

# **IASP**

**FACULDADE ADVENTISTA DE CIÊNCIAS HUMANAS  
CURSO DE SISTEMAS DE INFORMAÇÃO**



**PROJETO DE TRABALHO DE CONCLUSÃO DE CURSO**

## **Estimativa de investimento do portal Impacto Jovem através da métrica Análise de Ponto de Função**

Alunos

Antonio Laurentino da Silva Filho e William Henrique da Silva

Orientadora

Adriana Aparecida Carnevalli

Hortolândia, SP, de 2009.

Antonio Laurentino da Silva Filho

William Henrique da Silva

**Estimativa de investimento do portal Impacto Jovem através da  
métrica Análise de Ponto de Função**

Trabalho de Conclusão de  
Curso apresentado junto ao Curso de  
Sistemas de Informação da  
Faculdade Adventista de  
Ciências Humanas como requisito  
parcial à obtenção do título de  
Bacharel.

Orientador Prof<sup>a</sup>. Adriana  
Aparecida Carnevalli

Hortolândia, SP  
Dezembro de 2009

Antonio Laurentino Filho da Silva

William Henrique da Silva

**Estimativa de investimento do portal Impacto Jovem através da  
métrica Análise de Ponto de Função**

Trabalho de Conclusão de Curso  
apresentado junto ao Curso de Sistemas de  
Informação da

Faculdade Adventista de Ciências  
Humanas como requisito parcial à obtenção do  
título de Bacharel.

COMISSÃO EXAMINADORA

---

Profº MS. Wagner Luiz Schmidt

---

Profº Adriana Aparecida Carnevalli



## AGRADECIMENTOS

Eu, William Henrique da Silva, agradeço imensamente a Deus por nos dar capacidade para realizar esse trabalho, a minha querida mãe Elaine, meu grandioso pai Orivaldo e minha carinhosa irmã Gabi, pela compreensão e colaboração confiada a mim, a minha amada namorada Gabriela que precisou de uma enorme paciência, a todos os amigos que colaboraram com alguma informação para esse trabalho, em especial ao Fernando, e sem duvida nenhuma a minha orientadora Adriana que ajudou muitíssimo esse trabalho se tornar realidade.

Eu, Antonio Laurentino da Silva Filho, agradeço muito a Deus por permitir e ter nos dado sabedoria e paciência para a realização de mais um projeto em nossa vida, agradeço a minha amada mãe Nazaré, a meu grande pai e amigo Antonio, pela ajuda em todos os momentos no qual me dediquei a esse projeto, a minha amada namorada Rúbia a qual disponibilizou tempo, paciência e compreensão nos momentos de necessidade, a todos os amigos que colaboraram de alguma forma para esse projeto, em especial ao grande amigo Fernando, agradeço com todo carinho a minha orientadora Adriana que foi essencial para que esse projeto pudesse ser realizado.

“Os ventos que às vezes tiram algo que amamos, são os mesmos que trazem algo que aprendemos a amar... Por isso não devemos chorar pelo que nos foi tirado, e sim aprender a amar o que nos foi dado, pois tudo aquilo que é realmente nosso, nunca se vai para sempre...”

Bob Marley

## RESUMO

Este trabalho apresenta a metodologia de Análise de Pontos por Função, utilizada para a contagem dos pontos em um projeto de Gerenciamento de Conteúdo Web. A obtenção da contagem de pontos de função se dá a partir de especificações funcionais fornecidas no início do projeto de desenvolvimento do produto de software para que se tenha a possibilidade de estabelecer estimativas de custo (esforço) e tempo mais precisas, beneficiando significativamente o planejamento do projeto. Nesse sentido são apresentadas algumas técnicas de estimativas de esforço. Para todo esse processo, será utilizada uma ferramenta específica que aperfeiçoa todas as etapas da contagem dos pontos de função. Também são apresentadas e comparadas algumas métricas de software com ênfase na métrica de pontos de função.

Palavra – chave: Análise de Pontos de Função, contagem, métricas

## ABSTRACT

This college paper, presents the methodology of the Analyzes of Points for Function, used for counting of the points in a project of WEB administration content. The obtaining of the counting of function points it happens at the starting from functional specifications supplied, in the beginning of the project of the software product development, so that able to be had of establishing estimates (effort) and accurate time, benefitting significantly. the project planning. In that direction are presented some effort on estimation techniques. For entire process, a specific tool will be used for improve all stages of counting of the function points. They are also presented and compared some metric of software with emphasis in the metric of function points.

Key Words: Analysis of Function Points, counting, metric

<b>1</b>	<b>INTRODUÇÃO</b> .....	<b>1</b>
<b>2</b>	<b>OBJETIVO</b> .....	<b>2</b>
2.1	OBJETIVO ESPECÍFICO.....	2
<b>3</b>	<b>JUSTIFICATIVA</b> .....	<b>2</b>
<b>4</b>	<b>PROBLEMATIZAÇÃO</b> .....	<b>3</b>
<b>5</b>	<b>HIPÓTESE</b> .....	<b>3</b>
<b>6</b>	<b>METODOLOGIA</b> .....	<b>3</b>
<b>7</b>	<b>IFPUG</b> .....	<b>4</b>
<b>8</b>	<b>INICIO DA APF NO BRASIL</b> .....	<b>4</b>
<b>9</b>	<b>MÉTRICAS DE SOFTWARE</b> .....	<b>5</b>
9.1	OUTRAS MÉTRICAS FUNCIONAIS UTILIZADAS.....	5
9.2	MÉTRICAS MENOS UTILIZADAS .....	5
9.3	COCOMO II.....	6
9.4	A MÉTRICA DE PONTOS DE FUNÇÃO.....	8
<b>10</b>	<b>O PROCESSO DE CONTAGEM DE PONTO DE FUNÇÃO</b> .....	<b>9</b>
10.1	DETERMINAR O TIPO DE CONTAGEM.....	11
10.2	CONTAGEM DE UM PROJETO EM DESENVOLVIMENTO .....	11
10.3	CONTAGEM DE UM PROJETO EM MELHORIA.....	12
10.4	CONTAGEM DE UMA APLICAÇÃO .....	12
10.5	FRONTEIRA DA APLICAÇÃO E O ESCOPO DA CONTAGEM.....	13
10.6	CONTAR FUNÇÕES DO TIPO DADO .....	14
10.6.1	<i>Definição de termos utilizados</i> .....	15
10.6.2	<i>Definição de um Arquivo Lógico Interno (ALI)</i> .....	17
10.6.3	<i>Definição de Arquivo de Interface Externa (AIE)</i> .....	18
10.6.4	<i>Determinação da Complexidade</i> .....	18
10.6.5	<i>Definição de Tipo de Dado (TD)</i> .....	19
10.6.6	<i>Definição do Tipo de Registro (TR)</i> .....	21
10.6.7	<i>Determinação da Contribuição</i> .....	21
10.7	CONTAR FUNÇÕES DO TIPO TRANSAÇÃO .....	22
10.7.1	<i>Definição de Entrada Externa (EE)</i> .....	22
10.7.2	<i>Definição de Saída Externa (SE)</i> .....	23
10.7.3	<i>Definição de Consulta Externa (CE)</i> .....	24

10.7.4	<i>Determinação da Complexidade</i> .....	25
10.7.5	<i>Definição de Arquivo Referenciado (AR)</i> .....	26
10.7.6	<i>Determinação da Contribuição</i> .....	27
10.8	DETERMINAR OS PONTOS DE FUNÇÃO NÃO AJUSTADOS .....	27
10.9	DETERMINAR O FATOR DE AJUSTE .....	29
10.10	CALCULAR OS PONTOS DE FUNÇÃO AJUSTADOS .....	31
10.10.1	<i>Cálculo de Pontos de Função para Projeto de desenvolvimento</i> .....	31
10.10.2	<i>Cálculo de Pontos de Função de Projeto de Melhoria</i> .....	33
10.10.3	<i>Cálculo de uma Aplicação</i> .....	34
<b>11</b>	<b>VANTAGENS E DESVANTAGENS DO PONTO DE FUNÇÃO</b> .....	<b>35</b>
<b>12</b>	<b>DEFINIÇÃO DE CONTAGEM DE UM SISTEMA</b> .....	<b>37</b>
12.1	CONTAGEM SISTEMA DE GERENCIAMENTO DE CONTEÚDO WEB .....	37
12.1.1	<i>Sistema “Portal Impacto Jovem”</i> .....	37
12.1.2	<i>Funcionalidades do Sistema</i> .....	37
<b>13</b>	<b>SOFTWARE APFPLUS</b> .....	<b>39</b>
13.1	CONTAGEM INDICATIVA .....	49
13.2	CONTAGEM ESTIMADA .....	50
13.3	ESTIMATIVA DE INVESTIMENTO PARA O SISTEMA PORTAL “IMPACTO JOVEM” .....	54
<b>14</b>	<b>CONSIDERAÇÕES FINAIS</b> .....	<b>54</b>
<b>15</b>	<b>REFERÊNCIAS BIBLIOGRÁFICA</b> .....	<b>ERRO! INDICADOR NÃO DEFINIDO.</b>

## Lista de Figuras

Figura 1 - Processo de Contagem de Pontos de Função.....	9
Figura 2 - Componentes de Contagem .....	10
Figura 3 - Tipo de Contagem.....	11
Figura 4 - Relacionamento entre os tipos de contagem.....	12
Figura 5 - Fronteira e Escopo da Contagem .....	13
Figura 6 - Contar Funções Tipo Dado .....	15
Figura 7 - Tabela de complexidade funcional dos ALI e AIE.....	19
Figura 8 - Tabela de contribuição dos pontos de função não ajustados das funções do tipo dado .....	21
Figura 9 - Contar funções do tipo transação .....	22
Figura 10 - Funções do tipo transação .....	22
Figura 11 - Tabela de complexidade para entradas externas (EEs).....	25
Figura 12 - Tabela de complexidade para saídas externas (SEs) e consultas externas (CEs).....	25
Figura 13 - Tabela de contribuição dos pontos de função não ajustados das Funções do tipo transação.....	27
Figura 14 - Determinar os pontos de função não ajustados.....	27
Figura 15 - Cálculo de pontos de função não ajustados .....	28
Figura 16 - Exemplo de cálculo de ponto de função não ajustados.....	29
Figura 17 - Determinar fator de ajuste.....	29
Figura 18 - Cálculo dos pontos de função ajustados .....	31
Figura 19 Apresentação APFplus .....	39
Figura 20 Cadastro da Aplicação.....	40
Figura 21 Cadastro do Projeto .....	40
Figura 22 Registro de Fase .....	41
Figura 23 Registro de Ambiente.....	42
Figura 24 Registro de Recursos .....	43
Figura 25 Registro de Equipe .....	44
Figura 26 Registro de Usuários .....	44
Figura 27 Registro de Grupo .....	46

Figura 28 Registro de funções do tipo dado .....	47
Figura 29 Registro de funções do tipo transação.....	48
Figura 30 Registro dos Itens de Influência .....	49
Figura 31 Contagem Indicativa .....	50
Figura 32 Contagem Estimada.....	51
Figura 33 Estimativa de Custos .....	51
Figura 34 Definição de Ambiente.....	52
Figura 35 Definição de Recurso .....	52

# 1 Introdução

A APF teve seu início em 1979, resultado de um projeto do pesquisador da IBM Allan Albrecht, que era definir uma técnica de estimativa para esforço no desenvolvimento de software, independente da linguagem de programação que fosse utilizada no projeto. Apesar de ter surgido na IBM, o resultado desse projeto está disponível à comunidade de software desde 1984, onde se tornou reconhecida mundialmente e passou a ser de domínio público da comunidade e indústria de software em todo o mundo.

Após ele analisar vários programas, traçou os fundamentos dessa técnica, que visa medir uma aplicação através das funções desempenhadas no projeto para se mostrar ao usuário final, pois ela é baseada na visão do mesmo. A APF mede o que é o projeto, e não como será, ou foi desenvolvida, ela se divide basicamente em dois tipos de funções: dados e transações.

Um dos principais conceitos da teoria de APF é que as funções devem ser contadas a partir da perspectiva do usuário e não do analista desenvolvedor.

Até 1984, a técnica era usada basicamente para identificar o tamanho de um sistema e auxiliar na estimativa do esforço de desenvolvimento. Agora a utilização tem se dado através da implantação de programas de métricas para melhorar estimativas, gerenciar a qualidade, monitorar a produtividade, além de controlar indicadores financeiros, ou seja, melhorar todo o processo de desenvolvimento e manutenção de sistemas.

Essa métrica foi originalmente projetada para sistemas de aplicações comerciais e, por isso, necessitou evoluir devido ao rápido aumento da complexidade algorítmica das aplicações. Nasceram então os chamados Pontos Característicos ou Pontos de Particularidade (*Feature Points*). Deve-se notar que os Pontos de particularidade e os Pontos por Função representam a mesma coisa, a funcionalidade ou a utilidade de um software.

A APF é uma técnica que oferece a medição de funcionalidades de um software a partir da visão do cliente. O Ponto de Função (PF) é a unidade que tem o objetivo de se medir um projeto independente da tecnologia em que o software foi construído. Portanto o processo de medição (também chamado contagem de pontos de função) é baseado em uma avaliação padronizada dos requisitos lógicos do usuário. Em 1993 Gustav Karner propôs em um trabalho acadêmico a metodologia dos Pontos por Caso de Uso (baseado na Análise de Pontos de Função) com o objetivo de estimar recursos para projetos de software orientados a objeto desenvolvidos utilizando o processo *Objectory*.

O método Feature Point foi proposto por Capers Jones em 1986, o mesmo ano em que foi fundada a IFPUG (*International Function Point Users Group*), e antes da primeira versão do Manual de Práticas de Contagem - publicado em 1988. O objetivo do método era o de medir com maior precisão softwares com grande complexidade algorítmica. O objetivo era acrescentar à APF mais uma dimensão além dos dados e transações - os algoritmos. No entanto a falta de uma ciência para classificar algoritmos dificultou a aplicação do método. Resumindo, embora o método *Feature Points* ainda seja eventualmente citado em alguns trabalhos acadêmicos, o mercado não aderiu a esta proposta.

## 2 Objetivo

Hoje em dia, encontramos um mercado cada vez mais competitivo e exigente. E é por essa razão que diversas empresas buscam se destacar com recursos e ferramentas que se diferenciam da concorrência. Por esse motivo é que a Análise de Pontos de Função (APF) foi escolhida, pelo simples fato de possuir técnicas embasadas em cálculos matemáticos e ser internacionalmente um padrão de medida de software.

### 2.1 Objetivo Específico

Utilizar a métrica de APF para estimar o tempo, custo e o tamanho do projeto em questão. Analisando a importância da APF na identificação do tamanho funcional de um projeto de software como uma ferramenta de contagem e armazenamento dos pontos de função. Com essa ferramenta podemos dimensionar o tamanho de um software, podendo ser aplicado em qualquer fase do projeto levando em consideração apenas a funcionalidade geral que o produto proporciona ao usuário final. Pois com ele podemos medir o que foi requisitado e recebido pelo usuário, independente da tecnologia utilizada para a implementação. Sendo assim, uma forma de estimar o tamanho do software apenas com a visão do usuário.

## 3 Justificativa

As empresas estão sempre em busca de um diferencial a mais, isso ocorre para que elas possam estar proporcionando maior qualidade em seus produtos a cada dia. Em meio a tanto avanço e tanta competitividade surgem produtos que se destacam. O software é medido por diversas razões, pois é necessário indicar a qualidade de um produto, avaliando a sua

produtividade e das pessoas que o desenvolvem. Avaliar os benefícios (em termos de produtividade e qualidade) derivados de novos métodos e ferramentas de softwares formando uma *baseline* para estimativas e ajudar a justificar os pedidos de novas ferramentas ou treinamento adicional.

## 4 Problematização

Geralmente, as empresas constroem software sem dimensioná-lo, assim sendo, para estimar seu tempo de construção e seu valor, fazem por comparação com outras construções de software e experiências anteriores. Isso acarreta, muitas vezes, erros de estimativas de preço e prazo que prejudicam as empresas fornecedoras de software. Ou seja, “Não se pode gerenciar o que não se pode medir” (Tom de Marco), e com isso não é possível estimar o prazo e o custo desse sistema para que possa ser comunicado ao cliente.

