

This article explains how difficult is to estimate the effort for software's projects in a Software House that has Health Care Systems as its main product. Based in its reality, the article suggests the technique of Function Point Analysis(FPA) as an adjusted tool to assist the area of IT. Initially, the FPA will be used to estimate software's projects, but its use can be extended to measure: the team's productivity, the product's quality and still create a standard environment to compare software.

**Palavras-chave:** Análise de pontos de função, métrica de software.

## **1. Introdução**

A evolução do uso da informação no mundo dos negócios não pode mais ser ignorada, pois é fator determinante de competitividade das empresas, independente do seu ramo de atividades. Atualmente há uma grande preocupação em obter respostas para perguntas do tipo: estamos usando adequadamente a informação? possuímos recursos tecnológicos suficientes? qual o esforço para desenvolver software? qual o custo do software? qual a taxa de produção e manutenção de software? Tem-se consciência da importância de se medir software, porém selecionar uma métrica adequada para o gerenciamento de TI não é uma tarefa simples, especialmente para se estimar projetos de software devido à natureza dinâmica da área. Surgem as indagações: quais as métricas de desempenho de sistemas mais usadas? qual a que melhor se aplica à realidade da nossa empresa? [2]

O uso de uma métrica procura tornar a estimativa para realizar a atividade de desenvolvimento e manutenção de software mais próxima do real, ou seja, deve possibilitar que as estimativas sobre o comportamento das principais variáveis

envolvidas (custo, tempo e qualidade) tenham maior probabilidade de se confirmar nas suas respectivas medições.

Análise por Ponto de Função é uma técnica que dimensiona uma aplicação na visão do usuário final, sem se prender a características técnicas da linguagem utilizada, ou seja, independente da tecnologia adotada para a implementação [1]. Através dela é possível medir a produtividade associada ao esforço humano no processo de desenvolvimento, otimização e manutenção de sistemas, calcular o esforço despendido por unidade ou por atividade no departamento de TI, e ainda fornecer base para melhor compreensão das falhas e dos problemas de planejamento dos projetos já concluídos ou em andamento, de modo a permitir a realização de projetos futuros mais eficientes, além de prover uma unidade padrão para comparação de software.

Em sistemas de informação, cada aplicação é diferente, o processo de desenvolvimento varia e as atividades executadas são significativamente afetadas pelas habilidades dos analistas e/ou desenvolvedores. Devido a estas variáveis, este processo torna-se muito complexo; contudo, é importante que a produtividade do departamento de informática seja medida através de um método independente do tipo do sistema, da tecnologia utilizada para desenvolver o produto e da habilidade dos desenvolvedores [4].

Baseado nesse universo, o objetivo principal dessa pesquisa é a análise da métrica utilizada por uma empresa de informática para estimativa de esforço necessário para desenvolvimento dos seus projetos de software. O resultado da análise permitiu enumerar os pontos falhos encontrados e, a partir deles, sugerir melhorias no processo.

## **2. A Empresa**

A empresa foco da análise é uma empresa sólida, com mais de doze anos no mercado de software para Gestão Hospitalar. Ela possui um sistema desenvolvido há vários anos, que contempla praticamente todos os segmentos de negócio dentro do âmbito hospitalar, como Custos, Internação, Centro Cirúrgico, Banco de Sangue, Estoque, Farmácia e Faturamento de Convênios, entre outros. Por dispor de um sistema pronto, que atende satisfatoriamente a todos os seus clientes, a empresa não desenvolve produtos novos nessa área. A sua linha de trabalho é realizar customizações e alterações no sistema já consolidado.

## **3. Metodologia**

A pesquisa envolveu levantamento das estimativas de tempo de projetos realizados na área de Faturamento Hospitalar já concluídos, análise de dados e entrevistas com analistas da empresa. Os questionários utilizados estão disponíveis nos Anexos.

É importante ressaltar que não foi feito um estudo para saber que amostra de dados deveria ser utilizada, o levantamento dos dados concentrou-se na leitura de horas estimadas e realizadas para implementação e testes nos projetos concluídos pela empresa nos últimos seis meses.

#### 4. Cenário Atual

É utilizada uma planilha Excel, desenvolvida há pouco mais de dezoito meses por um grupo de analistas da própria empresa para estimar o esforço humano necessário (n.º horas/homem) para realizar um projeto de software. O tempo total é composto pelo número de horas necessário para:

- especificação funcional
- especificação técnica
- desenvolvimento
- validação e testes

Como a empresa já possui um produto comercializado, os projetos são customizações ou alterações/melhorias a funcionalidades já existentes no sistema. Para cada nova funcionalidade desejada ou aperfeiçoamento de uma já existente, o cliente abre uma solicitação<sup>1</sup> descrevendo o que deseja e a empresa se encarrega de elaborar o documento de especificação funcional e estimar com auxílio da planilha o total de horas necessária para fornecer a funcionalidade solicitada.

