

**Instituto de Ciências Matemáticas e de Computação**

ISSN - 0103-2569

**PONTOS DE FUNÇÃO**

**Uma medida funcional de tamanho de software**

**WAINE TEIXEIRA JÚNIOR  
ROSELY SANCHES**

**Nº 105**

**RELATÓRIOS TÉCNICOS DO ICMC**

São Carlos

**MARÇO/2000**

# SUMÁRIO

LISTA DE ABREVIATURAS .....	i
ÍNDICE DE EQUAÇÕES .....	ii
ÍNDICE DE FIGURAS .....	iii
ÍNDICE DE QUADROS .....	iv
ÍNDICE DE TABELAS .....	v
<b>1 INTRODUÇÃO - VISÃO GERAL DE PONTOS DE FUNÇÃO .....</b>	<b>1</b>
1.1 HISTÓRICO DE PONTOS DE FUNÇÃO.....	1
1.2 USOS DE PONTOS DE FUNÇÃO .....	5
1.3 VISÃO GERAL DO PROCESSO DE ANÁLISE DE PONTOS DE FUNÇÃO .....	6
1.4 UM EXEMPLO SIMPLIFICADO.....	8
<b>2 REGRAS PARA IDENTIFICAÇÃO E CONTAGEM DE FUNÇÕES.....</b>	<b>10</b>
2.1 IDENTIFICAÇÃO DA FRONTEIRA DA APLICAÇÃO .....	10
2.2 REGRAS DE CONTAGEM PARA ARQUIVOS LÓGICOS INTERNOS (ALI) .....	11
2.2.1 Complexidade de Arquivo Lógico Interno.....	12
2.3 REGRAS DE CONTAGEM PARA ARQUIVOS DE INTERFACE EXTERNA (AIE).....	14
2.3.1 Complexidade de Arquivo de Interface Externa.....	15
2.4 REGRAS DE CONTAGEM PARA ENTRADAS EXTERNAS (EE).....	16
2.4.1 Complexidade de Entradas Externas.....	17
2.5 REGRAS DE CONTAGEM PARA SAÍDAS EXTERNAS (SE) .....	19
2.5.1 Complexidade de Saídas Externas.....	20
2.6 REGRAS DE CONTAGEM PARA CONSULTAS EXTERNAS (CE) .....	23
2.6.1 Complexidade de Consulta Externa.....	24
<b>3 CÁLCULO DO FATOR DE AJUSTE .....</b>	<b>28</b>
3.1 DETERMINAÇÃO DO NÍVEL DE INFLUÊNCIA (NI) .....	28
3.1.1 Comunicação de dados.....	29
3.1.2 Processamento distribuído ou Funções distribuídas .....	29
3.1.3 Performance .....	30
3.1.4 Configuração do equipamento.....	30
3.1.5 Volume de transações .....	31
3.1.6 Entrada de dados on-line.....	31
3.1.7 Interface com o usuário .....	31
3.1.8 Atualização on-line.....	32
3.1.9 Processamento complexo.....	32
3.1.10 Reusabilidade .....	33
3.1.11 Facilidade de implantação .....	33
3.1.12 Facilidade operacional.....	34
3.1.13 Múltiplos locais.....	34
3.1.14 Facilidade de mudanças (flexibilidade).....	35
3.2 CÁLCULO DO FATOR DE AJUSTE E DOS PONTOS DE FUNÇÃO AJUSTADOS.....	35
<b>4 CONTAGEM DE PONTOS DE FUNÇÃO EM PROJETOS .....</b>	<b>37</b>
4.1 CONTAGENS E FASES DE DESENVOLVIMENTO .....	37
4.2 EXEMPLO DE ANÁLISE DE PONTOS DE FUNÇÃO EM PROJETO DE NOVA APLICAÇÃO .....	38
4.2.1 Descrição do sistema a ser desenvolvido .....	38
4.2.2 Levantamento das principais funções do sistema .....	39
4.2.3 Layout dos Arquivos Internos e Externos do sistema .....	39
4.2.4 Layout das telas do sistema .....	40
4.2.5 Layout dos Relatórios do sistema .....	44
4.2.6 Identificação das funções do sistema.....	44
4.2.7 Classificação das funções quanto a complexidade.....	45