## 5 Hipótese

Dentro das métricas disponíveis, a APF foi escolhida pelo fato de possuir técnicas embasadas em cálculos matemáticos e ser internacionalmente um padrão de medida de software. Utilizar a métrica de APF para estimar o tempo, custo e o tamanho do projeto em questão, tem-se por hipótese que a estimativa de tempo e custo será mais próxima do real.

## 6 Metodologia

Para a realização deste projeto, foi utilizada a metodologia de pesquisa Ação, que consiste em uma pesquisa com base empírica que é realizada em associação com uma ação ou resolução de um problema coletivo onde os participantes então envolvidos de modo cooperativo ou participativo.

## 7 IFPUG

O IPFUG teve seu início em 1986 com um grupo de pessoas 12 pessoas que se reuniram para a discussão sobre o uso da FPA (*Function Point Analysis* ou Análise de Pontos de Função), e a troca de suas experiências. Com isso nascia o *International Function Point Users Group* (Grupo Internacional de Usuários de Pontos de Função). Na segunda reunião o grupo já contava com a participação de 45 profissionais de 25 empresas.

Com a possibilidade de utilizar a APF para benefícios gerenciais o número de usuários aumentou rapidamente e fez com que o IFPUG criasse um grupo específico para cuidar da padronização da APF e do seu controle de evolução. Em 1990 foi aprovada a versão 3.0 do manual de práticas de contagem. Seguindo para as versões 3.1, 3.2, 3.3, 3.4 e em 1994 foi criada a versão 4.0, e hoje em dia o IPFUG utiliza a versão 4.3, onde já contém todas as práticas e normas atualizadas.

A primeira empresa brasileira a se filiar ao IFPUG foi a *Unisys*, no ano de 1989, onde mais tarde seria responsável pela realização do primeiro evento relacionado ao assunto realizado no Brasil.

## 8 Início da APF no Brasil

Segundo Braga (1996), a introdução da APF no Brasil se deu em novembro de 1991, onde a empresa *Unisys* realizou o 1º Encontro Nacional de Usuários de Pontos de Função (ENUPF) na cidade de Santos, e contou com a presença de 28 profissionais de informática de 22 empresas distintas e palestras de especialistas do assunto na época. O ENUPF ocorreu ainda em outras ocasiões e foi acontecendo em diferentes cidades com Salvador, São Paulo e Rio de Janeiro. Em cada encontro o evento era enriquecido com profissionais de grandes empresas nacionais e internacionais, e membros do IFPUG (*International Function Point Users Group*) e do *Counting Practices Committee* (Comitê do IFPUG que cuida da evolução da técnica de FPA), onde esse comitê foi representado pelo Dr. Eberhard Rudolph da Universidade de Bremerhaven na Alemanha. O ENUPF teve seis edições e a cada edição o volume de pessoas envolvidas e empresas participantes sempre cresceram.

Em 1998 foi criado o BFPUG (*Brazilian Function Point Users Group*) que foi a evolução de um grupo de usuários de APF na cidade do Rio de Janeiro. Com a criação do BFPUG se associaram centenas de pessoas como estudantes, desenvolvedores, consultores e

gerentes de sistemas. Hoje em dia o BFPUG promove a troca de informações e experiências entre seus membros, além de realizar palestras e a aplicação do exame de certificação do IFPUG aqui no Brasil.

## 9 Métricas de Software

### 9.1 Outras métricas funcionais utilizadas

Existem alguns outros métodos de medição funcional, os quais são:

NESMA – (*Netherlands Software Metrics Users Association*), foi criado na Holanda, onde tem o seu próprio manual de contagem, porém foi feito em 1990, baseado no manual do IPFUG, tendo somente algumas diretrizes diferentes, fazendo com que a NESMA possua uma abordagem distinta do IPFUG.

A NESMA, originalmente chamado de NEFPUG (*Netherlands Function Point Users Group*), foi fundada em 1989, sendo o primeiro grupo a ser criado após o IFPUG. Mark II – Criado por Charles Symons na década de 80 e foi durante anos propriedade de uma coUKSnsultoria, onde atualmente já é um método de domínio público mantido pela MA (*United Kingdom Software Metrics Association*), mas esse método é restrito ao Reino Unido.

COSMIC-FFP – Foi criada em 1997 por um grupo de pesquisadores da Universidade de Quebec, onde desenvolveram um novo método de medição funcional para sistemas de tempo real, denominado *Full Function Points* (FFP). Em 1998 um grupo de especialistas em medição de software constituiu o *COSMIC* (*Common Software Measurement International Consortium*) com o objetivo de desenvolver um novo método de medição de tamanho funcional baseado nas melhores características dos métodos existentes e que incorporasse novas idéias. Ele foi proposto em 2000, batizado de *COSMIC-FFP*, na prática foi um refinamento do método FFP. Ainda não é uma técnica tão disseminada no mundo quanto à do IFPUG, porém observa-se que muita pesquisa está sendo realizada sobre este método. Pode-se dizer que o mercado acompanha atento seu passo.

### 9.2 Métricas menos utilizadas

Além desses métodos funcionais, podemos citar outras métricas de estimativas de softwares: Contagem de Linhas de Código Fonte (LOC): Esta técnica é a mais antiga forma existente para medir um fator que é considerado crítico em um programa de mensuração: o

tamanho de um sistema. A técnica tem como vantagem a simplicidade, porém está totalmente ligada à linguagem de programação utilizada e com a forma de implementação, tendo praticamente nenhum significado para o usuário final.

Além disso, ela só pode ser aplicada após a codificação dos programas de um produto, tornando-se inviável para estimativas de projetos.

**COCOMO:** O método de COCOMO (*Constructive Cost Model*) tem sua teoria baseada no número de instruções fonte (número de linhas de código) e fundamentada em fórmulas matemáticas, podendo ser aplicado nas diversas fases do ciclo de desenvolvimento.

O COCOMO está dividido em três modelos:

**Básico:** Modelo que pode ser aplicado a grande maioria de projetos de software de pequeno e médio porte. As estimativas fornecidas por este método são limitadas, pois desconsidera alguns atributos importantes pertinentes ao projeto, como as restrições de hardware e a qualificação e experiência das pessoas envolvidas no projeto.

**Intermediário:** Este modelo é uma versão melhorada do modelo básico, abrangendo as limitações contidas no outro modelo e também suas estimativas são mais precisas.

**Detalhado:** É uma técnica que permite realizar estimativas em nível de módulo, subsistema e sistema, inclusive mostrando os atributos de custo em cada fase do projeto.

### 9.3 Cocomo II

O Modelo COCOMO II teve como precursor o COCOMO. Devido à idade dos projetos que embasaram o modelo, assim como sua incapacidade de lidar com ciclos de vida iterativos e com a utilização de componentes 3 *Commercial-Off-The-Shelf* (COTS).

O COCOMO II aplicado ao RUP estima o esforço, prazo e equipe média para as fases de *Elaboration* e *Construction*.

O CII produz estimativas para a região do ciclo de vida do RUP situada entre os marcos LCO e IOC. Estabelecido o escopo de estimativa do CII, vejamos como o modelo prevê o Esforço.

O CII define o Esforço (PM = *Persons-Month*), em pessoas-mês, como

$PM = A * (Size^E) * Produtório (i=1 \text{ até } n, EM(i))$ , onde

$E = B + 0,001 * Somatório (j=1 \text{ até } 5, SF(j))$

A notação “*Produtório (m=1 até n, F(m))*” indica o produto de todos os elementos  $F(m)$ ,  $m$  variando de 1 a  $n$ , inclusive. A notação “*Somatório*” tem utilização análoga. O circunflexo (“^”) denota a operação de exponenciação.

$A$  = Constante que deve ser calibrada para o ambiente, a partir de dados históricos.

$Size$  = Tamanho do software em LOC. Caso o tamanho seja dado em Pontos de Função, estimar a quantidade de LOC a partir de tabelas de *backfiring* [8], que permitem converter Pontos de Função para LOC.

$EM(i)$  = *Effort Multipliers* = Até 17 fatores que irão adequar o modelo ao projeto específico.

A quantidade de fatores utilizados dependerá do momento no qual será feita a estimativa. No início do projeto e até o final da *Elaboration* utiliza-se o *Early Design Model*, com apenas 7 multiplicadores. Após a definição da arquitetura deve ser utilizado o *Post-Architecture Model*, com 16 multiplicadores. Os multiplicadores referem-se a fatores relativos ao produto, à plataforma, ao pessoal e ao projeto. Um fator freqüentemente considerado em separado é aquele que diz respeito à compressão do prazo de desenvolvimento, ou seja, a exigência de que o projeto seja concluído antes do que seria o prazo normal. O significado de cada um dos fatores e a orientação para a sua determinação pode ser encontrado em:

$E$  = Expoente do Esforço, dado pela fórmula acima.

$B$  = Constante que deve ser calibrada a partir de dados históricos.

$SF(j)$  = *Scale Factors* = Cinco fatores que irão ajustar o expoente do Esforço, de modo a adequar a fórmula a um projeto específico. Deve-se tomar cuidado ao ajustar estes fatores, uma vez que o impacto dos mesmos sobre as estimativas é exponencial. Por sua vez, o Prazo (TDEV), em meses, é calculado como

$TDEV = C * PM^F$ , onde

$F = D + (0,2 * 0,01) * \text{Somatório } (j=1 \text{ até } 5, SF(j))$ , e  $C$  e  $D$  são constantes que devem ser calibradas a partir de dados históricos. Quando cabível, deverá ser aplicada compressão de prazo sobre a fórmula acima.

A Equipe Média é obtida através da divisão do Esforço pelo Prazo. O CII considera um mês equivalente a 152 horas de trabalho, já excluídos fins de semana e feriados. Este valor pode ser modificado para um ambiente específico.

### ***Calibração do Modelo***

O Modelo COCOMO II foi originalmente calibrado com dados de 161 projetos. Os mesmos foram selecionados dentre mais de 2000 projetos candidatos. Para cada um dos 161 projetos escolhidos foram realizadas entrevistas e visitas, a fim de garantir a consistência das definições e suposições do modelo. O modelo nominal vem calibrado para esses projetos, cuja natureza pode diferir daquele que se deseja estimar.

Embora o CII possa ser executado com os parâmetros nominais, sua correta utilização pressupõe a calibração para o ambiente-alvo. Na ausência de dados históricos disponíveis para o ambiente-alvo em questão, devem ser selecionados projetos equivalentes para efetuar a calibração.

Os dados históricos selecionados devem ser validados antes de sua utilização, alimentando-os no software escolhido, calculando os coeficientes calibrados e, posteriormente, verificando se a diferença percentual (*estimado – real*) / *estimado* encontra-se compatível com o nível de erro pretendido para as estimativas. Devido ao seu impacto exponencial, não é recomendável calibrar os coeficientes B e D nas equações acima quando houver menos de 10 projetos disponíveis para a calibração.

## 9.4 A Métrica de Pontos de Função

A APF é uma técnica de mensuração de software utilizada por profissionais da área de sistemas e também por empresas de todos os portes e até mesmo outros segmentos. No início, o foco era para estimativas para projetos e posteriormente passou a ser aplicada também em contratos de desenvolvimentos de software, a fim de medir e gerenciar esses projetos. Ela é ainda um padrão normalizado mundialmente pela ISO através da norma ISSO/IEC 20926.

A utilização da APF no Brasil pode ser conferida em contratos e licitações ligados ao governo, pois a mensuração de projetos tornou a sua utilização uma realidade.

Diferentemente de outras métricas, ela avalia o tamanho funcional do projeto, ou seja, ela analisa o que o software realmente faz, quantificando as funções contidas no software em termos bem significativos, a partir da visão do usuário. Ela se relaciona diretamente aos requisitos do negócio que o software pretende tratar, pois ela pode ser aplicada de imediato a um amplo espectro de ambientes e ao longo da vida de um projeto em desenvolvimento, desde a definição dos requisitos até a fase utilização operacional.

A técnica resume-se em duas funções principais:

- Função de dados;
- Função de transação;

**Pode ser dividida em:**

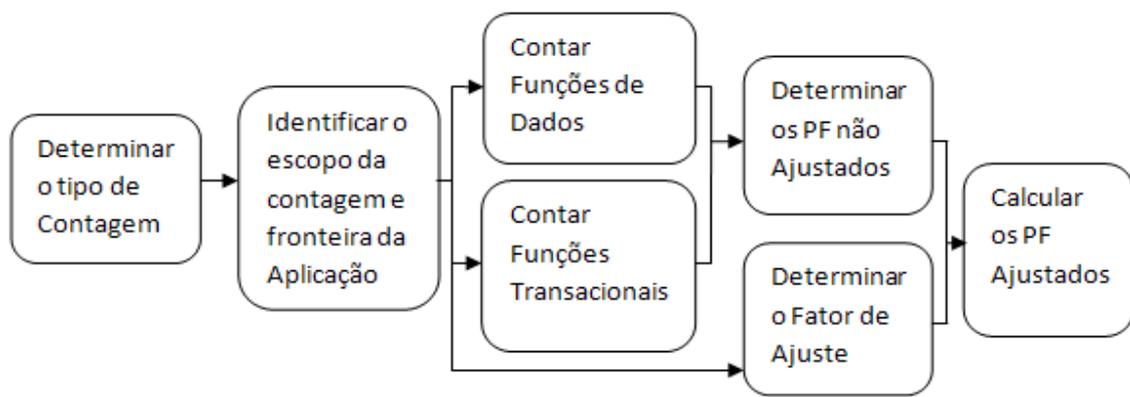
- Entrada Externa (EE);
- Saída Externa (SE);
- Consulta Externa (CE);

**Essas funções podem ser de dois tipos:**

- Arquivos Lógicos Internos (ALI);
- Arquivos de Interface Externa (AIE);

## 10 O Processo de contagem de Ponto de Função

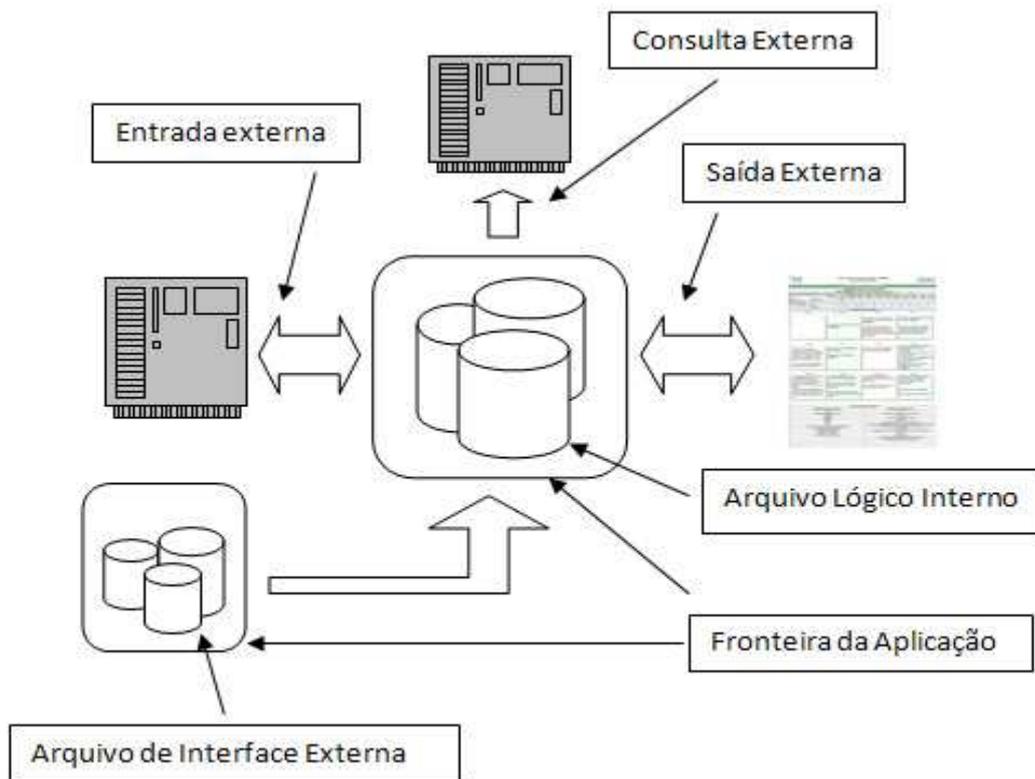
Com base nos elementos e nos processos, a contagem por ponto de função pode ser descrita resumidamente. O digrama a seguir mostra, resumidamente, uma visão geral do que seria o processo de contagem por ponto de função.