A planilha só auxilia a estimativa de tempo para as fases de Desenvolvimento e Testes. O analista informa o número de horas efetivamente gasto para realizar a Especificação Funcional e estima de acordo com a sua experiência o tempo necessário para elaborar a Especificação Técnica. A planilha sugere os valores para as fases de Desenvolvimento e Testes, baseado no preenchimento dos tópicos abordados nela. Estes tópicos são informações técnicas voltadas para a ferramenta de desenvolvimento da Oracle: Forms e Reports, utilizada em 80% das aplicações da empresa.

Existem, em pequena quantidade, algumas funcionalidades da aplicação principal desenvolvidas em outra tecnologia. Contudo, projetos demandados para essa área têm seus esforços em horas estimados através de estimativas diretas, ou seja, levando em consideração apenas a experiência dos analistas, sem fazer uso de nenhuma métrica.

Atualmente, o tempo estimado para validação e testes é feito baseado no total de horas destinado à implementação, quantificado em 50% desse total.

Como exemplo, suponha que um analista gastou 12 horas para elaborar a especificação funcional de uma solicitação, e após concluí-la, preencheu a planilha e estimou 8 horas para fazer a sua especificação técnica. Baseada nas informações de preenchimento, a planilha sugeriu um número de 70 horas para implementação e 35 horas para testes, resultando num total de 125 horas. Esse total multiplicado pelo valor hora/homem em Real é o valor do projeto orçado para o cliente. Caso ele não aprove o orçamento, o custo referente à especificação funcional é arcado pelo cliente.

Antes de ser criada a planilha, a empresa utilizava práticas da estimativa Wide Band Delphi, sem usá-la na sua totalidade. A estimativa Wide Band Delphi é uma técnica de grupo iterativa que permite que um participante melhore suas estimativas individuais a partir da colaboração dos demais especialistas. A técnica é composta por duas etapas; na primeira, é selecionada uma equipe de dois ou mais profissionais experientes para

---

<sup>1</sup> Solicitação é o termo usado na empresa para representar cada demanda de um projeto.

estimar individualmente o tempo para desenvolvimento do projeto, e na segunda, é feita uma reunião com os profissionais envolvidos, para juntos discutirem os tempos estimados individualmente até chegarem a um consenso e definirem o esforço em horas do projeto. A empresa não realizava a primeira etapa, sendo feita de imediato a reunião com dois ou mais analistas para se estimar o número de horas necessárias por projeto.

Com o aumento do número de projetos e a diversidade de experiência dos profissionais, sentiu-se a necessidade de se criar uma uniformidade na estimativa dos projetos dentro da empresa para que todos os profissionais utilizassem os mesmos critérios ao estimar o esforço em horas de um projeto, diminuindo a dependência do profissional. Estas foram uma das maiores motivações para elaboração da planilha para estimativa dos projetos de software.

Depois de um ano de uso, a planilha está passando por um processo de melhoria, o qual contempla basicamente a criação de um manual de preenchimento da mesma, a criação de níveis de complexidade com pesos diferenciados em tópicos já abordados e a divisão de tópicos que comparam ordem de grandeza diferentes. A nova planilha ainda não está em operação, sendo prevista a sua implantação dentro dos próximos dois meses.

A Figura 1 apresenta trecho da planilha utilizada atualmente e a Figura 2 apresenta a nova versão que está em elaboração. Observando-se as duas figuras, pode-se notar a criação do nível de complexidade, que pode variar de Muito Simples a Muito Complexo recebendo ponderações diferentes.

| Carcterísticas            | Quantidade | Total da Pontuação | Peso |
|---------------------------|------------|--------------------|------|
| <b>Banco de Dados</b>     |            |                    |      |
| Tabelas e Views Simples   | 0          | 0,00               | 1,00 |
| Tabelas e Views Médias    | 0          | 0,00               | 2,00 |
| Tabelas e Views Complexas | 0          | 0,00               | 3,00 |
| Colunas e Índices         | 0          | 0,00               | 1,00 |
| Triggers Simples          | 0          | 0,00               | 1,00 |
| Triggers Médias           | 0          | 0,00               | 2,00 |
| Triggers Complexas        | 0          | 0,00               | 3,00 |

Fonte: trecho da planilha utilizada atualmente pela empresa

**Figura 1**

| Características       | Nível | Quantidade | Total da Pontuação | Peso |
|-----------------------|-------|------------|--------------------|------|
| <b>Banco de Dados</b> |       |            |                    |      |
| Tabelas               |       | 0          | 0,00               | 2,00 |
| Colunas               |       | 0          | 0,00               | 1,00 |
| Índices               |       | 0          | 0,00               | 1,00 |
| Views                 |       | 0          | 0,00               | 0,00 |
| Views                 |       | 0          | 0,00               | 0,00 |
| Triggers              |       | 0          | 0,00               | 0,00 |
| Triggers              |       | 0          | 0,00               | 0,00 |
| Triggers              |       | 0          | 0,00               | 0,00 |