4.2.8	<i>Cálculo dos pontos de função não ajustados.....</i>	<i>46</i>
4.2.9	<i>Cálculo do Fator de Ajuste e dos Pontos de Função Ajustados.....</i>	<i>47</i>
<b>5</b>	<b>CONTAGEM DE PF EM PROJETOS DE MANUTENÇÃO DE APLICAÇÕES.....</b>	<b>49</b>
5.1	EXEMPLO DE ANÁLISE DE PONTOS DE FUNÇÃO DE UM PROJETO DE MANUTENÇÃO.....	51
5.1.1	<i>Descrição do escopo do projeto de manutenção do sistema .....</i>	<i>52</i>
5.1.2	<i>Layout das telas, arquivos e relatórios afetados, e/ou incluídos no sistema .....</i>	<i>52</i>
5.1.3	<i>Identificação das modificações, inclusões e exclusões no sistema .....</i>	<i>55</i>
5.1.4	<i>Classificação das funções incluídas, excluídas e modificadas no sistema .....</i>	<i>55</i>
5.1.5	<i>Cálculo de PFs Brutos das modificações no sistema.....</i>	<i>57</i>
5.1.6	<i>Cálculo do novo fator de ajuste dos pontos de função .....</i>	<i>59</i>
5.1.7	<i>Cálculo dos pontos de função ajustados do projeto de manutenção.....</i>	<i>59</i>
5.1.8	<i>Cálculo dos pontos de função ajustados após o projeto de manutenção .....</i>	<i>59</i>
<b>6</b>	<b>PONTOS DE FUNÇÃO E ESTIMATIVAS .....</b>	<b>61</b>
6.1	ESTIMATIVAS DE TAMANHO DE PROJETO .....	61
6.1.1	<i>A importância da Base de Dados Histórica.....</i>	<i>62</i>
6.2	MÉTODO DE INFERÊNCIA PARA ESTIMATIVA DO PRAZO DO PROJETO .....	64
6.3	MÉTODOS DE INFERÊNCIA PARA ESTIMATIVA DO ESFORÇO DO PROJETO .....	64
6.3.1	<i>Transformação por Interpolação.....</i>	<i>65</i>
6.3.2	<i>Transformação Direta .....</i>	<i>65</i>
<b>7</b>	<b>CONCLUSÃO .....</b>	<b>66</b>
<b>8</b>	<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>70</b>

## LISTA DE ABREVIATURAS

AIE	ARQUIVOS DE INTERFACE EXTERNA
ALI	ARQUIVOS LÓGICOS INTERNOS
ALR	ARQUIVO LÓGICO REFERENCIADO
BFPUG	BRAZILIAN FUNCTION POINTS USERS GROUP
CE	CONSULTAS EXTERNAS
COCOMO	CONSTRUCTIVE COST MODEL
CPM	MANUAL DE PRÁTICAS DE CONTAGEM DE PONTOS DE FUNÇÃO
DER	DADO ELEMENTAR REFERENCIADO
EE	ENTRADAS EXTERNAS
FPA	FUNCION POINT ANALYSIS
GUIDE	GRUPO DE USUÁRIOS IBM
IFPUG	INTERNATIONAL FUNCION POINTS USERS GROUP
ISO	INTERNATIONAL STANDARDIZATION ORGANIZATION
LOC	LINHAS DE CÓDIGO
PF	PONTO DE FUNÇÃO
RLR	REGISTRO LÓGICO REFERENCIADO
SE	SAÍDAS EXTERNAS

## ÍNDICE DE EQUAÇÕES

Equação 1 Equação do Fator de Ajuste .....	35
Equação 2 Equação de Cálculo de PFs Ajustados .....	35
Equação 3 Cálculo de Pontos de Função do Projeto de Manutenção .....	59
Equação 4 Cálculo após Projeto de Manutenção .....	60
Equação 5 Interpolação Aritmética.....	64
Equação 6 Transformação Direta de PFs em Esforço.....	65