**Figura 1 - Processo de Contagem de Pontos de Função.**

Esta técnica mostra algumas abstrações que são fundamentais e necessárias na elaboração e determinação do tamanho funcional do projeto. Estes conceitos representam e fornece funcionalidades, armazenamento e, processo de dados e, ainda, utiliza os mesmos como referência na identificação de componentes. As fronteiras de aplicação e tipos de funcionalidade que são chamadas assim para fins de APF, são apresentadas suas abstrações no diagrama abaixo. (APF, 2006).

Segundo diagrama do livro APF (componentes de contagem):



**Figura 2 - Componentes de Contagem**

A partir desse ponto iremos abordar todas as etapas necessárias para a contagem de um projeto, como mostrados na figura anterior.

## 10.1 Determinar o Tipo de Contagem

Como foi dito anteriormente, a partir desta desse ponto iremos explicar cada etepa no processo de contagem de um projeto, e esta por sinal é responsável por definir o tipo de contagem que será utilizada para se medir o software.

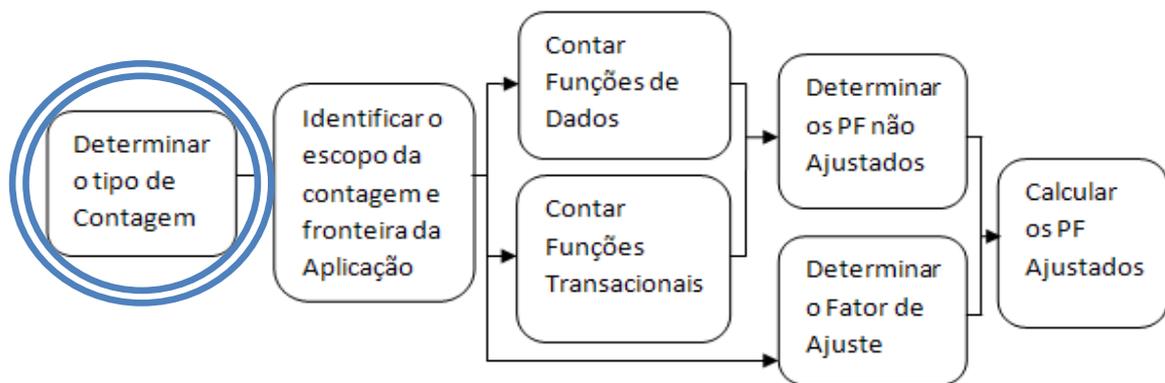


Figura 3 - Tipo de Contagem

Em APF existem 3 tipos de contagem que iremos mostrar a seguir:

- Contagem de um projeto em desenvolvimento
- Contagem de um projeto em melhoria
- Contagem de uma aplicação (ou baseline)

## 10.2 Contagem de um Projeto em Desenvolvimento

No livro APF (2008), a quantidade de pontos de função de um projeto em desenvolvimento, também é usada para medir as funcionalidades fornecidas ao usuário final de um determinado software em sua primeira instalação. Essa contagem por sua vez abrange inclusive as possíveis conversões de dados que serão necessários para a implantação do software.

É importante observar que uma contagem realizada antes do término do projeto, é simplesmente uma estimativa da fase final do mesmo, com isso ao decorrer do desenvolvimento de um projeto naturalmente surgirão funcionalidades que não haviam sido identificadas inicialmente.

### 10.3 Contagem de um Projeto em Melhoria

Segundo Vazquez, Simões e Albert (2008), no projeto de melhoria as funções adicionadas, modificadas ou até mesmo excluídas do sistema pelo projeto são medidas pelo número de pontos de função e também possíveis conversões de dados.

Após o projeto de melhoria estar concluído, deve-se calcular novamente o número de pontos de função para que seja possível refletir as alterações realizadas no sistema.

### 10.4 Contagem de uma Aplicação

De acordo com o livro APF (2008), as funcionalidades fornecidas aos usuários por uma aplicação instalada são medidas pelo seu próprio número de ponto de função, que por sua vez também são chamados de postos de função instalados ou *baseline*. Esse número proporciona uma medida atual da funcionalidade obtida pelo usuário da aplicação.

Sendo inicializado ao final da contagem do número de pontos de função do projeto de desenvolvimento e será atualizado ao terminar todo o projeto de melhoria, que por sua vez irá alterar a funcionalidade da aplicação.

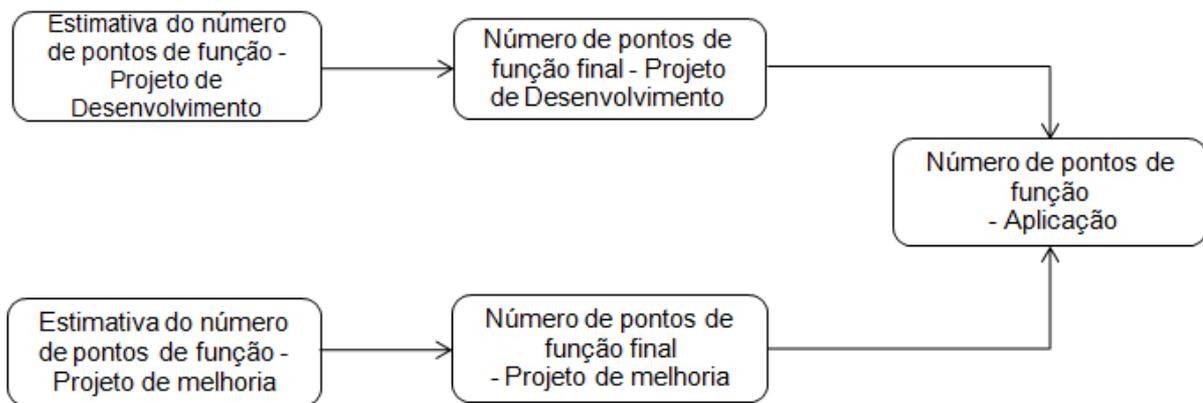


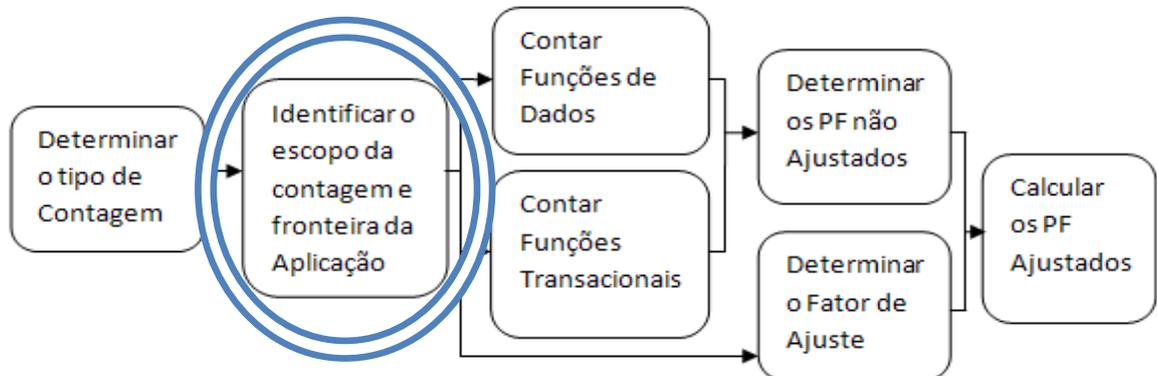
Figura 4 - Relacionamento entre os tipos de contagem

O termo “aplicação” possui vários significados na área de tecnologia da informação. Segundo o IFPUG (2008), “uma aplicação é um conjunto coeso de dados e procedimentos automatizados que suportam um objetivo de negócio, podendo consistir de um ou mais componentes, módulos ou subsistemas”.

É com base na visão do usuário e de acordo com as considerações de negócio que a aplicação deve ser definida e não baseado em questões técnicas como arquitetura do software ou plataforma tecnológica.

## 10.5 Fronteira da Aplicação e o Escopo da Contagem

Logo após a definição do tipo de contagem, o passo seguinte do processo é identificar a fronteira da aplicação e também o escopo da contagem.



**Figura 5 - Fronteira e Escopo da Contagem**

Conforme o livro APF (2008), a fronteira de aplicação é a interface conceitual que delimita o software que será medido e o mundo exterior (seus usuários). A definição de fronteira é um passo extremamente importante no processo de contagem, pois se a definição de fronteira não estiver bem definida o risco de que todo o trabalho de contagem posteriormente feito seja inválido, acabará sendo muito grande.

Usando o mundo real para exemplificar de uma maneira bem prática, podemos imaginar que a fronteira da aplicação seria a cerca de uma fazenda, da mesma forma que seria impossível medir o tamanho da fazenda se não houvesse uma cerca delimitando-a, também não é possível especificar o tamanho funcional de um sistema sem antes estabelecer a fronteira que o delimita.

Se houver mais de uma aplicação incluída no escopo da contagem, várias fronteiras deverão ser identificadas.

De acordo com o livro APF (2008), o IFPUG especifica algumas regras para determinar a fronteira da aplicação, que serão listadas abaixo:

- O ponto de vista do usuário deve ser usado para determinar a fronteira, mantendo o foco no que pode ser entendido ou descrito pelo mesmo;
- A fronteira da aplicação não deve estar baseada em considerações tecnológicas, mas sim na separação conforme fora estabelecido na pelos processos do negócio;

- Em projetos de manutenção, a fronteira inicial deve estar em conformidade com fronteira já estabelecida para a aplicação ou, aplicações que estão sendo modificadas.

Para tornar mais fácil a identificação da fronteira de aplicação, segue algumas dicas (BRAGA,1996):

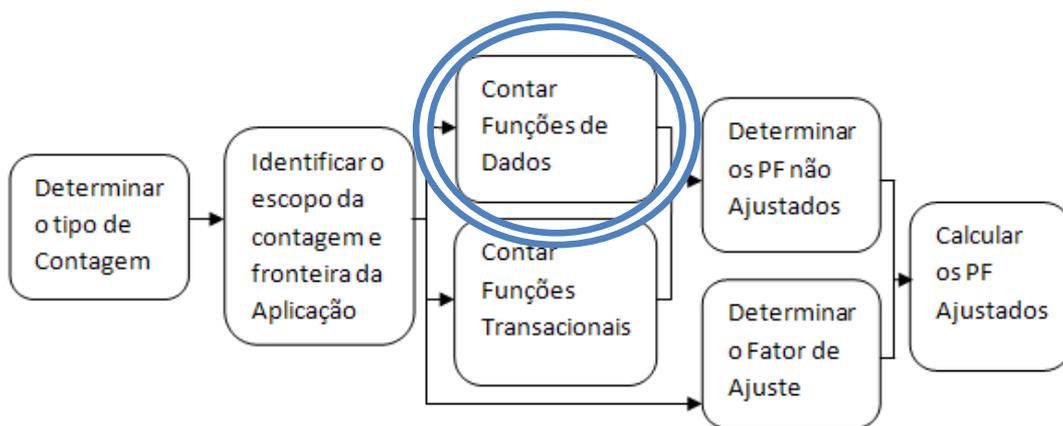
- Utilize o diagrama de fluxo de dados do sistema ou as especificações e, desenhe as fronteiras para destacar quais são as funções internas e externas da aplicação;
- Verifique como todos os grupos de dados são mantidos;
- Para auxiliar na determinação das fronteiras, utilize a própria definição da contagem para que assim fique mais fácil identificá-la;
- Fique atento e observe qual é o procedimento utilizado na manutenção de cada grupo de dados;
- Observe e identifique funções de caráter empresarial através da associação de propriedade de certos tipos de objetos ao sistema;

Comparar os critérios utilizados em outras métricas como duração, esforço, custo e defeito. As fronteiras que são utilizadas na contagem de pontos de função devem ser as utilizadas em outros tipos de medidas.

## **10.6 Contar Funções do Tipo Dado**

Nesta etapa do processo de contagem do projeto, as funções do tipo dados serão definidas a seguir, a partir de agora iremos explicar todos os processos e regras desta contagem.

Funções do tipo dados representam as funcionalidades fornecidas para suprir suas necessidades de dados internos e dados externos a aplicação. E os mesmos são classificados em Arquivos Lógicos Internos (ALI) e Arquivos de Interface Externos (AIE). (APF, 2007)



**Figura 6 - Contar Funções Tipo Dado**

O termo *arquivo* neste contexto se refere a um grupo de dados reconhecidos pelo usuário e, logicamente relacionados, e não como é comum ter seu significado ligado à área de processamento de dados. (Braga, 1996).

Explicaremos a seguir todas as definições de Arquivo Lógico Interno (ALI) e Arquivo de Interface Externa (AIE), e também termos mais utilizados, procedimentos de contagem e regras associadas com este tipo de função.

### 10.6.1 Definição de termos utilizados

Para que fique claro, é de extrema importância que se obtenha um entendimento perfeito das definições dos termos que serão utilizados.

#### Informação de Controle

É o dado utilizado pela aplicação para garantir a aderência da mesma aos requisitos das funções de negócio estabelecidas pelo usuário (BRAGA, 1996).

#### Identificados pelo Usuário

Nas definições de ALI e AIE este termo se refere ao requerimento especificado pelo usuário (BRAGA, 1996).

### **Processo Elementar**

É a menor atividade com algum significado para os negócios do usuário, pois o processo elementar deve ser auto-suficiente para que assim possa deixar em um estado consistente a aplicação que está sendo contada (BRAGA, 1996).

Para um PE ser único em relação aos demais eles devem satisfazer a umas das seguintes regras:

1. Os Arquivos lógicos referenciados serem diferentes dos demais processos elementares;
2. Os itens de dados que atravessam a fronteira da aplicação serem diferentes. Observe que segundo a Análise de Pontos de Função, o rótulo "Segunda Via" não é um item de dado e não deve ser contado;
3. A lógica de processamento ser diferente.

Obs: Existe uma tabela com os 13 tipos de lógicas de processamento previstos pela APF e excepcionalmente a lógica de processamento "Dados são reclassificados ou rearrumados" não determina a unicidade de um processo elementar. Isso significa que se tivermos uma consulta de clientes ordenada pelo nome do cliente e outra consulta idêntica ordenada de data de nascimento, embora tenha essa LP distinta entre as consultas devemos contá-las uma única vez.

### **Manutenido**

Nos processos elementares de aplicação, o termo mantido (mantido) é utilizado nas definições de ALI e AIE no que se refere à habilidade de incluir, excluir e alterar dados (LEMOS, 2006).

### **Fatores de Complexidade**

O número de Arquivo Lógico Interno (ALI) e de Arquivo de Interface externa (AIE) determina a contribuição das funções de tipos dados para a contagem dos pontos de Função do tipo Bruto. (LEMOS 2006).

### **10.6.2 Definição de um Arquivo Lógico Interno (ALI)**

Segundo o livro APF (2008) um Arquivo lógico Interno (ALI) é;

1. Um grupo de dados ou informações de controle;
2. Identificável pelo usuário;
3. Logicamente relacionado;
4. Mantido na fronteira da aplicação.

Um ALI tem como principal intenção armazenar dados mantidos (inseridos, alterados ou excluídos) através de um ou mais **processos elementares** da aplicação que está sendo controlada.

#### **Exemplos de ALI**

Veremos a seguir alguns exemplos de ALI, segundo (BRAGA, 1996):

- Cadastro de Produtos
- Cadastro de Funcionários
- Cadastro de Cliente
- Arquivo de controle de acesso à aplicação
- Arquivo de dependente

#### **Não-Exemplos de ALI**

Veremos agora exemplos que não são ALIs, de acordo com (BRAGA, 1996):

- Arquivos de classificação
- Arquivo temporário
- Arquivo de trabalho
- Arquivos de índices alternativos
- Arquivos inseridos por motivos tecnológicos.

### 10.6.3 Definição de Arquivo de Interface Externa (AIE)

Segundo o livro APF (2008) um Arquivo de Interface Externa é:

1. Um grupo de dados ou informações de controle;
2. Identificável pelo usuário;
3. Logicamente relacionado;
4. Referenciado (lido) pela aplicação.

Um AIE tem como principal intenção armazenar dados **referenciados** por meio de um ou mais **processos elementares** dentro da fronteira da aplicação que está sendo contada. Sendo assim, o AIE deve ser obrigatoriamente um ALI de outra aplicação.