Fonte: trecho da nova versão da planilha

**Figura 2**

## **5. Análise do Cenário Atual**

Apesar da planilha apresentada na Seção 4 ser utilizada em todas as solicitações, percebe-se que as pessoas têm dificuldade de utilizá-la de forma adequada. Isso ocorre, entre outros, por diversos fatores:

- 5.1 falta de documentação;
- 5.2 limitações da planilha;
- 5.3 especificações complexas;
- 5.4 documentos de especificação de requisitos não detalhados;
- 5.5 falta de hábito para utilização de padrões de desenvolvimento de software.

### **5.1 Falta de Documentação**

A falta de documentação da planilha e de um manual de preenchimento faz com que muitos profissionais da empresa desconheçam a maneira de como preenchê-la, gerando dúvidas do significado de vários itens contidos nela. A treinamento de uso, ministrado por uma das pessoas que participou da sua elaboração, foi dado a um grupo de analistas e estes ficaram de disseminar o conhecimento informalmente entre os demais da empresa.

### **5.2 Limitações da Planilha**

A planilha contém tópicos muito subjetivos como Views, Triggers, Cálculos, que na abordagem atual não nos permite definir melhor a complexidade de cada um dentro do projeto em questão. A planilha ainda sofre a limitação de não poder ser aplicada em todas as áreas de sistemas da empresa, pois como é voltada para uma ferramenta de desenvolvimento específica, não atende setores que desenvolvem em outra plataforma.

Com as limitações para se detalhar melhor o grau de complexidade do que está sendo desenvolvido, em muitos casos, o número de horas estimado fica muito inferior ao esforço necessário, e também há casos em que esse número ultrapassa muito o número de horas requerido.

Por isso é muito comum estimar o esforço necessário baseado na experiência do profissional que está especificando e preencher a planilha posteriormente de forma a chegar ao número de horas desejado, ou preenchê-la e depois de observar o resultado sugerido por ela fazer-se ajustes em seu preenchimento para gerar novo número de horas para mais ou para menos de acordo com o desejado.

### **5.3 Especificações Complexas**

Em projetos grandes, com funções complexas englobando iterações entre mais de um módulo do sistema, a dificuldade de preenchimento da planilha é maior, pois o número de informações é grande e a distribuição delas dentro da planilha não é muito clara.

### **5.4 Documentos de Especificação de Requisitos não Detalhados;**

A equipe de analistas, por muitas vezes, não consegue atender a toda a demanda de

solicitações oriundas de clientes devido ao seu número insuficiente de pessoas. Para agilizar as atividades de quem está com um acúmulo de projetos a serem especificados, há uma prática de se elaborar especificações de requisitos não detalhadas. Porém, isso dificulta o preenchimento da planilha, uma vez que para utilizá-la da melhor forma é necessário saber toda a abrangência da alteração.

### 5.5 Falta de Hábito para Utilização de Padrões de Desenvolvimento de Software;

Como a empresa passou boa parte da sua existência sem fazer uso de uma métrica para estimar esforço em projetos de software e ainda não possui uma metodologia de desenvolvimento de sistemas bem definida, as pessoas não têm o hábito de utilizar padrões, de documentar, Até recentemente todo o tempo estimado era baseado nas experiências individuais dos analistas.

Com isso, vê-se que apesar de ter sido estabelecida uma métrica para se tornar impessoal a estimativa de horas para um projeto de software, a dependência à experiência do profissional que está especificando continua existindo, principalmente pelo fato da planilha não atender a todas áreas da empresa.

Com a análise dos dados gerados por projetos passados pôde-se observar que o quantitativo de horas despendida para testes está sendo bem inferior ao sugerido pela planilha. Isso ocorre principalmente por dois fatores importantes: o primeiro é pelo fato da empresa não possuir um plano de testes bem definido. Testa-se a partir da especificação funcional do projeto, porém não é gerado um documento onde é dito o que testar, como testar e também não é gerado um documento do resultado dos testes realizados. O outro ponto é que a estimativa de testes está ligada ao esforço necessário para codificação, o qual pode ser bem diferente do tempo necessário para testá-lo.

No período avaliado, os projetos realizados na área de Faturamento de Convênios, um dos módulos do Sistema, tiveram seu esforço em horas distribuído conforme Tabela 1.