## ÍNDICE DE FIGURAS

Figura 1 Componentes de Pontos de Função em relação à aplicação [Dekkers, 1998] .....	3
Figura 2 Visão Geral do Processo de Análise de Pontos de Função.....	7
Figura 3 Identificação da Fronteira da Aplicação .....	10
Figura 4 <i>Layout</i> da Tela de Inclusão dos Dados de Clientes .....	19
Figura 5 <i>Layout</i> do Relatório de Dados de Clientes .....	22
Figura 6 <i>Layout</i> da Tela de Consulta ao Cadastro de Clientes.....	26
Figura 7 Tela de Inclusão de clientes.....	40
Figura 8 Exclusão de clientes .....	40
Figura 9 Alteração dos dados de cliente .....	41
Figura 10 Consulta a Clientes .....	41
Figura 11 Inclusão de Produtos .....	41
Figura 12 Alteração dos Dados de Produto .....	42
Figura 13 Consulta a Produtos.....	42
Figura 14 Exclusão de Produtos .....	42
Figura 15 Pedido de Clientes.....	42
Figura 16 Tela de Identificação .....	42
Figura 17 Menu do Sistema .....	43
Figura 18 Telas de auxílio .....	43
Figura 19 Relação de Clientes .....	44
Figura 20 Relação de Clientes Inativos.....	44
Figura 21 Relação de Produtos .....	44
Figura 22 <i>Layout</i> da tela Pedido de Clientes (alterado).....	53
Figura 23 <i>Layout</i> da tela Cadastramento de Vendedores.....	53
Figura 24 <i>Layout</i> do Menu do sistema (alterado) .....	54
Figura 25 <i>Layout</i> da tela de Comparação entre os cinco maiores clientes .....	54
Figura 26 <i>Layout</i> da Relação de Pedidos de Clientes .....	54

## ÍNDICE DE QUADROS

Quadro 1 Complexidade de Arquivos Lógicos Internos .....	13
Quadro 2 <i>Layout</i> do Arquivo Clientes .....	13
Quadro 3 <i>Layout</i> do Arquivo Senhas .....	15
Quadro 4 Complexidade de Entradas Externas .....	18
Quadro 5 Complexidade de Saídas Externas .....	21
Quadro 6 Complexidade da Parte de Entrada de uma Consulta Externa. ....	25
Quadro 7 Complexidade da Parte de Saída de uma Consulta Externa .....	26
Quadro 8 As 14 Características Gerais do Sistema .....	28
Quadro 9 <i>Layout</i> do Arquivo de Clientes .....	39
Quadro 10 <i>Layout</i> do Arquivo de Produtos .....	39
Quadro 11 <i>Layout</i> do Arquivo de Pedidos .....	39
Quadro 12 <i>Layout</i> do Arquivo de Senhas .....	39
Quadro 13 <i>Layout</i> do Arquivo de Histórico de Clientes .....	39
Quadro 14 Funções Identificadas no Sistema para o Exemplo .....	44
Quadro 15 Classificação de complexidade das funções identificadas .....	45
Quadro 16 Cálculo do Fator de Ajuste .....	47
Quadro 17 <i>Layout</i> do cadastro de pedidos (alterado) .....	52
Quadro 18 <i>Layout</i> do cadastro de vendedores (incluído) .....	53
Quadro 19 Modificações e inclusões de funções no sistema .....	55
Quadro 20 Funções incluídas, excluídas e modificadas no sistema .....	55

## ÍNDICE DE TABELAS

Tabela 1 Pontos de Função de Arquivos Lógicos Internos.....	14
Tabela 2 Pontos de Função de Arquivos de Interface Externa .....	15
Tabela 3 Pontos de Função de Entradas Externas.....	19
Tabela 4 Pontos de Função de Saídas Externas .....	22
Tabela 5 Pontos de Função de Consultas Externas.....	27
Tabela 6 Cálculo dos Pontos de Função não ajustados.....	47
Tabela 7 Total de PFs das funções incluídas .....	58
Tabela 8 Total de PFs das funções excluídas.....	58
Tabela 9 Total de PFs das funções alteradas – antes projeto de manutenção .....	58
Tabela 10 Total de PFs das funções alteradas – após projeto de manutenção .....	58
Tabela 11 Conversão de Pontos de Função em Linhas de Código [COCOMOII, 2000].....	62
Tabela 12 Base de dados histórico [Fernandes, 1995].....	63



# 1 INTRODUÇÃO - VISÃO GERAL DE PONTOS DE FUNÇÃO

## 1.1 Histórico de Pontos de Função

Para que um projeto de software seja bem sucedido é preciso, antes de tudo, compreender o escopo do trabalho a ser realizado e os riscos envolvidos, para então estimar e planejar os recursos necessários, as tarefas que devem ser realizadas, os marcos de referência que devem ser acompanhados, o custo e o cronograma que deve ser seguido.

Estimar esforço, custo e cronograma de projetos constitui uma das principais preocupações dos gerentes de projeto. Essas estimativas normalmente são baseadas no tamanho do software e as questões com que os gerentes se deparam são: Como medir o tamanho de um sistema? Como estimar o tamanho de um sistema antes de desenvolvê-lo?