#### Exemplos de AIE

Segundo o livro APF (2008) são exemplos de Arquivos de Interface Externa:

- Dados de referência externos utilizados pela aplicação

#### Não-Exemplos de AIE

Veremos agora exemplos que não são AIEs, segundo o livro APF (2008);

- Arquivos de movimento recebidos de outra aplicação para manter um ALI (exemplos: arquivos de remessa e de retorno). No entanto os processos de carga e de geração desses arquivos podem ser funções do tipo transação. O arquivo de movimento é simplesmente um relatório gerado em formato de arquivo texto;
- Dados mantidos pela aplicação e utilizados por outra aplicação;
- Dados formatados e processados para uso de outras aplicações.

#### Diferença entre ALI e AIE

A diferença entre um Arquivo Lógico Interno e um Arquivo de Interface Externa, é simplesmente que o AIE não é mantido pela aplicação sendo contada ele está conceitualmente fora da fronteira da aplicação, já o ALI é mantido dentro da aplicação que está sendo contada. (VASQUEZ; SIMOES; ALBERT, 2008)

### 10.6.4 Determinação da Complexidade

A complexidade funcional é classificada em baixa, média e alta. Cada AIE e ALI deve ser classificado de acordo com sua complexidade, baseado em:

- Número de Tipos de Dados (TD);

- Número de Tipos de Registros (TR).

Depois de especificadas as quantidades de TD e TR, a complexidade é classificada de acordo com a seguinte tabela:

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

Figura 7 - Tabela de complexidade funcional dos ALI e AIE

Com isso, um ALI que contém 45 tipos de dados e um tipo de registro é classificado com complexidade baixa e outro AIE contendo 55 tipos de dados e um tipo de registro é classificado como complexidade média.

### 10.6.5 Definição de Tipo de Dado (TD)

Segundo o livro APF (2008), um tipo de dado é um campo único, reconhecido pelo usuário, não repetido.

#### Regras de Contagem de Tipos de Dados

A seguir veremos algumas regras de contagem que devem ser consideradas, para que um determinado campo seja contado ou classificado como um tipo de dado.

1. Deve ser contado um tipo de dado para cada campo que o usuário reconhecer e não repetido, mantido ou recuperado de um ALI ou AIE através da execução de um processo elementar. Veja a seguir Alguns exemplos:
  - No caso do agendamento de um recebimento, a data de vencimento poderia estar armazenada em múltiplos campos (dia, mês e ano), porém continuaria sendo contada como apenas um único tipo de dado.
  - Uma imagem anterior e posterior a uma atualização de um grupo de dez campos mantidos para o propósito de uma auditoria é contada como tipo de

dado da imagem anterior (todos os dez campos) e um tipo de dado para a imagem posterior (todos os dez campos), totalizando assim dois tipos de dados.

- Também devem ser contados como tipos de dados, os campos em um ALI que são calculados e armazenados.
  - Se forem reconhecidos pelo usuário, os campos do tipo *timestamps* devem ser contados como tipo de dados.
  - Arquivos com várias ocorrências do mesmo campo: valor janeiro, valor fevereiro, ..., e valor dezembro; são contados como dois tipos de dados: um tipo de dado para o determinado mês e outro tipo de dado para um determinado valor.
2. Se duas aplicações mantêm ou referenciam o mesmo ALI/AIE deve ser contado apenas os campos utilizados pela aplicação em análise. Abaixo será mostrado alguns exemplos:
- Uma aplicação irá referenciar os campos de um arquivo como CPF e nome, onde deverão ser contados dois tipos de dados referentes ao arquivo. A outra irá referenciar os campos do mesmo arquivo, como: nome, logradouro, cidade, estado e CEP, onde deverão ser contados cinco tipos de dados para o mesmo arquivo.
  - Em uma aplicação é fundamental identificar cada parte do endereço do cliente, como logradouro, cidade, estado e CEP. Estas mesmas informações são relevantes apenas no conjunto. Para o endereço devem ser contados quatro tipos de dados na primeira aplicação, o mesmo já não ocorre com a outra que se utiliza apenas de um tipo de dado para o endereço.
3. Para se estabelecer um relacionamento com outro arquivo lógico (AIE ou ALI), é necessário contar um tipo de dado para cada campo que o usuário solicitar.
- Usando como exemplo uma aplicação de controle de ponto as informações de entrada e saída são mantidas no ALI que iremos chamar de Apontamento. Fazem parte da identificação da pessoa as informações do Apontamento; serve para estabelecer uma ligação entre o ALI Apontamento com o AIE Pessoa. Dessa maneira serão contados quatro tipos de dados no ALI que chamamos de Apontamento, que são: identificação da pessoa (chave estrangeira), data, horário de entrada e horário de saída.

- Se acaso a chave estrangeira estiver sendo composta por vários campos, todos esses campos deverão ser contados como tipos de dados.

### 10.6.6 Definição do Tipo de Registro (TR)

Segundo o livro APF (2008), um tipo de registro é um subgrupo de dados, reconhecido pelo usuário, componente de um Arquivo Lógico Interno ou Arquivo de Interface Externa.

Existem dois tipos de subgrupos:

- **Opcionais:** são aqueles em que o usuário pode optar por informar ou não no processo elementar que cria ou adiciona dados ao arquivo.
- **Obrigatórios:** O usuário requer que sejam sempre utilizados pelo processo elementar que cria ou adiciona dados ao arquivo.

### Regras de Contagem de Tipos de Registro

Veremos a seguir algumas regras que devem ser utilizadas para determinar o número de tipos de registro de um ALI ou de um AIE.

- Para cada subgrupo deve ser contado um tipo de registro, seja ele obrigatório ou opcional de um ALI ou AIE.
- Será contado o próprio ALI ou AIE como um tipo de registro, se não houver nenhum subgrupo.

### 10.6.7 Determinação da Contribuição

Segundo o livro APF (2008), deve-se calcular a contribuição após a determinação da complexidade de cada arquivo identificado.

Veremos a seguir a tabela que é utilizada para efetuar esse cálculo.

<b>Tipos de Função</b>	<b>Baixa</b>	<b>Média</b>	<b>Alta</b>
Arquivo Lógico Interno	7 PF	10 PF	15 PF
Arquivo de Interface Externa	5 PF	7 PF	10 PF

**Figura 8 - Tabela de contribuição dos pontos de função não ajustados das funções do tipo dado**

Exemplo: No caso de um ALI obter 15 pontos de função não ajustados, será classificado com complexidade alta, também será classificado com complexidade alta um AIE que contribuir com 10 pontos de função não ajustados.

## 10.7 Contar Funções do Tipo Transação

Nesta parte do processo de contagem, será explicado o processo e as regras de contagem relacionadas e as definições do tipo transação.

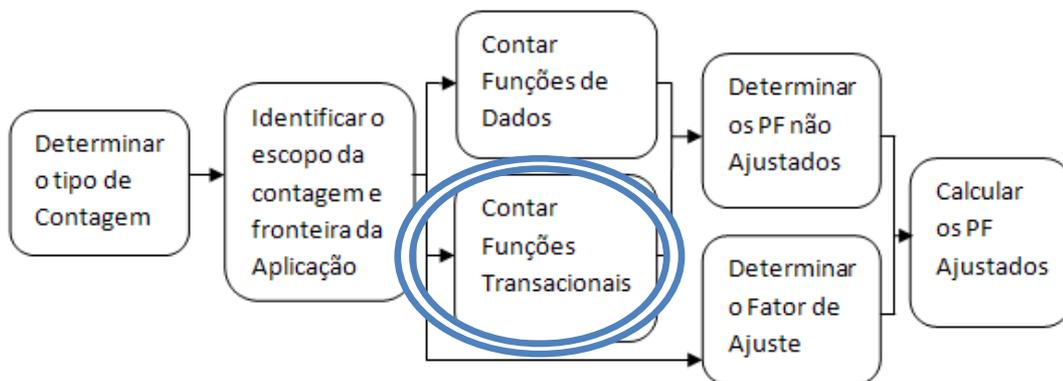


Figura 9 - Contar funções do tipo transação

E também dentro do tipo de transação explicaremos e definiremos as Entradas Externas (EE), Saídas Externas (SE), e Consultas Externas (CE).

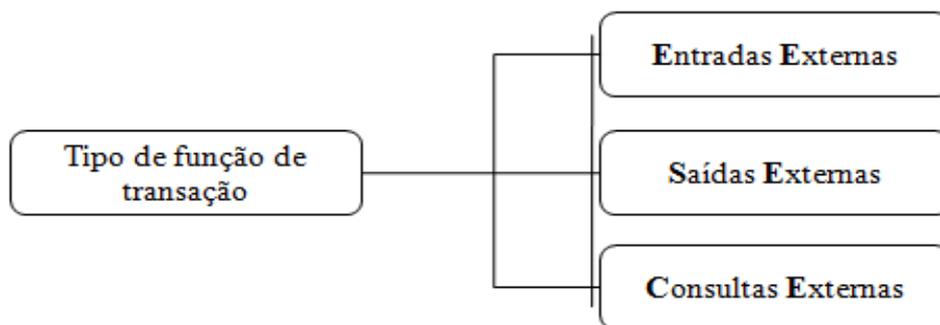


Figura 10 - Funções do tipo transação

### 10.7.1 Definição de Entrada Externa (EE)

De acordo com o livro APF (2008), uma Entrada Externa é:

1. Um **processo elementar**;
2. Faz o processamento de dados ou **informações de controle** recebido de fora da fronteira da aplicação;

3. Tem como objetivo **manter** (incluir, alterar ou excluir dados de) um ou até mais ALIs e ou **modificar o comportamento do sistema**.

### **Exemplos de Entrada Externa (EE)**

Veremos a seguir alguns exemplos de entradas externas, de acordo com o livro APF (2008):

- Dados externos que são utilizados na manutenção de um ALI, que são recebidos através de transação;
- Arquivos que contribuem com três entradas externas como, por exemplo, uma janela que permite incluir, excluir e alterar um determinado registro.
- Processamento em lotes de atualizações de bases cadastrais a partir de arquivos de movimentos.

### **Não-Exemplos de Entrada Externa (EE)**

Segundo APF (2008), não são exemplos de entradas externas:

- Menus;
- Telas de login;
- Telas de filtro de relatório e consultas.

## **10.7.2 Definição de Saída Externa (SE)**

No livro APF (2008), uma Saída Externa é:

1. Um **processo elementar**;
2. Enviam dados ou informações de controle para fora da fronteira da aplicação;
3. Tem como principal intenção por meio de lógica de processamento apresentar ao usuário informação. Essa lógica de processamento deve conter obrigatoriamente pelo menos uma fórmula matemática ou cálculo.

### **Exemplos de Saída Externa (SE)**

Segundo BRAGA (1996), os itens a seguir são exemplos de saída externa:

- Arquivos que também são atualizados por relatórios;
- Informações em formato geográfico;
- Tela de login (com criptografia);

- Gerado para outra aplicação arquivo de movimento (ex: arquivo de retorno ou remessa);
- Relatório com totalização de dados;
- Apresentação de dados derivados ou consultas com cálculos;

### **Não-Exemplos de Saída Externa (SE)**

Não são exemplos de saída externa:

- *Drop-downs* (apenas apresentação e recuperação de dados);
- Telas de *help*;
- Relatórios e consultas sem nenhum totalizador, que não tem dados derivados, não atualizam arquivos ou modificam o comportamento do sistema. (APF, 2008)

### **10.7.3 Definição de Consulta Externa (CE)**

Segundo o livro APF (2008), uma Consulta Externa é:

1. Um **processo elementar**;
2. Envia dado ou informação de controle para fora da fronteira da aplicação;
3. Tem como principal objetivo mostrar informações ao usuário através de uma recuperação de dados ou informações de controles de ALIs e ou AIE. Sua lógica de processamento não deve haver nenhuma fórmula ou cálculo matemático.

### **Exemplos de Consulta Externa (CE)**

São exemplos de consulta externa, de acordo com BRAGA (1996):

- Consulta ao cadastro de clientes;
- Telas de *logon* com funções de segurança;
- Telas de *menu* se solicitado ao usuário;
- Tela de *menu* que só provê seleção de tela e consulta de dados.

### **Não-Exemplos de Consulta Externa (CE)**

Não são exemplos de consultas externas, BRAGA (1996).

- Menus de navegação;
- Telas de consulta;
- Tela de menu que só provê seleção de telas;
- Tela de logon que provê segurança da aplicação.

#### 10.7.4 Determinação da Complexidade

No livro APF (2008), cada Entrada Externa (EE), Saída Externa (SE) e Consulta Externa (CE) deve ser classificada de acordo com sua complexidade funcional que são compostas por três níveis de medida (baixa, média, alta) com base em:

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

A seguir veremos duas tabelas que classificarão a complexidade dos arquivos referenciados e de tipos de dados, isso ocorre logo após a determinação das quantidades dos mesmos. APF (2008)

Arquivos Referenciados (ARs)	Tipos de Dados (TDs)		
		< 5	5 - 15
< 2	Baixa	Baixa	<b>Média</b>
2	Baixa	<b>Média</b>	Alta
> 2	<b>Média</b>	Alta	Alta

Figura 11 - Tabela de complexidade para entradas externas (EEs)

Arquivos Referenciados (ARs)	Tipos de Dados (TDs)		
		< 6	6 - 19
< 2	Baixa	Baixa	<b>Média</b>
2 - 3	Baixa	<b>Média</b>	Alta
> 3	<b>Média</b>	Alta	Alta

Figura 12 - Tabela de complexidade para saídas externas (SEs) e consultas externas (CEs)

Assim, uma EE que contém dezesseis tipos de dados e dois arquivos referenciados é classificada como complexidade alta. E uma SE com os mesmo valores da EE é classificada como complexidade média. Uma CE contendo dezenove tipos de dados e um arquivo referenciado é de classificação baixa.

### **10.7.5 Definição de Arquivo Referenciado (AR)**

Arquivo referenciado segundo o livro APF (2008) é:

- Arquivo Lógico Interno lido ou mantido pela função do tipo transação;
- Arquivo de Interface Externa lida ou mantida pela função do tipo transação.

### **Regras de Contagem para Arquivo referenciado**

Veremos a seguir regras válidas para a contagem de um arquivo referenciado, sendo que as duas primeiras por tratarem de atualização de arquivos, não serão aplicadas para consultas externas. APF (2008)

- Será contado um arquivo referenciado para cada ALI mantido;
- Deve ser contado um arquivo de referenciado para cada ALI que seja mantido ou até mesmo lido;
- Conte um arquivo referenciado para cada ALI ou AIE que for lido durante o processamento. O ALI e AIE não devem ser contados mais de uma vez, também não serão contados arquivos que não se classificam como ALI ou AIE e não contar o mesmo arquivo mais de uma vez mesmo que a transação faça várias leituras nele.

### 10.7.6 Determinação da Contribuição

A tabela a seguir será usada para calcular a contribuição após determinada a complexidade das funções do tipo transação:

Tipos de Função	Baixa	Média	Alta
Entrada Externa	3 PF	4 PF	6 PF
Saída Externa	4 PF	5 PF	7 PF
Consulta Externa	3 PF	4 PF	6 PF

Figura 13 - Tabela de contribuição dos pontos de função não ajustados das Funções do tipo transação

**Exemplo:** Um EE de complexidade alta contribui com 6 pontos de função não ajustados, uma SE também de complexidade alta contribui com 7 pontos de função não ajustados e uma CE também de complexidade alta contribui com 6 pontos de função não ajustados. APF (2008).

### 10.8 Determinar os Pontos de Função Não Ajustados

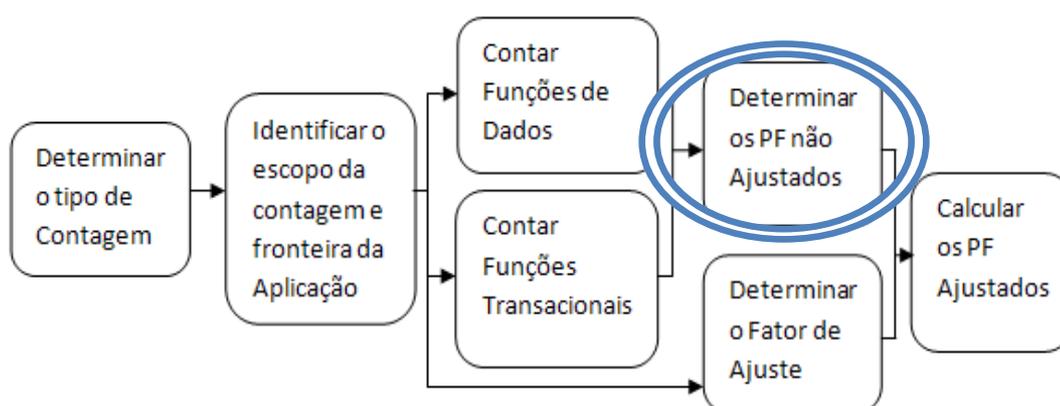


Figura 14 - Determinar os pontos de função não ajustados

Neste capítulo veremos como determinar a quantidade de pontos de função não ajustados.