Observando-se o gráfico da Figura 3, pode-se concluir que apesar da diferença entre o planejado e o realizado do projeto total ter sido pequena, as fases apresentaram alto índice de variação.

| Fases           | Planejado em horas | Realizado em horas | %       |
|-----------------|--------------------|--------------------|---------|
| Desenvolvimento | 3192               | 3689               | > 15,57 |
| Teste           | 1596               | 878                | < 44,98 |
| TOTAL           | 4788               | 4567               | < 4,62  |

**Tabela 1 – Disposição em horas do Planejado e do Realizado dos Projetos nos últimos 6 meses**

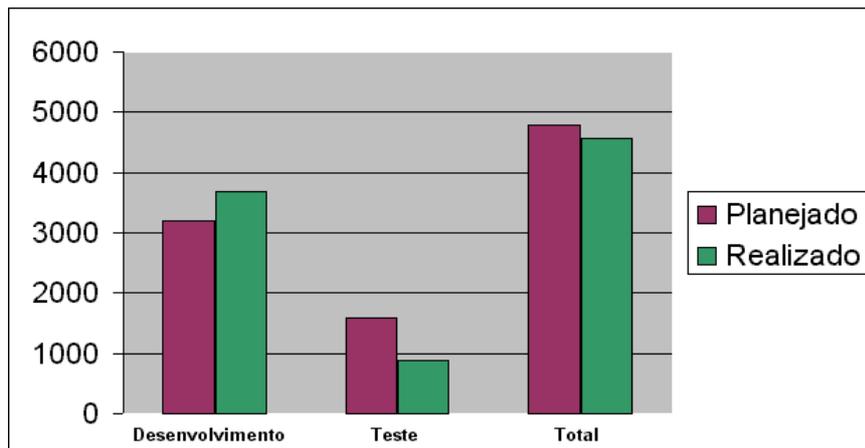


Figura 3 – Comparativo entre as horas planejadas e efetivamente realizadas

## 6. Análise de Pontos de Função

Apresenta-se nesta seção uma visão geral de um método para estimativa de esforço – a análise de pontos de função.[1] [3] [5]

### 6.1 Significado e Propósito dos Pontos de Função

A Análise de Pontos de Função tem como propósito medir um sistema baseado nas suas características funcionais e nos objetivos identificados a partir da visão do usuário, independente da tecnologia empregada para o desenvolvimento, ou do conhecimento, habilidade e experiência do grupo que desenvolve ou faz a manutenção do sistema.

### 6.2 Objetivos e Benefícios da Técnica:

- Fornecer ferramenta para auxiliar nas estimativas de desenvolvimento de software;
- Avaliar com base nas funções que o sistema executa, considerando apenas os componentes da aplicação definidos e requeridos pelo usuário, independente de tecnologia;
- Identificar um padrão de medida para a produtividade e qualidade;
- Prover uma unidade padrão de medida para comparação de software.

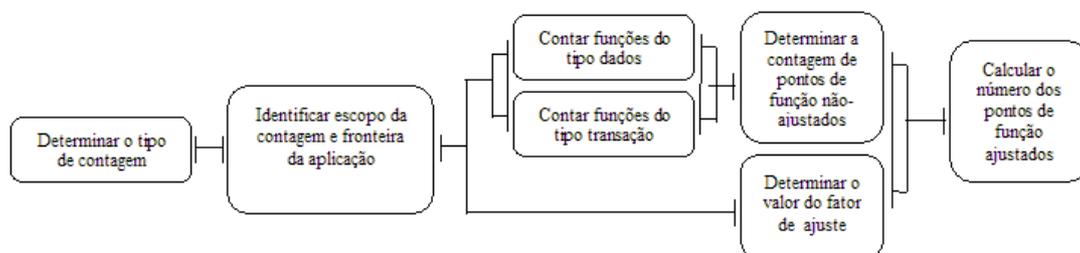


Figura 4 – Etapas da Análise de Pontos de Função [1]

### 6.3 As Etapas

A Figura 4 representa bem as etapas da Análise de Pontos de Função.

- Tipo de Contagem;
- Escopo da Contagem e a Fronteira da Aplicação;
- A Contagem de Pontos de Função Não Ajustados;
- Fator de Ajuste;
- A Contagem de Pontos de Função Ajustados.

#### 6.3.1 Tipo de Contagem

O tipo de contagem é determinado de acordo com os seguintes tipos:

- projeto de desenvolvimento (aplicação nova) ;
- projeto de melhoria (aplicação já existente);
- aplicação (tamanho real)

#### 6.3.2 Escopo da Contagem e a Fronteira da Aplicação

O escopo define se a contagem englobará um ou mais sistemas ou apenas parte dele. Já a fronteira da aplicação delimita o software a ser medido e o mundo externo, ou seja, determina onde uma aplicação termina e outra começa. A Figura 3 exemplifica bem isso.