Tendo as definições de cada função e sua complexidade relativa em posse, devemos calcular os pontos de função não ajustados, através da seguinte representação:

Tipo Função	Complexidade Funcional	Total por Complexidade	Total por Tipo
<b>Arquivo lógico Interno (ALI)</b>	Simple	x 7 =	
	Média	x 10 =	
	Complexa	x 15 =	
<b>Arquivo de Interface Externa (AIE)</b>	Simple	x 5 =	
	Média	x 7 =	
	Complexa	x 10 =	
<b>Entrada Externa (EE)</b>	Simple	x 3 =	
	Média	x 4 =	
	Complexa	x 6 =	
<b>Saída Externa (SE)</b>	Simple	x 4 =	
	Média	x 5 =	
	Complexa	x 7 =	
<b>Consulta Externa (CE)</b>	Simple	x 3 =	
	Média	x 4 =	
	Complexa	x 6 =	

Figura 15 - Cálculo de pontos de função não ajustados

**Exemplo de cálculo dos pontos de função brutos:**

Suponhamos um sistema com as seguintes características:

- ALIs: 10 Simple, 5 Médio, 2 Complexo
- AIEs: 3 Simple, 2 Médio, 0 Complexo
- EEs: 13 Simple, 7 Médio, 5 Complexa
- SEs: 8 Simple, 5 Médio, 3 Complexa
- CEs: 9 Simple, 5 Médio, 4 Complexa

Fazendo as seguintes adições aos valores respectivos na tabela acima teríamos:

Tipo Função	Complexidade Funcional	Total por Complexidade	Total por Tipo	Soma por tipo função
Arquivo lógico Interno (ALI)	10	x 7 =	70	150
	5	x 10 =	50	
	2	x 15 =	30	
Arquivo de Interface Externa (AIE)	3	x 5 =	15	29
	2	x 7 =	14	
	0	x 10 =	0	
Entrada Externa (EE)	13	x 3 =	39	97
	7	x 4 =	28	
	5	x 6 =	30	
Saída Externa (SE)	8	x 4 =	32	78
	5	x 5 =	25	
	3	x 7 =	21	
Consulta Externa (CE)	9	x 3 =	27	71
	5	x 4 =	20	
	4	x 6 =	24	
<b>Total de pontos de função não ajustados</b>				<b>425</b>

Figura 16 - Exemplo de cálculo de ponto de função não ajustados

## 10.9 Determinar o Fator de Ajuste

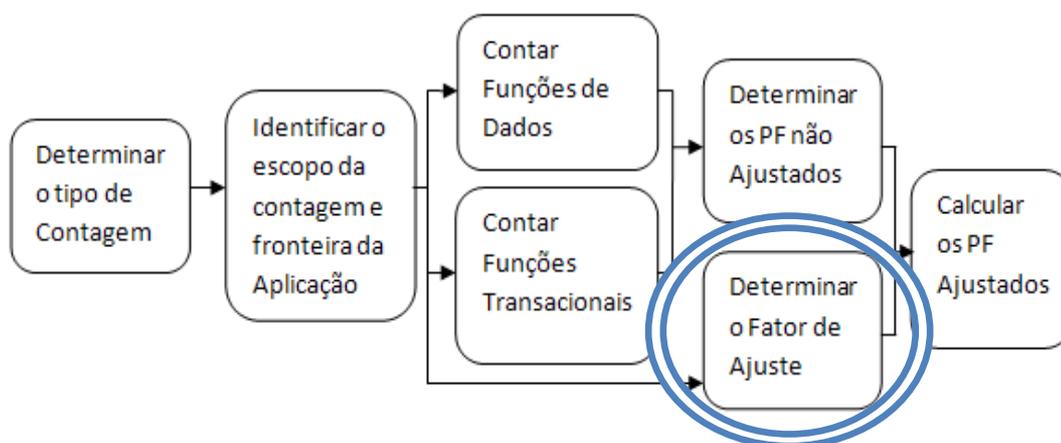


Figura 17 - Determinar fator de ajuste

Segundo Mecenas (2009) e Braga (1996), o Fator de Ajuste está baseado em 14 características gerais do sistema que classificam as funcionalidades gerais da aplicação que está sendo contada. Cada característica tem descrições associadas que ajudam a determinar o nível de influência da característica.

O nível de influência de cada característica pode variar em escala de 0 a 5 ou de sem influência até forte influência.

Essas características estão resumidas no Fator de Ajuste, as características gerais do sistema que avaliam a complexidade da aplicação como um todo são:

- Comunicação de Dados
- Processamento Distribuído
- Performance
- Configuração Intensamente Utilizada (ou Configuração Altamente Utilizada)
- Volume de Transações
- Entrada de Dados *on-line*
- Eficiência do Usuário Final
- Atualizações *on-line*
- Processamento Complexo (ou Complexidade de Processamento)
- Reusabilidade (ou Reutilização)
- Facilidade de Instalação (ou Facilidade de Implantação)
- Facilidade de Operação
- Múltiplos Locais
- Facilidade de Mudança (ou Modificação Facilitada)

A fórmula para obtenção do Fator de Ajuste é a seguinte:

$VAF = (TDI * 0,01) + 0,65$  onde:

VAF = Valor do Fator de Ajuste

TDI = Total do nível de influência

O valor de VAF varia 0,65 a 1,35.

## 10.10 Calcular os Pontos de Função Ajustados

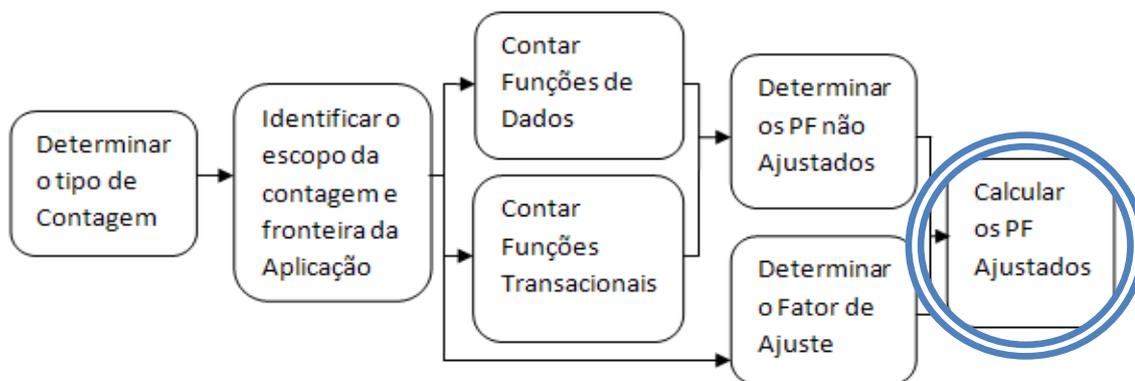


Figura 18 - Cálculo dos pontos de função ajustados

De acordo com APF (2008), para se calcular os pontos de função ajustados, é necessário definir os três tipos de contagem possíveis:

**Projeto de desenvolvimento:** é utilizado em projetos de desenvolvimento onde pode envolver, ou não, conversão de cadastros já existentes.

**Projeto de melhoria:** utiliza-se apenas em casos de manutenções evolutivas na aplicação, ou seja, existe um sistema já desenvolvido e desejamos quantificar um projeto de manutenção.

**Aplicação:** é utilizado para calcular o tamanho de um sistema após a execução de um projeto de manutenção.

### 10.10.1 Cálculo de Pontos de Função para Projeto de desenvolvimento

São três os componentes da funcionalidade necessários para o cálculo dos PF de um projeto de desenvolvimento:

#### Funcionalidade da aplicação

São funções utilizadas após a instalação do software para satisfazer as necessidades correntes do negócio do usuário.

#### Funcionalidade da conversão

Consiste das funções providas para converter dados ou necessidades específicas de conversão especificadas pelo usuário.

### **Fator de ajuste da aplicação**

O fator de ajuste é determinado através das 14 características gerais do sistema.

### **Fórmula de cálculo do Projeto de Desenvolvimento**

A seguinte fórmula é usada para calcular os pontos de função de um projeto de desenvolvimento.

$$\underline{DFP = (UFP + CFP) * VFA}$$

Sendo que:

- DFP é o número de PF de desenvolvimento.
- UFP é o número de PF brutos.
- CFP é o número de PF adicionados pelo processo de conversão.
- VFA é o valor do fator de ajuste.

### **Exemplo Lemos (2006)**

Dada a contagem de Pontos de Função Brutos (UFP) do projeto de desenvolvimento de 107 e a contribuição da conversão em Pontos de Função Brutos (CFP) de 7, considerando um fator de ajuste (VAF) de 1.26, encontrar o número de PFs do projeto aplicando a seguinte fórmula:

$$DFP = (UFP + CFP) * VAF$$

$$DFP = (107 + 7) * 1.26$$

$$DFP = 144 \text{ Pontos de Função}$$

### **10.10.2 Cálculo de Pontos de Função de Projeto de Melhoria**

Segundo o livro APF (2008) são três os componentes da funcionalidade necessários para o cálculo dos PF de um projeto de manutenção:

#### **Funcionalidade da aplicação**

A funcionalidade da aplicação é composta pelos pontos de função que são adicionados através da funcionalidade que é no sistema; pontos de função que são contados, pois representam funcionalidade que sofreu alguma alteração e por pontos de função que são contados, pois representam a funcionalidade que foi excluída durante a manutenção.

#### **Funcionalidade de conversão**

Refletem os PF entregues através da funcionalidade de conversão requerida pelo usuário.

#### **Fator de ajuste da aplicação**

São considerados dois fatores:

1. Fator de ajuste antes do início da manutenção da aplicação.
2. Fator de ajuste após o término da manutenção da aplicação

#### **Formula de cálculo**

$$\underline{EFP = [(ADD + CHGA + CFP) * VAFA] + (DEL * VAFB)}$$

Sendo que:

- EFP: São PF do projeto em melhoria.
- ADD: São PF não ajustados que foram incluídos na aplicação pelo projeto de melhoria. Mostram as funções que foram adicionadas à aplicação.
- CHGA: são os PF que foram adicionados no processo de conversão.
- CFP: Número de PF não ajustados das funções adicionadas pela conversão.
- VAFA: Valor do fator de ajuste da aplicação depois do projeto de melhoria.
- DEL: São PF que foram excluídos da aplicação pelo projeto de manutenção. Mostram as funções que foram excluídas da aplicação.
- VAFB: É o fator de ajuste da aplicação antes do projeto de manutenção.

É importante enfatizar que os Pontos de Função excluídos (DEL) são somados no projeto de manutenção, já que efetivamente há um esforço no processo de sua exclusão.

**Exemplo Lemos (2006)**

O exemplo é composto por:

- Funcionalidade incluída: 7 PF
- Funcionalidade alterada: 14 PF
- Funcionalidade excluída: 3 PF

Considerando que o fator de ajuste da aplicação permaneceu com o mesmo índice de 1,26, teríamos:

$$EFP = [(ADD + CHGA + CFP) * VAFA ] + (DEL * VAFB)$$

$$EFP = [(7 + 14 + 0) * 1,26] + (3 * 1,26)$$

$$EFP = 30,24 \text{ PF}$$

### 10.10.3 Cálculo de uma Aplicação

Quando um projeto de manutenção é finalizado, as contagens de Pontos de Função da aplicação devem que ser atualizadas para refletir as manutenções feitas. A funcionalidade da aplicação pode ser modificada de varias formas:

- A adição de novas funções aumenta o tamanho da aplicação;
- A modificação de funções pode aumentar, diminuir ou não alterar o tamanho de uma aplicação;
- O fato de excluir funções diminui o tamanho do sistema;
- A alteração do fator de ajuste pode aumentar, diminuir ou não modificar o tamanho da aplicação.

A funcionalidade de conversão não afeta a contagem dos PF de uma aplicação.

Após um projeto de manutenção utiliza-se da seguinte forma para se calcular os seus Pontos de Função:

$$\underline{APF = [(UFPB + ADD + CHGA) - (CHGB + DEL)] * VAFA}$$

- APF: Pontos de Função Brutos da aplicação antes do projeto de manutenção.
- UFPB: Pontos de Função Brutos da aplicação antes do projeto de manutenção.
- ADD: Pontos de Função Brutos que foram adicionados pelo projeto de manutenção.
- CHGA: Pontos de Função Brutos correspondentes às funções que sofreram algum tipo de alteração durante o projeto de manutenção. Este número reflete as funções depois da manutenção.
- CHGB: Pontos de Função Brutos correspondentes às funções que sofreram alteração durante o projeto de manutenção. Estes números refletem as funções antes da manutenção.
- DEL: Pontos de Função não ajustados que correspondem às funções que foram excluídas da aplicação pelo projeto de melhoria.
- VAFA: Fator de ajuste da aplicação verificado depois do projeto de melhoria.

#### **Exemplo Lemos (2006)**

A seguir será mostrada a contagem do tamanho do aplicativo após o projeto de manutenção:

$$APF = [(UFPB + ADD + CHGA) - (CHGB + DEL)] * VAFA$$

$$APF = [(107 + 7 + 14) - (14 + 3)] * 1,26$$

$$APF = 139,86$$

$$APF = 140 \text{ Pontos de Função}$$

## 11 Vantagens e Desvantagens do Ponto de Função

Após descrever e analisar a métrica de contagem por pontos de função encontramos vantagens e desvantagens em sua utilização na estimativa de um determinado sistema. Descrevemos as seguintes:

### **Vantagens**

BRAGA (1996) apud LEMOS (2006) relata as vantagens do uso da técnica de Análise por Pontos de Função como sendo as seguintes: Independência de tecnologia; fornece uma unidade padrão de medida de software; fornece um veículo para estimativas de

desenvolvimento de sistemas; é simples o bastante para minimizar o esforço gasto com as medidas; é consistente através de diferentes projetos e utilizado por diferentes empresas, podendo assim essas empresas trocar informações visando a melhoria de seus processos e produtos; é de fácil compreensão, atendendo a pessoas não técnicas como o usuário final; utilizável desde o início do ciclo de desenvolvimento de um sistema; uma ferramenta para determinar o tamanho de um pacote a ser adquirido através da contagem dos PF que ele possui; auxilia a determinar os benefícios que um pacote operacional pode oferecer a uma empresa, por meio da contagem dos PF que refletem suas necessidades; um meio para medição de estimativas de custo e recursos requeridos para o desenvolvimento e manutenção de software; auxilia na comparação de softwares; acompanhamento da produtividade e qualidade visando melhorar ao máximo o processo de desenvolvimento de sistemas; pode auxiliar também na tomada de decisão quanto a compra de um pacote ou o seu próprio desenvolvimento.

Análise de pontos de função é a melhor alternativa para determinar o tamanho do software nas fases iniciais de projeto, por ser independente de linguagem de programação e padronizado (MOHAGHEGHI & OUTROS,2005 apud LEMOS,2006)

### **Desvantagens**

LEMOS (2006) cita alguns autores que escreveram suas visões a respeito das desvantagens na utilização dos pontos de função, e elas são as seguintes:

KEREMER (1993), em sua visão o ponto fraco identificado na contagem de pontos por função é referente à subjetividade. Em sua pesquisa KEREMER (1993) identificou uma diferença de 12% na contagem de pontos de função do mesmo produto realizado por pessoas diferentes.

Na visão de FELTON & PFLEEGER (1997) existem vários problemas com as medições de pontos de função, e enfatizam que os usuários têm que conhecer tais limitações. São as seguintes limitações:

Problemas com subjetividade no fator tecnologia; problemas com contagem dupla; problemas com valores intuitivos por parte do contador de pontos de função; problemas com precisão; problemas com uso de pontos de função nas etapas iniciais do ciclo de vida; problemas com mudanças de requisitos; problemas em distinguir itens especificados; problemas com dependência de tecnologia; problemas com domínio da aplicação; problemas com peso no quesito subjetividade.