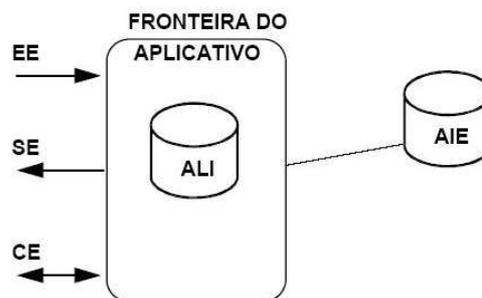


Figura 5 – A fronteira da aplicação com o meio externo [4]

#### 6.3.3 A Contagem de Pontos de Função Não Ajustados

Para se contar os pontos de função não ajustados, é preciso contar primeiro as funções do tipo dados e do tipo transação.

As funções do tipo dados representam as funções fornecidas ao usuário, para atender a suas necessidades de dados internos e externos à aplicação. São classificadas em:

- 1) ALI – Arquivo Lógico Interno
- 2) AIE – Arquivo de Interface Externa

##### 1) Um Arquivo Lógico Interno (ALI) é:

- Um grupo de dados ou informações de controle;
- Identificável pelo usuário;
- Logicamente relacionados;
- Mantido dentro da fronteira da aplicação.

##### 2) Um Arquivo de Interface Externa (AIE) é:

- Um grupo de dados ou Informações de Controle;
- Identificável pelo usuário;
- Logicamente relacionados;
- Referenciado (lido) pela aplicação.

O AIE deve se obrigatoriamente um ALI de outra aplicação. A diferença básica entre eles é que um Arquivo de Interface Externa não é mantido pela aplicação que está sendo contada.

A complexidade de cada função é classificada em baixa, média e alta de acordo com:

- 1) Número de Tipos de Dados (TD) – é campo único, reconhecido pelo usuário, não repetido.
- 2) Número de Tipos de Registros (TR) – é um subgrupo de tipos de dados, reconhecido pelo usuário, componente de um ALI ou AIE.

A complexidade de cada tipo possui um valor em Pontos de Função não-ajustados, fornecido conforme Tabela 2, representada abaixo.

| Tipos de Registros | Tipos de Dados |         |       |  |
|--------------------|----------------|---------|-------|--|
|                    | < 20           | 20 - 50 | > 50  |  |
| 1                  | Baixa          | Baixa   | Média |  |
| 2 - 5              | Baixa          | Média   | Alta  |  |
| > 5                | Média          | Alta    | Alta  |  |

**Tabela 2 – Tabela de complexidade funcional dos ALI e AIE**

Após a determinação da complexidade, deve-se calcular sua contribuição utilizando-se como base a Tabela 3.

| Tipo de Função | Baixa | Média | Alta  |
|----------------|-------|-------|-------|
| <b>ALI</b>     | 7 PF  | 10 PF | 15 PF |
| <b>AIE</b>     | 5 PF  | 7 PF  | 10 F  |

**Tabela 3– Tabela de contribuição dos pontos de função não ajustados das funções do tipo dado**

As funções do tipo Transação representam as funcionalidades de processamento de dados fornecidas pelo sistema ao usuário e são classificadas em:

- Entradas Externas (EE)
- Saídas Externas (SE)
- Consultas Externas (CE).

**Uma Entrada Externa (EE) é:**

Um *processo elementar*<sup>2</sup> que processa dados ou informações de controle recebidos de fora da fronteira da aplicação, cuja principal intenção é manter um ou mais Arquivos Lógicos Internos e/ou modificar o comportamento do sistema.

<sup>2</sup> Processo elementar é a menor atividade percebida pelo usuário que deve ser realizada pelo sistema.

**Uma Saída Externa (SE) é:**

Um processo elementar que envia dados ou informações de controle para fora da fronteira da aplicação, cuja principal intenção é apresentar informação ao usuário por meio de *lógica de processamento*<sup>3</sup> que não seja apenas a recuperação de dados ou informações de controle;

**Uma Consulta Externa (CE) é:**

Um processo elementar que envia dados ou informações de controle para fora da fronteira da aplicação, cuja principal intenção é apresentar informação ao usuário por meio de uma simples recuperação de dados ou informações de controle de um ALI ou AIE. A lógica de processamento não deve conter fórmula matemática ou cálculo nem criar dados derivados.

Regras para determinar se um processo elementar é único

Em todas as funções do tipo transação, para determinar se devemos contar um ou mais processos, pelo menos uma das proposições abaixo devem ser verdadeiras:

- O conjunto de tipos de dados identificado é diferente do identificado em outros processos elementares da aplicação.
- Os ALI e AIE referenciados são diferentes dos referenciados por outros processos elementares da aplicação.
- A lógica de processamento é diferente da executada por outros processos elementares da aplicação. A exceção é apenas quanto à ordenação, ou seja, dois relatórios, cuja diferença seja somente a ordenação de dados, constituem um único processo elementar.