## 12 Definição de contagem de um sistema

A seguir iremos apresentar o sistema que iremos utilizar para aplicação da APF, onde se trata de um gerenciador de conteúdo, que é utilizado para gerenciar todo tipo de informação de uma rede de igrejas em um determinado distrito.

O tipo de contagem que será realizado é o de um projeto em desenvolvimento.

### 12.1 Contagem Sistema de Gerenciamento de Conteúdo Web

Um Sistema de Gestão de Conteúdo - SGC, (em inglês *Content Management Systems - CMS*), é um sistema gestor de web-sites, portais e intranets que integra ferramentas necessárias para criar e gerenciar conteúdos em tempo real, sem a necessidade de um programador, pois o objetivo é estruturar e facilitar a criação, administração, distribuição, publicação e disponibilidade da informação. A sua maior característica é a grande quantidade de funções existentes.

O CMS é um sistema que já possui recursos e ferramentas disponíveis para todo tipo de gerenciamento de páginas web, por exemplo, e que não necessita de um profissional especializado no assunto para administrar suas funções, o que pode gerar uma economia em contratação de pessoal dentro de uma empresa.

Um exemplo famoso de um CMS atualmente é o *WORDPRESS*, que foi desenvolvido na linguagem de programação PHP, a mesma utilizada no sistema em que iremos contar, e que pode ser facilmente modificado/customizado pela internet de qualquer computador.

#### 12.1.1 Sistema “Portal Impacto Jovem”

O principal objetivo desse site é gerenciar o conteúdo do Distrito de Campos Verdes da Igreja Adventista do Sétimo Dia, o que consiste em uma igreja como sede do Distrito e mais 6 igrejas afiliadas.

#### 12.1.2 Funcionalidades do Sistema

O gerenciador possui diversas funcionalidades iniciais solicitadas pelo cliente, e estão listadas abaixo:

- Agenda (Agenda dos Eventos do Distrito e das igrejas)
- Aniversariantes (Dados do aniversário dos membros das igrejas do Distrito)

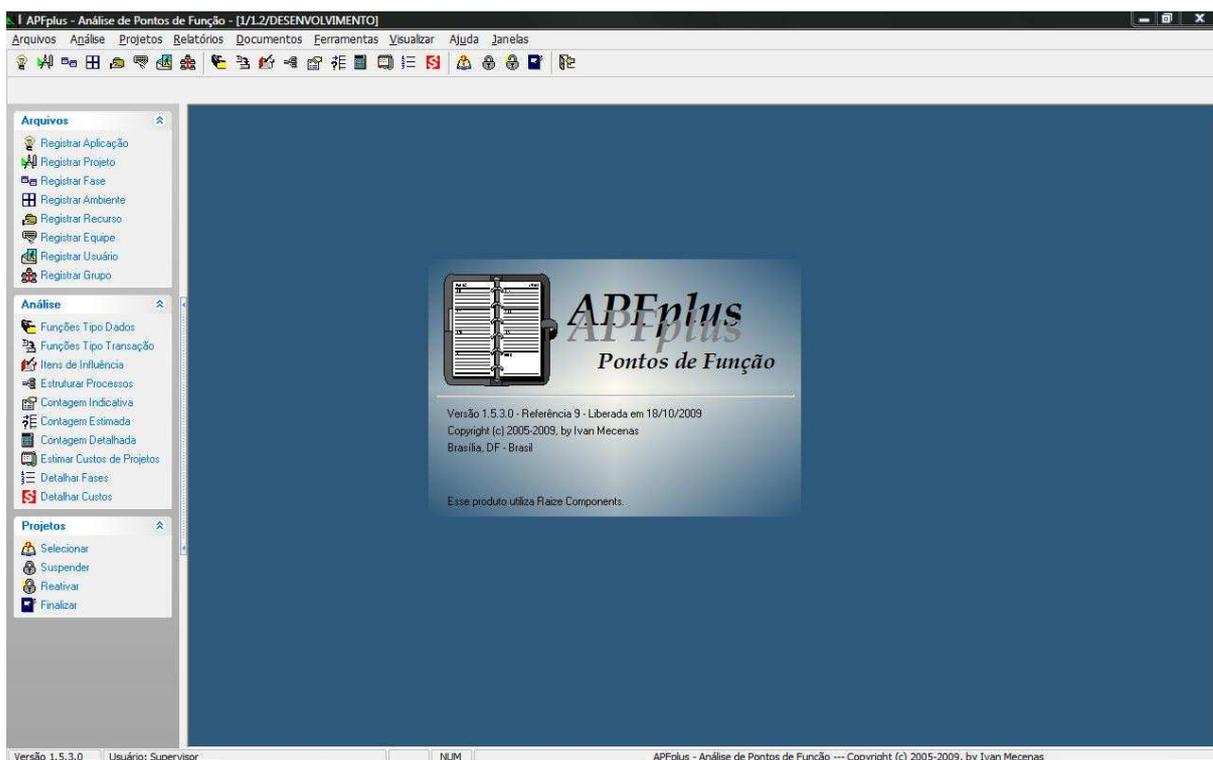
- Chat (Sistema de Bate Papo)
- Contato (Dados pessoais dos usuários de cada igreja)
- Fotos (Fotos das atividades e eventos das igrejas)
- Estudos bíblicos (Link para cursos gratuitos)
- Enquete
- Quiz
- Mural de Recados
- Nisso cremos (Doutrina Adventista)
- Pastor (Informações úteis sobre o pastor distrital, como Apresentação, Agenda de Batismos, Comissões, Evangelismo, Itinerário, Objetivos, Santa Ceia e Visitação)
- Utilidades (Horário do pôr-do-sol, Links Musicais, Receitas)
- Vídeos (Links de vídeos de sites externos)

Baseado nessas funcionalidades é que iremos definir quais funções da APF existentes esse sistema possui.

A seguir iremos apresentar a ferramenta APFplus, com as informações do cálculo dos pontos de função Sistema Portal “Impacto Jovem”.

## 13 Software APFplus

A APFplus - Análise de Pontos de Função foi construída para dar suporte à contagem dos pontos de função de aplicações, tanto em desenvolvimento quanto a melhoria. A ferramenta está em conformidade com a metodologia de contagem prevista no Manual de Práticas de Contagens, versão 4.2.1, do IFPUG.

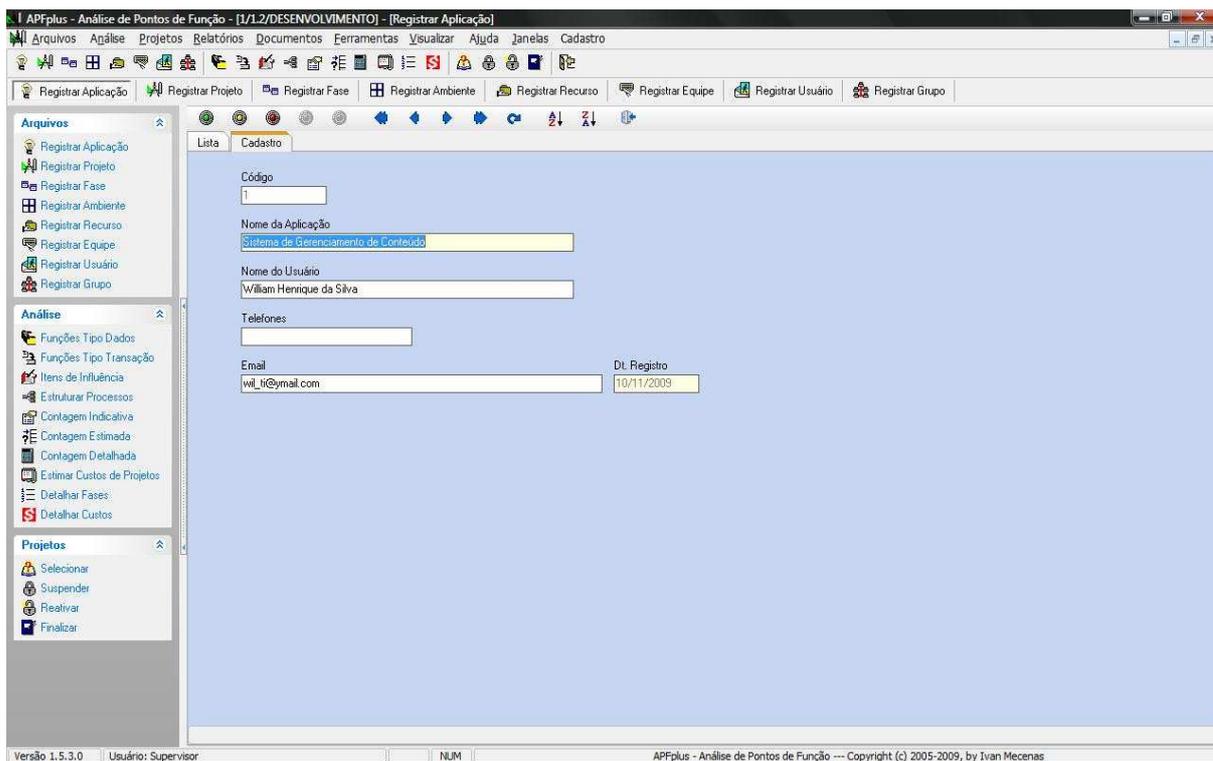


**Figura 19 Apresentação APFplus**

A ferramenta incorpora os métodos de Contagem Indicativa e de Contagem Estimada de Pontos de Função, conforme proposto pela NESMA.

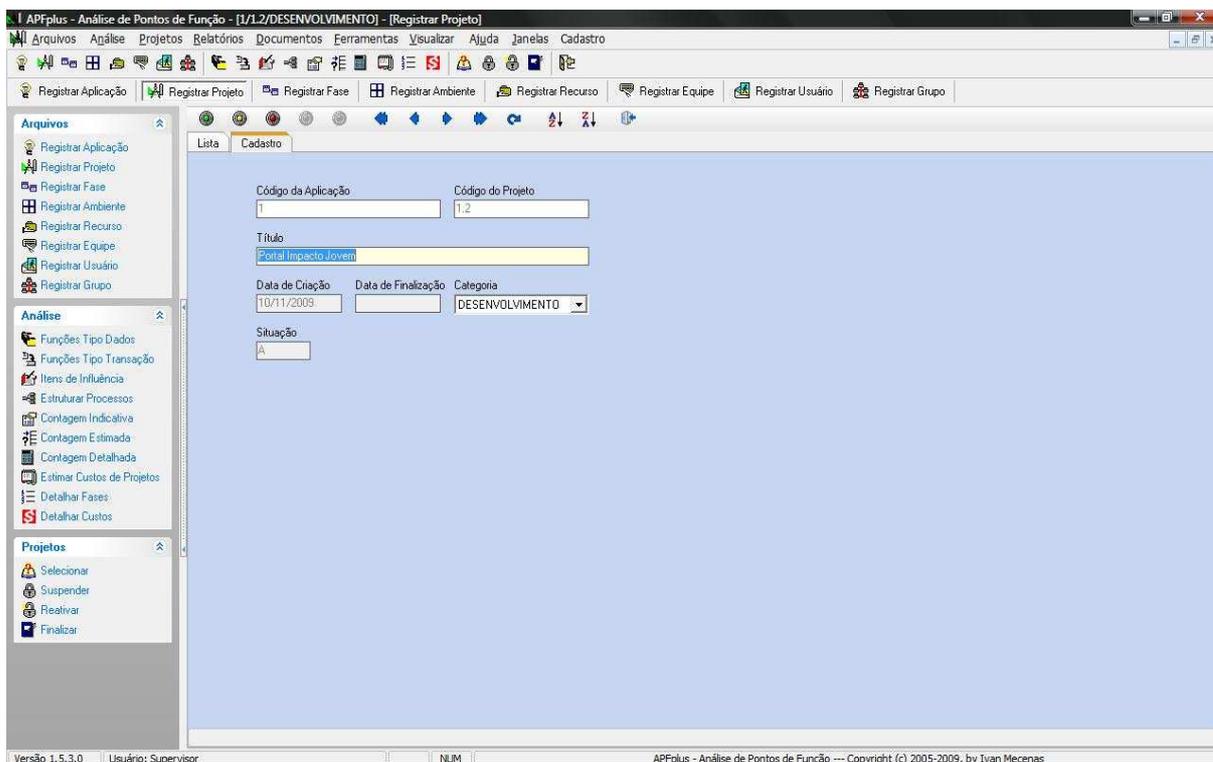
A seguir iremos abordar o passo a passo da ferramenta APFplus, onde explicaremos cada funcionalidade sua.

Para que seja iniciada a contagem é necessário registrar a aplicação, onde cada aplicação pode possuir vários projetos associados ao longo de sua existência, tanto de melhoria quanto de desenvolvimento.



**Figura 20 Cadastro da Aplicação**

Logo depois de cadastrado devemos registrar o projeto em questão. Cada projeto deverá estar associado a uma determinada aplicação e os projetos se dividem em duas categorias: desenvolvimento e melhoria.

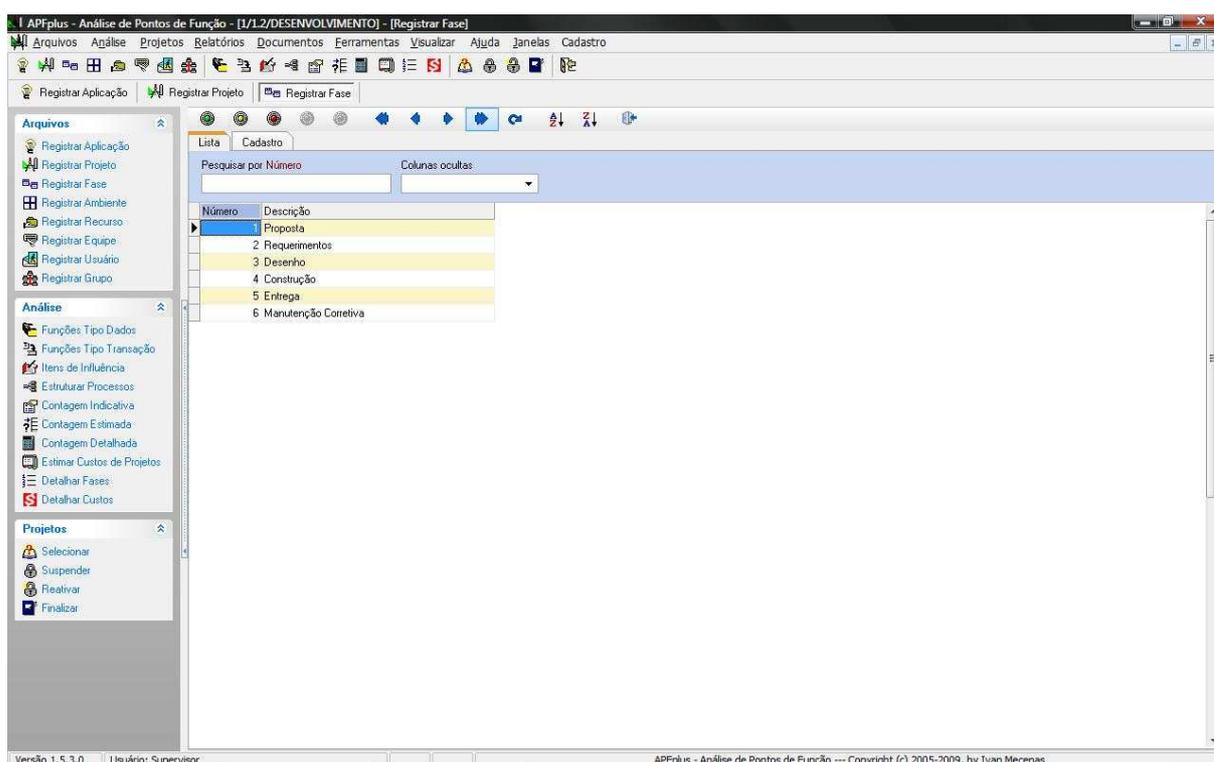


**Figura 21 Cadastro do Projeto**

Quando você cria um projeto de melhoria, todas as funções (tipo dados e tipo transações) e itens de influência do projeto de desenvolvimento são exportadas para o projeto de melhoria. Isso facilita o trabalho de alteração ou exclusão de funções. Somente um projeto de desenvolvimento pode estar associado a uma aplicação, e vários projetos de melhoria poderão se referir a mesma aplicação, eles só não poderão ser aplicados quando o projeto de desenvolvimento ainda estiver em andamento ou estiver suspenso.

Devemos também registrar a fase em que a aplicação está e para isso devemos classificá-la de acordo com o seu ciclo de vida, e de acordo com a proposta oferecida pela metodologia de APF existe uma classificação padrão que é:

- Proposta (*Proposal*)
- Requerimentos (*Requirements*)
- Desenho (*Design*)
- Construção (*Construction*)
- Entrega (*Delivery*)
- Manutenção Corretiva (*Corrective Maintenance*)



**Figura 22 Registro de Fase**

Podemos ainda adicionar ou modificar as fases, mas não excluí-las, de acordo com a necessidade do usuário.

É necessário também o registro do(s) ambiente(s) em que será desenvolvida a aplicação e a cada ambiente devemos associar dois parâmetros essenciais que são Esforço e Produtividade.

Código	Descrição	Esforço (h/pf)	Produtividade (pf/h)	KSLOC
1	COBOL	16,00	0,0625	
2	COBOL II	13,80	0,0725	
3	ADS	6,50	0,1538	
4	ORACLE	10,30	0,0971	
5	PL/I	13,60	0,0735	
6	CSP	32,10	0,0312	
7	EASYTRIEVE	13,00	0,0769	
8	VISUAL BASIC	7,50	0,1333	
9	SAS	11,20	0,0893	
10	NATURAL	9,60	0,1042	
11	CLIPPER	8,80	0,1136	
12	C	14,90	0,0671	
13	C++	17,40	0,0575	
14	ACCESS	2,00	0,5000	
15	JAVA	19,60	0,0510	
16	SQL	13,60	0,0735	
17	ABAP	13,80	0,0725	
18	PERIPHONICS	18,60	0,0538	
19	TELON	10,90	0,0917	
20	Delphi/Firebird	4,00	0,2500	
21	Delphi/InterBase	4,00	0,2500	
22	Delphi/Paradox	3,80	0,2632	
23	PHP/MySQL	15,00	0,0667	

**Figura 23 Registro de Ambiente**

O parâmetro Esforço é dado em horas por ponto de função (h/pf), ou seja, a quantidade de horas necessárias para se construir um ponto de função, já o parâmetro Produtividade é o inverso do parâmetro Esforço pois ele é dado em pontos de função por hora (pf/h).

A dedução dos valores desses parâmetros são baseados em estatísticas aplicadas a vários projetos ao longo do tempo. Pesquisas realizadas em 1994 pela ASMA (*Australian Software Metrics Association*), com base em 54 projetos, indicaram os seguintes valores para o parâmetro Esforço:

- a) 8,8h/pf (utilizando linguagens de terceira geração)
- b) 4,4 h/pf (utilizando linguagens de quarta geração)

Existe ainda a opção do parâmetro KSLOC/PF (*backfiring*) que se baseia na premissa de alguns autores que afirmam haver uma proporcionalidade entre o total de pontos de função não ajustados e o número de linhas de código fonte (linhas físicas, no COCOMO 81, e linhas lógicas, no COCOMO II). Esse parâmetro somente deve ser utilizado quando não se conhece o parâmetro Esforço e não é recomendado.

Devemos registrar recursos (cargos), para que possam ser utilizados no desenvolvimento e manutenção de projetos, pois é necessário identificar cada cargo porque é baseado no recurso que definimos o tempo e o esforço empregado. Exemplos:

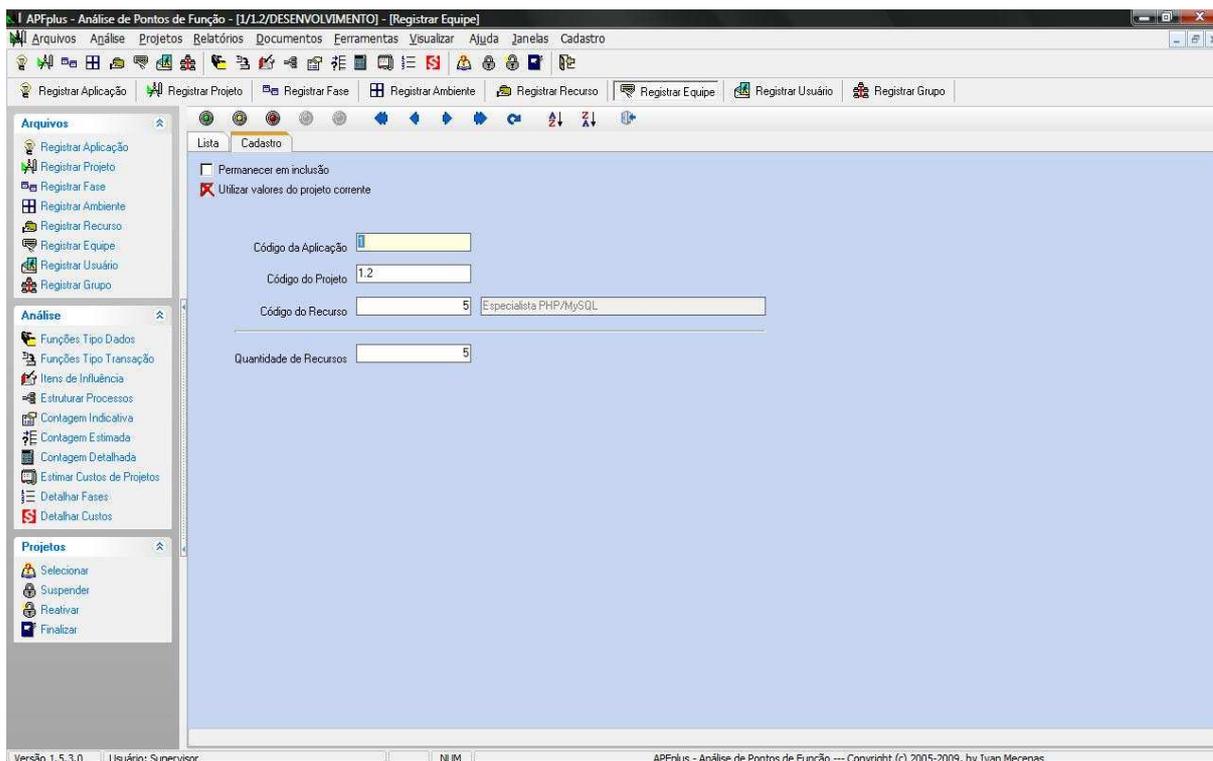
- Analista Sênior
- Analista Pleno
- Analista Júnior
- Estagiário

Código	Descrição	Carga Horária/Dia	Custo/Hora
1	Analista Programador Jr	8,00	20,00
2	Analista Programador Pleno	8,00	35,00
3	Analista Programador Senior	8,00	80,00
4	Estagiario	6,00	10,00
5	Especialista PHP/MySQL	8,00	150,00

**Figura 24 Registro de Recursos**

É com base no cargo que é empregado quanto tempo é gasto para desenvolver um PF.

A ferramenta permite também registrar equipes, pois isso é essencial para se estimar o custo dos projetos e o tempo necessário para realizá-los. Na realidade, equipe é a combinação de projeto, ambiente e recurso. Assim, esses elementos devem ser registrados com antecedência.



**Figura 25 Registro de Equipe**

Depois de elaborada a equipe, devemos estabelecer os usuários que irão utilizar o software, pois antes disso o único usuário existente é o “Supervisor”, que não pode ser apagado e é identificado pelo código “0”. Existem três níveis de classificação, que são:

- Supervisão
- Gerência
- Técnico

Código	Nome	Nível de Acesso	Situação
0	Supervisor	Supervisão	A
1	Antonio Santos	Técnico	A
2	William Henrique	Gerência	A

**Figura 26 Registro de Usuários**

O usuário classificado com o nível “Supervisão” possui as mesmas funcionalidades do “Supervisor”, mas para isso acontecer ele tem que ser alocado em todos os grupos do projeto, porém esse acesso não pode ser dado para qualquer usuário.

Todos os usuários com a permissão de “Gerência” não tem acesso as seguintes funções:

- Registro de Fase
- Registro de Usuário
- Registro de Grupo
- Finalização de Projeto

Todos os usuários com a permissão de “Técnico” não têm acesso às seguintes funções:

- Registro de Aplicação
- Registro de Projeto
- Registro de Fase
- Registro de Recurso
- Registro de Equipes
- Registro de Usuários
- Registro de Grupos
- Estimar Custo de Projetos
- Detalhar Fases
- Suspende Projeto
- Reativar Projeto
- Finalizar Projeto

Os usuários com nível de acesso "Gerência" ou "Técnico" recebem os seguintes direitos quando são alocados em um grupo de projetos no papel de "Liderança":

- Estimar Custo de Projetos
- Suspende Projeto
- Reativar Projeto
- Finalizar Projeto.

Com os usuários gerados, devemos criar os grupos, onde a funcionalidade da ferramenta permite que você associe um usuário a um projeto específico. Vários usuários associados a um determinado projeto formam o "grupo" do projeto.

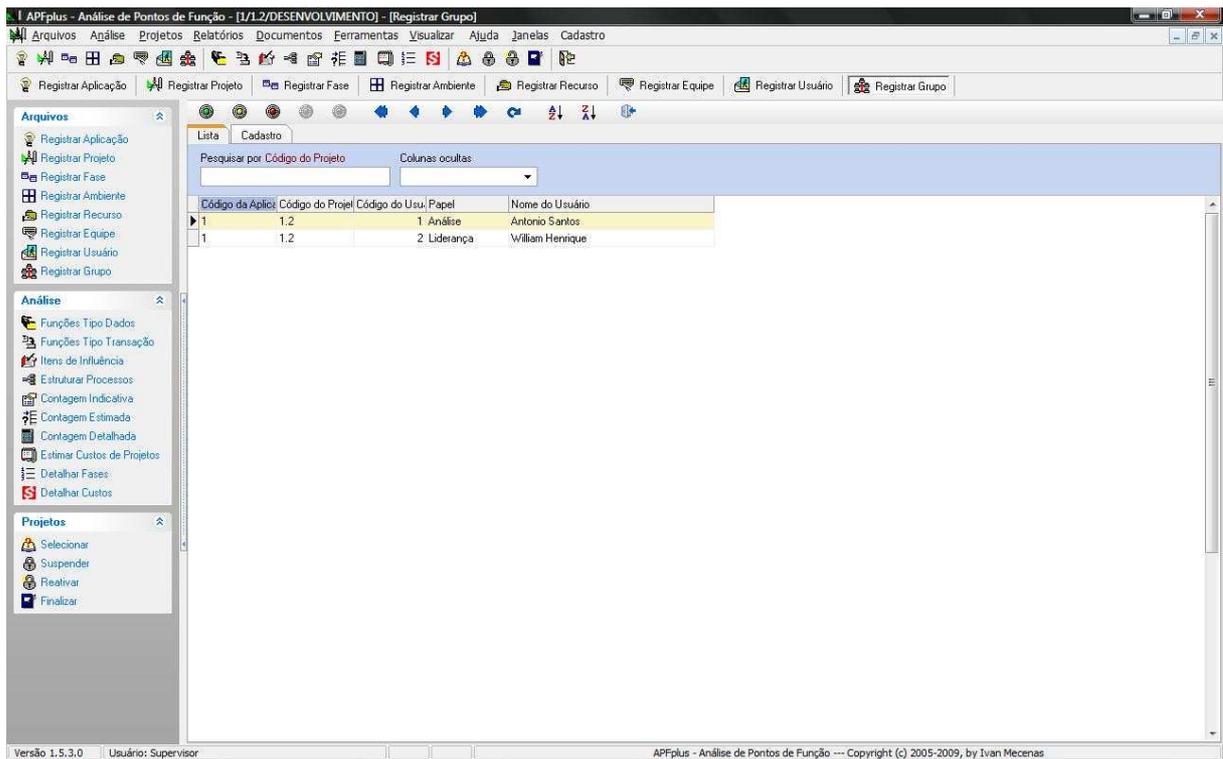


Figura 27 Registro de Grupo

Há dois tipos de papel que podem ser atribuídos a um usuário de um grupo: Liderança e Análise.

Normalmente um dos usuários do grupo é registrado com o papel de "Liderança". Os demais são cadastrados com o papel de "Análise".

O papel de "Liderança" dará ao usuário o direito de acesso às seguintes funções:

- Estimar Custo de Projetos
- Suspende Projeto
- Reativar Projeto
- Finalizar Projeto

O papel do usuário complementa a característica "Acesso" atribuída ao usuário.

Depois de inserir todas as informações necessárias para contagem, será feito o registro das funções, seja do tipo dado ou do tipo de transação, de um determinado projeto. Em outras palavras, devemos registrar as do tipo dado:

- ALI
- AIE

Identificador	Código da Aplicação	Código do Projeto	Função	Descrição	Tipos de Registro	Tipos de Dados	Complexidade	Pontos de Função
1	1	1.2	ALI	Perfil Particular do Usuario	14	4 Média		10
2	1	1.2	ALI	Itens do Quiz	3	2 Baixa		7
3	1	1.2	ALI	Perfil Pastor	8	1 Média		10
4	1	1.2	ALI	Cartão Virtual	5	2 Baixa		7
5	1	1.2	ALI	Perfil Administrador	15	3 Média		10

**Figura 28 Registro de funções do tipo dado**

E registrar as funções do tipo transação:

- EE
- SE
- CE

Identificador	Código da Apli	Código do Projeto	Função	Descrição	Arquivos Referenciados	Tipos de Dados	Complexidade	Pontos de Função
1	1	1.2	EE	Agenda	0	4	Baixa	3
2	1	1.2	EE	Aniversariantes	1	3	Baixa	3
3	1	1.2	EE	Contatos	2	10	Média	4
4	1	1.2	EE	Destaques	1	5	Baixa	3
5	1	1.2	EE	Fotos	2	10	Média	4
6	1	1.2	EE	Noticias	1	3	Baixa	3
7	1	1.2	CE	Videos	1	4	Baixa	3
8	1	1.2	EE	Perguntas sobre o Q	0	5	Baixa	3
9	1	1.2	CE	Inserção de video at	1	1	Baixa	3
10	1	1.2	CE	Consulta sobre por d	3	3	Baixa	3
11	1	1.2	CE	Como chegar	1	3	Baixa	3
12	1	1.2	CE	Horário de Cultos	0	1	Baixa	3
13	1	1.2	EE	Nisso cremos	1	1	Baixa	3

**Figura 29 Registro de funções do tipo transação**

A ferramenta oferece também a opção de registrar entradas externas temporárias, aquelas que se referem à conversão de dados de sistemas existentes, e essas entradas externas são codificadas como EEC (Entradas Externas-Conversão).

Logo após devemos estabelecer os itens de influencia, de acordo com o projeto, identificando as 14 características para serem assinaladas.

Atualização On-line	Processamento Complexo	Reusabilidade	Facilidade de Instalação	Facilidade de Operação	Múltiplos Locais	Modificação Facilitada
Comunicação de Dados	Processamento Distribuído	Performance	Configuração Altamente Utilizada	Volume de Transações	Entrada de Dados On-line	Eficiência do Usuário Final

**Comunicação de Dados**

Descreve até que ponto a aplicação se comunica diretamente com o processador.

Os dados e informações de controle utilizados pela aplicação são enviados ou recebidos através de recursos de comunicação. Considera-se que os dispositivos conectados localmente à unidade de controle utilizam recursos de comunicação.

O protocolo é um conjunto de convenções que permite a transferência ou intercâmbio de informações entre dois sistemas ou dispositivos.

Todos os links de comunicação de dados necessitam de algum tipo de protocolo.

**Itens**

- A aplicação é essencialmente batch ou uma estação de trabalho isolada.
- A aplicação é essencialmente batch, mas possui entrada de dados ou impressão remota.
- A aplicação é batch, mas possui entrada de dados e impressão remota.
- A aplicação possui entrada de dados on-line ou front-end de teleprocessamento para um processamento batch ou sistema de consulta.
- A aplicação é mais que um front-end, mas suporta apenas um tipo de protocolo de comunicação.
- A aplicação é mais que um front-end e suporta mais de um tipo de protocolo de comunicação.

**Figura 30 Registro dos Itens de Influência**

### 13.1 Contagem Indicativa

Essa função permite realizar uma contagem indicativa dos pontos de função de um determinado projeto, em qualquer momento. O método de contagem indicativa observa as definições propostas pelo NESMA, e para realizar esse tipo de contagem basta clicar no botão "Calcular".