A complexidade de cada tipo é classificada em baixa, média e alta baseado em:

- Número de Arquivos Referenciados (AR);
- Número de Tipos de Dados (TD)

Após determinar a quantidade de arquivos referenciados e de tipos de dados, a classificação em grau de complexidade se dá pelas tabelas 4 e 5 .

| Arquivos Referenciados | Tipos de Dados |       |       |       |
|------------------------|----------------|-------|-------|-------|
|                        |                | < 5   | 5-15  | > 15  |
|                        | < 2            | Baixa | Baixa | Média |
|                        | 2              | Baixa | Média | Alta  |
|                        | > 2            | Média | Alta  | Alta  |

**Tabela 4 – Tabela de complexidade funcional para entradas externas**

<sup>3</sup> Lógica de processamento é o conjunto de procedimentos de conferência, cálculos, referência e acessos, solicitados pelo usuário.

| Arquivos Referenciados | Tipos de Dados |       |       |       |
|------------------------|----------------|-------|-------|-------|
|                        |                | < 6   | 6-19  | > 19  |
|                        | < 2            | Baixa | Baixa | Média |
|                        | 2-3            | Baixa | Média | Alta  |
|                        | > 3            | Média | Alta  | Alta  |

**Tabela 5 – Tabela de complexidade funcional para saídas externas e consultas externas**

### 6.3.4 O Fator de Ajuste

A metodologia de Pontos de Função considera que outros fatores afetam o tamanho funcional de um sistema. Estes fatores estão relacionados com 14 características gerais de sistema (CGS), listadas em seguida [1]:

- |                                     |                              |
|-------------------------------------|------------------------------|
| 1. Comunicação de Dados             | 8. Atualização On-Line       |
| 2. Processamento Distribuído        | 9. Processamento Complexo    |
| 3. Performance                      | 10. Reutilização             |
| 4. Configuração Altamente Utilizada | 11. Facilidade de Instalação |
| 5. Taxa de Transações               | 12. Facilidade de Operação   |
| 6. Entrada de Dados On-Line         | 13. Múltiplos Locais         |
| 7. Eficiência do Usuário Final      | 14. Modificações Facilitadas |

O nível de influência de cada uma das características sobre a aplicação pode variar de 0 a 5, correspondendo a uma escala com limites inferior a superior de nenhuma influência a grande influência.

Determinados os níveis de influência, o fator de ajuste (FA) é calculado pela fórmula:

$$FA = (NI \times 0,01) + 0,65$$

, onde NI = somatório dos níveis de influência das 14 características.

O fator de ajuste pode resultar na variação de  $\pm 35\%$  quando aplicado aos Pontos de Função não ajustados. Observando-se a fórmula, pode-se concluir que cada CGS influencia em até 5% o valor final da contagem, e cada ponto atribuído ao nível de influência afeta o resultado final em 1%.

A soma total para o ajuste de complexidade de processamento é chamada de grau de influência total que é utilizado no cálculo do total de Pontos de Função Ajustado. O processo de ajuste de complexidade é a base de toda correção da medição. O tamanho final do Ponto de Função é calculado através da multiplicação dos Pontos de Função Não- Ajustados pelo Fator de Ajuste, representada pela seguinte fórmula:

$$PF = PF(\text{não ajustado}) \times FA$$

Estes indicadores fornecem o grau de influência das 14 características analisadas, que o fator de ajuste calculado reflete no sistema.

### 6.3.5 A Contagem de Pontos de Função Ajustados

Processo que realiza a correção das possíveis distorções cometidas durante o cálculo dos Pontos de Função Não-Ajustados, aproximando as medidas à situação real.

Existem 3 tipos de contagem:

- Projeto de Desenvolvimento
- Projeto de Melhoria
- Aplicação

Como a empresa, foco da análise já possui um sistema consolidado, os projetos desenvolvidos por ela se enquadram como Projetos de Melhoria, ou seja, são alterações na aplicação para atender aos novos requisitos de negócio do usuário. Não estão contempladas manutenções corretivas e preventivas.

Componentes do cálculo dos Pontos de Função de um Projeto de Melhoria [1]:

- **Funcionalidade da Aplicação Requisitada pelo Usuário para o Projeto:**  
Funções adicionadas, alteradas ou excluídas pelo projeto de melhoria.
- **Funcionalidade de Conversão**  
Funções disponíveis no momento da instalação da aplicação para converter dados ou fornecer outros requisitos de conversão especificados pelo usuário. Após a instalação essas funções são descartadas.
- **Valor do Fator de Ajuste da Aplicação**  
Reflete a complexidade da aplicação e é determinado pela ponderação das características gerais de sistema.

É importante ressaltar que não é necessário saber o número de pontos de função da aplicação para determinar o tamanho do projeto de melhoria.

#### Considerações para Funções do Tipo Dados

Pode-se dizer que uma função do tipo Dado(ALI, AIE) foi alterada quando há modificação em sua estrutura, ou seja, campos acrescentados, excluídos ou terem algum atributo alterado.

#### Considerações para Funções do Tipo Transação

A função é considerada modificada quando há alteração em alguns dos itens:

- Arquivos de dados
- Arquivos referenciados
- Lógica de processamento

O número de pontos de função de um projeto de melhoria é dado pela fórmula:

$$EFP = [(ADD + CHGA + CFP) \times VAGA] + (DEL \times VAFB)$$

**ADD:** número de pontos de função não-ajustados das funções incluídas pelo projeto de melhoria.

**CHGA:** número de pontos de função não-ajustados das funções modificadas pelo

projeto de melhoria.

**CFP:** número de pontos de função não-ajustados adicionadas pela conversão.

**VAVA:** valor do fator de ajuste da aplicação depois do projeto de melhoria.

**DEL:** número de pontos de função não-ajustados das funções excluídas pelo projeto de melhoria.

**VAFB:** valor do fator de ajuste da aplicação antes do projeto de melhoria.

## 7. Aplicação de Pontos de Função como Métrica para Estimativa de um Projeto de Software

Para uma descrição e referência mais detalhada sobre Análise de Pontos de Função ver [1] e [5].

O processo de estimativa de um projeto de software envolve, basicamente, quatro atividades:

- Estimar o tamanho do produto a ser gerado;
- Estimar o esforço empregado na execução do projeto;
- Estimar a duração do projeto;
- Estimar o custo do projeto.

Para obter as respostas aos questionamentos de tempo e custo a partir da aplicação, é necessário decidir a unidade utilizada para medir o tamanho do produto. Após a decisão da unidade de tamanho empregada, pode-se utilizar um modelo de processo de estimativa como o representado na Figura 6.

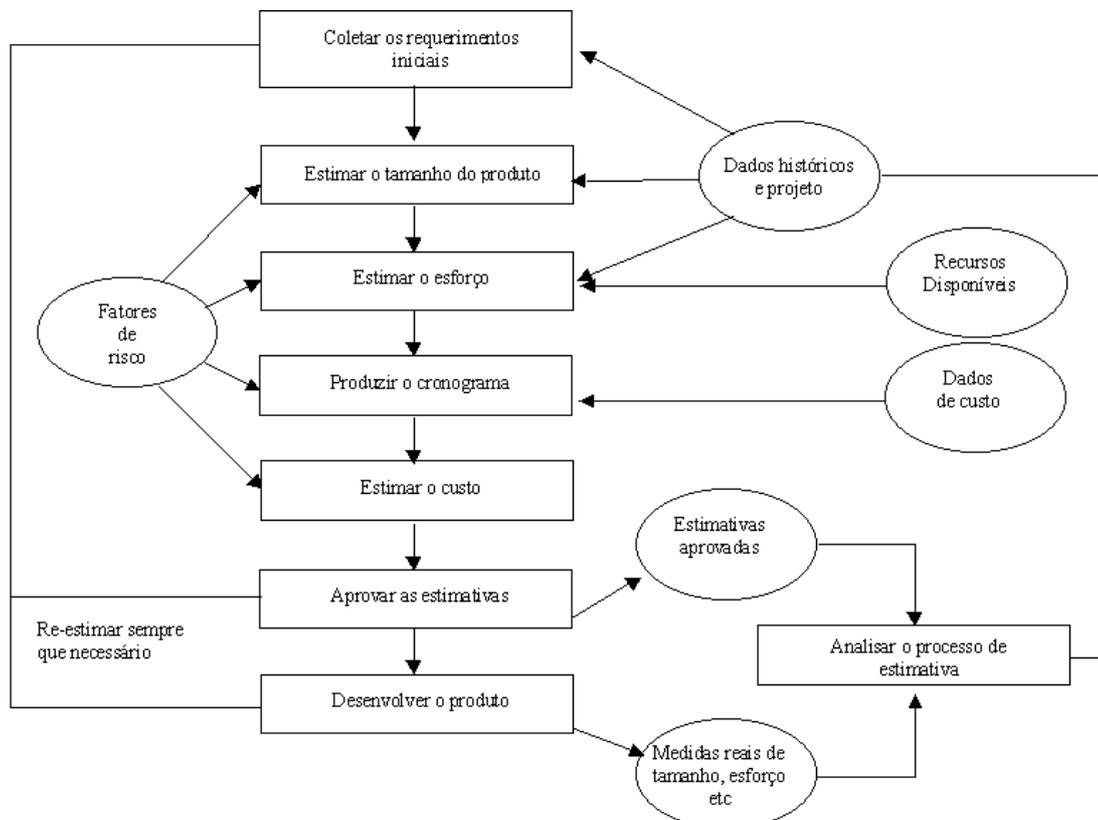


Figura 6 – Processo de estimativa de projeto de software [1]