Ela baseia-se exclusivamente nos ALI e nos AIE do projeto. A contagem pode (deve) ser utilizada quando não existem outras informações além dos ALI e AIE do projeto, assumindo que cada ALI possui 35 pontos de função e cada AIE possui 15 pontos.

Funções		Características Gerais da Aplicação	
<b>Pontos de Função (detalhe)</b>		Comunicação de Dados	<b>3</b>
Arquivos Lógicos Internos	<b>175</b>	Processamento Distribuído	<b>4</b>
Arquivos de Interface Externa	<b>0</b>	Performance	<b>1</b>
Entradas Externas	.....	Configuração Altamente Utilizada	<b>0</b>
Saídas Externas	.....	Volume de Transações	<b>0</b>
Consultas Externas	.....	Entrada de Dados On-line	<b>5</b>
Conversão de Dados	.....	Eficiência do Usuário Final	<b>3</b>
		Atualizações On-line	<b>0</b>
		Processamento Complexo	<b>1</b>
		Reusabilidade	<b>1</b>
		Facilidade de Instalação	<b>0</b>
		Facilidade de Operação	<b>1</b>
		Múltiplos Locais	<b>0</b>
		Modificação Facilitada	<b>0</b>
		<b>Total</b>	<b>19</b>
<b>Pontos de Função (resumo)</b>			
Não Ajustados	<b>175,00</b>		
Fator de Ajuste	<b>0,84</b>		
Total de Pontos	<b>147,00</b>		
<b>Tamanho do Projeto</b>			
<b>Pequeno</b>			

Figura 31 Contagem Indicativa

## 13.2 Contagem Estimada

Essa contagem também realiza uma estimativa dos pontos de função de um determinado projeto, em qualquer momento. O método de contagem estimada observa as definições propostas pelo NESMA também. A contagem estimada segue o mesmo padrão da contagem indicativa, porém ela assume que as funções dados (ALI e AIE) apresentam grau de complexidade Simples e que as funções transação (EE, EEC, SE e CE) têm grau de complexidade Médio.

Funções		Características Gerais da Aplicação	
<b>Pontos de Função (detalhe)</b>		Comunicação de Dados	3
Arquivos Lógicos Internos	35	Processamento Distribuído	4
Arquivos de Interface Externa	0	Performance	1
Entradas Externas	32	Configuração Altamente Utilizada	0
Saídas Externas	0	Volume de Transações	0
Consultas Externas	20	Entrada de Dados On-line	5
Conversão de Dados	0	Eficiência do Usuário Final	3
<b>Pontos de Função (resumo)</b>		Atualizações On-line	0
Não Ajustados	87,00	Processamento Complexo	1
Fator de Ajuste	0,84	Reusabilidade	1
Total de Pontos	73,08	Facilidade de Instalação	0
<b>Tamanho do Projeto</b>		Facilidade de Operação	1
<b>Muito Pequeno</b>		Múltiplos Locais	0
		Modificação Facilitada	0
		<b>Total</b>	<b>19</b>

Figura 32 Contagem Estimada

Após definido todos esses processos, devemos estimar os custos que o projeto terá.

The screenshot shows the 'Estimar Custo de Projetos' window in APFplus. The main area displays the following parameters:

- Pontos de Função: 85,00
- Definição do Projeto: Muito Pequeno
- Parâmetros para Cálculo: Ambiente (dropdown), KSLOC/FF (Backfitting) (checkbox), Custo/FF: 50,00
- Estorço (h/pf) (input), Produtividade (pl/h) (input), Quantidade de dias no mês: 19
- Equação de Tempo = 3,67 x [E°C] ==> (input) x (input) ^ (input) <== Utilizar (checkbox)

Below these parameters is a table for 'Custo Previsto' (COCOMO II):

Código do Recurso	Quantidade (Recurso)	Carga Horária	Custo/Hora

On the right side, there is a 'PLANILHA' section with fields for:

- Prazo Ideal (meses)
- Equipe Ideal (pessoas)
- Prazo Previsto (meses)
- Equipe Prevista (pessoas)
- Horas Previstas
- Custo Previsto (por prazo)
- Custo Previsto (por PF)
- Fases atuais: 1:6

At the bottom, there are buttons for 'Alocar', 'Excluir', and 'Apropriar Custos'. The status bar at the bottom indicates 'Versão 1.5.3.0', 'Usuário: Supervisor', and 'APFplus - Análise de Pontos de Função --- Copyright (c) 2005-2009, by Ivan Mecenas'.

Figura 33 Estimativa de Custos

No momento em que está se estimando o custo, devemos definir o ambiente em que o projeto será feito.

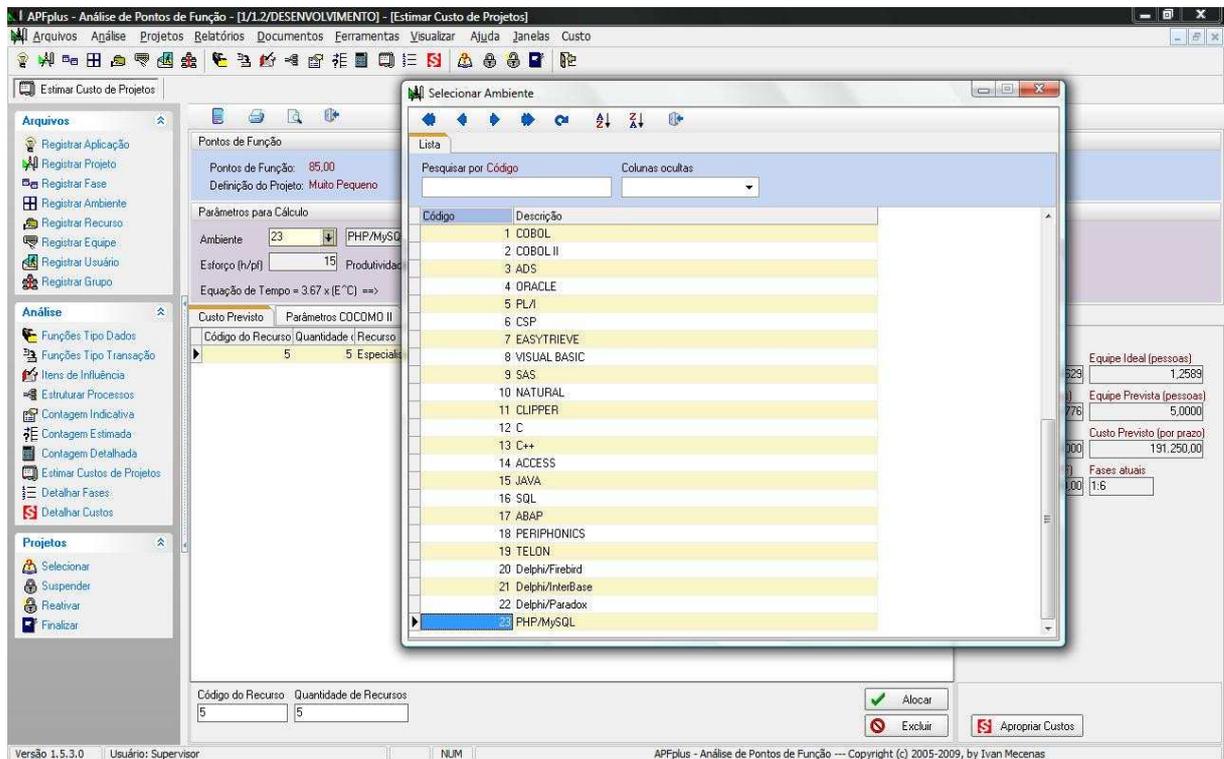


Figura 34 Definição de Ambiente

Em seguida alocamos os recursos (Profissionais) que participarão do projeto.

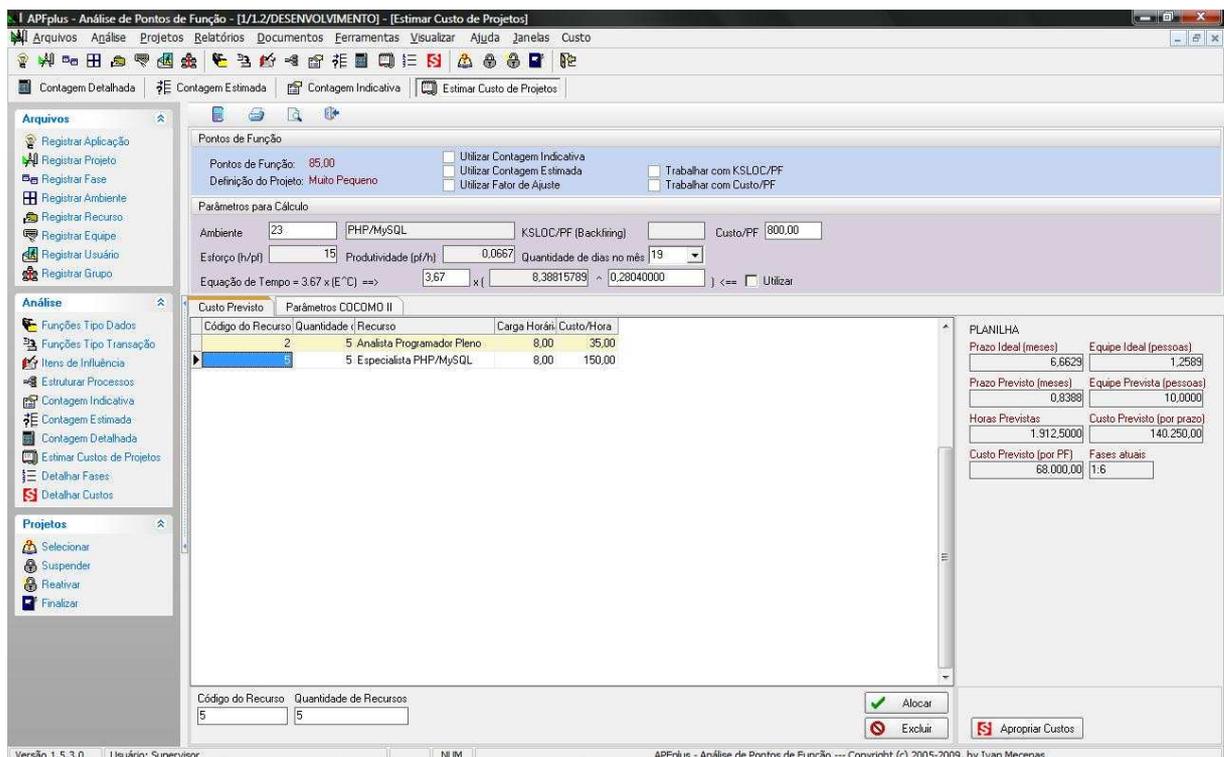


Figura 35 Definição de Recurso

A estimativa de custo de projetos é feita com base no tamanho do projeto (pontos de função) e na produtividade do ambiente de desenvolvimento do projeto.

Nesse momento podemos visualizar a planilha de custos onde aparecem, entre outros dados, os parâmetros Prazo Ideal e Equipe Ideal.

Não devemos calcular o prazo de forma linear, pois há um limite para se reduzir o prazo de desenvolvimento, ou seja, se 1 pessoa consegue desenvolver um projeto em 100 horas, isso não significa dizer que 100 pessoas fariam o trabalho em 1 hora.

A partir do cálculo do Prazo Ideal e da Equipe Ideal feito pela ferramenta, deve-se alocar a equipe que desenvolverá o projeto (Equipe Prevista), para que se estime o Prazo Previsto e o respectivo custo de desenvolvimento. Os botões "Alocar" e "Excluir" permitem inserir (ou alterar) e excluir recursos. Os recursos disponíveis devem ter sido registrados no momento em que se registram os recursos.

É importante que a montagem da equipe esteja em torno da Equipe Ideal calculada pela ferramenta. Assim, se o sistema calcular uma equipe de 0,5 pessoa, deve-se alocar 1 pessoa, pois é impossível alocar meia pessoa. Porém, se a ferramenta calcular 2,4 pessoas, pode-se alocar 2 ou 3 pessoas para o projeto. Alocando-se 2, o prazo previsto de desenvolvimento será maior que o prazo previsto para 3 pessoas. Não é prudente alocar mais de 3 pessoas para que não se caia na zona de prazo previsto impossível de ser cumprido.

Embora o custo de um projeto de software seja normalmente calculado pelo critério de prazo de desenvolvimento, a ferramenta permite que o custo seja calculado pelo total de pontos de função. Para isso, é necessário informar, no campo Custo/PF, o valor referente a um ponto de função e acionar o botão Calcular Custo. Além disso, deve-se marcar a opção Trabalhar com Custo/PF. Com isso, a apropriação de custos e a impressão e visualização de relatórios será feita com base no valor do campo Custo Previsto (por PF).

### **13.3 Estimativa de Investimento para o Sistema Portal “Impacto Jovem”**

Tendo como base as discussões em fórum<sup>1</sup> sobre APF, o valor do ponto de função no Brasil, pode variar de 300,00 a 1000,00 reais, isso se dá muitas vezes pelo fato de que as grandes empresas que utilizam esse método, são empresas do governo e para se realizar um projeto é necessária uma licitação e as fábricas de software concorrentes, devem apresentar um escopo que atenda as exigências da licitação. Com base nisso, acreditamos que o investimento empregado pela instituição que solicitou o Portal “Impacto Jovem” seria de 300,00 reais por ponto de função. O valor a ser investido deve variar de acordo com a linguagem utilizada para o desenvolvimento do software.

## **14 Considerações Finais**

Nosso estudo teve como objetivo apresentar o conceito sobre Análise de Pontos de Função e demonstrar uma ferramenta que utilizasse essa metodologia com eficiência e clareza.

Para que fosse apresentada uma estimativa de custo e tempo de um projeto de software, constatamos que é de suma importância mensurar o tamanho de um software, pois é com base em seu tamanho que definimos o seu custo e tempo empregado em sua construção. Além disso, a estimativa de custo e tempo possibilita a distribuição de mão de obra, de acordo com prazo e custo estabelecidos.

Para que se tenha uma estimativa cada vez mais precisa, a instituição que utiliza a metodologia de APF deve estar constantemente exercitando esse conceito, porque é baseado nos históricos de contagens que iremos buscar informações para contagens futuras.

Chegamos à conclusão de que essa metodologia é sem dúvida a melhor ferramenta para se estimar custo, prazo e tempo, pois sempre se baseia em fórmulas matemáticas e normas internacionais, fazendo com que ela tenha uma credibilidade extraordinária.

---

<sup>1</sup> [forum-bfpug@yahoo grupos.com.br](mailto:forum-bfpug@yahoo grupos.com.br); [livro-apf@yahoo grupos.com.br](mailto:livro-apf@yahoo grupos.com.br), São fóruns de discussão sobre a metodologia, onde diversos especialistas certificados buscam sanar dúvidas e divulgar informações pertinentes à APF. Esse fórum é aberto ao público e possui um acervo interessante para se tirar dúvidas e curiosidades sobre o assunto.

## 15 Referência Bibliográfica

VAZQUEZ, C. E. ; SIMÕES, G. S. ; ALBERT, R. M. **Análise de Pontos de Função: Medição, Estimativas e Gerenciamento de Projetos de Software.** São Paulo: Érica, 2008.  
BRAGA, A. **Análise de Pontos de Função.** São Paulo: Infobook S. A., 1996.

LEMOS, J. C. **Desenvolvimento de uma Ferramenta Automatizada para Suporte à Análise de Pontos de Função Formando uma Baseline de Projetos de Software.** 2006. 118f. Dissertação (Mestrado em Ciências da Computação) – Pós-Graduação Ciências da Computação, Universidade Metodista de Piracicaba, Piracicaba, 2006.

**DEKKERS CAROL A. Pontos de Função e Medidas.** 1998. Disponível em:<<http://www.bfpug.com.br>>. Acesso em: 22 Mai. 2009.

RAFAEL TRINDADE. **Análise por Pontos de Função.** 2008. Disponível em:<<http://rafaeltrindade.com.br>>. Acesso em: 10 Jun. 2009.

ANDRÉ MANCHESTER. **Benefícios da Análise de Pontos de Função como Instrumento de Apoio para a Gestão de Projetos de Sistemas.** 2007. Disponível em:<<http://www.abramti.org.br>>. Acesso em: 25 Jun. 2009.

MECENAS, IVAN. **Software APFplus,** 2009. Disponível em:<<http://www.ivanmecenas.ecn.br/apf.htm>>. Acesso em 18 Jun. 2009.