## **7. Sugestões de Melhorias**

Estão listadas a seguir algumas sugestões com o propósito de melhorar a gerência de projetos:

- Tornar a técnica para estimativa de software mais impessoal possível, incorporando na métrica, a estimativa de tempo para elaboração da especificação técnica de um projeto, desvinculando essa responsabilidade do profissional. Com isso, será possível realizar comparativo entre o planejado e o realizado de todo o projeto, desde a sua especificação até a fase final de teste, diminuindo a influência da experiência profissional individual na estimativa de projetos;
- O problema da inexistência de um plano de teste bem definido não é foco da nossa análise, porém a elaboração de um plano, contemplando quais os requisitos devem ser testados e em que condições, é de suma importância para auxiliar a estimativa da fase de testes. O esforço necessário para essa fase deve ser estimado baseado nos requisitos do sistema e nas condições contidas no plano de testes.
- Implantar a técnica de Análise de Pontos de Função nas áreas da empresa onde ela for aplicável, priorizando as que trabalham com o desenvolvimento de sistemas fazendo uso da ferramenta de desenvolvimento Java e ainda não fazem uso de uma métrica para estimativa de projetos de software.
- Determinar qual a estratégia a ser adotada para contabilizar os pontos de função dos sistemas, criando um inventário de sistemas.

## **8. Conclusões e Trabalhos Futuros**

Estimar projetos de software não é uma tarefa simples, por isso é de fundamental importância o constante acompanhamento das estimativas de projetos realizados para poder avaliar o seu real custo de produção.

Analisando a métrica empregada atualmente, pôde-se concluir que ela não está atendendo às expectativas, pois além de não ser aplicável a todas as áreas de sistemas da empresa, nas áreas onde é aplicável os resultados obtidos apresentam uma disparidade muito grande em relação aos valores sugeridos por ela. E, mesmo quando se faz uso da métrica, a estimativa é fortemente influenciada pelo profissional.

A Análise de Pontos de Função, por ser uma técnica independente de ferramenta que dimensiona software na perspectiva do usuário, apresenta-se como uma ferramenta adequada para auxiliar no processo de gestão de Tecnologia da Informação. Além de estimar esforço e custo de um projeto, a Análise de Pontos de Função é capaz não só de medir a qualidade de um produto como também de fornecer índices de produtividade, taxa de produção e de manutenção de software.

Como trabalhos futuros, pretende-se preparar uma apresentação para o corpo gerencial da Fábrica e da área de Projetos da empresa, abordando os propósitos e benefícios da técnica de Análise de Pontos de Função e, também, realizar um estudo mais aprofundado da métrica para aplicá-la em um projeto interno, viabilizando a sua implantação como a métrica padrão para estimativa de esforço em projetos de software.

## **Referências Bibliográficas**

[1] Vazquez, Carlos Eduardo. Simões, Guilherme Siqueira. Albert, Renato Machado. Análise de Pontos de Função Medição, Estimativas e Gerenciamento de Projetos. 2ª Edição. 2004. Editora Érica.

[2] Rezende, Denis Alcides. Engenharia de Software e Sistemas de Informação. 2ª Edição. 2002. Editora Brasport.

[3] BFPUG – acessível em <http://www.bfpug.com.br> .Último acesso em Agosto/2004.

[4] SERPRO – acessível em <http://www.serpro.gov.br>. Último acesso em Setembro/2004.

[5] Function point counting practices manual - release 4.1.1 . 2003

## Apêndice A . Anexos

### Questionário 1

1. Existe alguma documentação da planilha utilizada pela empresa para medir esforço necessário para desenvolvimento de um projeto de software?
2. Como era medido o esforço para estimar projetos de software na empresa antes de ser elaborada a planilha?
3. Qual foi a motivação para a construção da planilha?
4. Como foram definidos os tópicos e a distribuição dos pesos?
5. Baseou-se em algum método já vivenciado em outra empresa na qual trabalhou para elaboração da planilha?
6. Houve quantas versões da planilha até ser estabelecida uma usual ? Se sim, o que basicamente evoluiu de uma para outra?
7. Há quanto tempo é usada essa metodologia sem sofrer modificação?
8. Já foi feito algum trabalho ao longo desses anos para avaliar a métrica utilizada?

### Questionário 2

1. Você sabe o que significa todos os tópicos contidos na atual planilha e a forma de preenchê-la?
2. Como você tomou conhecimento da maneira correta de preenchê-la, uma vez que não há documentação e orientações de preenchimento?
3. Qual a maior dificuldade ao se usar a planilha?
4. Que pontos você enumera como falhos na métrica atual?