Responder a essas perguntas é um assunto polêmico e envolve muitas controvérsias [Azevedo, 1999]. Ainda não estão claramente definidas quais métricas são mais apropriadas para medir o tamanho de um sistema. Há um debate contínuo a respeito das melhores medidas de tamanho de software e não existe, de fato, uma resposta unânime.

Uma forma bastante familiar de se medir tamanho de software é através da contagem de linhas de código, cuja unidade é LOC (*Lines Of Code* – Linhas de Código), uma *métrica orientada ao tamanho*<sup>1</sup> do software [Conte, 1986]. Trata-se de uma métrica amplamente aplicável, preferida nas organizações pela facilidade com que pode ser contada e compreendida, e por facilitar a contagem automatizada.

A contagem de linhas de código é relativamente fácil de ser obtida. Pode-se obter o tamanho de um software, por exemplo, com o resultado direto da contagem que um compilador apresenta ao compilar um programa. Todavia, é muito difícil

---

<sup>1</sup> Segundo Boehm [Boehm, 1981], uma métrica é uma forma padrão de medir um atributo do processo de desenvolvimento de *software*. As métricas fornecem informações quantitativas sobre o ambiente e o progresso de desenvolvimento do produto. Métricas orientadas ao tamanho são medidas diretas do software e do processo por meio do qual ele é desenvolvido.

prever o total de linhas de código logo no início do projeto, quando devem ser elaboradas as estimativas.

Existe muita controvérsia quanto ao uso de linhas de código como medida chave [Jones, 1996]:

- ❑ Não faz sentido comparar linhas de código escritas em linguagens diferentes, e mesmo que uma mesma linguagem tenha sido utilizada no desenvolvimento de dois programas, fica difícil fazer a comparação.
- ❑ Embora pareça ser uma métrica simples que pode ser contada algoritmicamente, não há consenso geral do que realmente constitui uma linha de código. Devemos considerar linhas de comentários ou linhas em branco? Para a maioria dos pesquisadores, linhas de comentários e linhas em branco não deveriam ser contadas, uma vez que essas servem para documentação interna, e sua presença ou ausência não afeta as funções do programa. A inclusão desses dois tipos na contagem de um sistema pode encorajar programadores a introduzir artificialmente diversas linhas nos sistemas em desenvolvimento, criando a ilusão de grande produtividade.
- ❑ Uma certa função poderia ser codificada por um programador experiente em 200 linhas de código, um inexperiente codificaria em 400 linhas de código. Como comparar os dois programas que fazem exatamente a mesma coisa, mas um tem o dobro de número de linhas de código do outro?

Outra forma de se medir tamanho do software é através da Análise de Pontos de Função (FPA – *Function Point Analysis*), uma *métrica orientada à função*<sup>2</sup>.

No final dos anos 70, Allan J. [Albrecht, 1979], por solicitação do GUIDE (Grupo de Usuários IBM), começou analisar sistemas pragmaticamente, buscando identificar fatores críticos que determinassem o tamanho de um sistema e o esforço gasto para desenvolvê-lo.

Após analisar centenas de programas, Albrecht traçou os fundamentos da *Análise de Pontos de Função*, e introduziu seus conceitos em uma conferência do GUIDE em 1979. Esses conceitos foram mais tarde refinados em uma metodologia formal e publicados para domínio público em 1984.

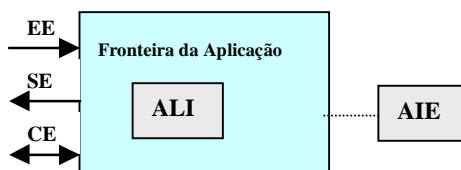
---

<sup>2</sup> *Métricas orientadas a função* são medidas do software e/ou do processo por meio do qual ele é desenvolvido que se concentram na “funcionalidade” ou “utilidade” do programa [Pressman, 1997].

Um dos principais conceitos da teoria de Análise de Pontos de Função é que as funções de um sistema devem ser contadas a partir da perspectiva do usuário e não do analista ou programador. Além disso, a contagem de Pontos de Função é independente da maneira como ele é desenvolvido e implementado. Isso significa que, dado um conjunto de requisitos de usuário, o tamanho funcional do software será o mesmo, seja ele desenvolvido em linguagem COBOL ou DB2, usando Desenvolvimento Rápido de Aplicações (RAD) ou métodos estruturados de desenvolvimento.

Pontos de função medem o tamanho funcional<sup>3</sup> de um projeto de software através da quantificação da informação da funcionalidade de processamento da aplicação. Para contar pontos de função, cinco tipos de componentes lógicos da aplicação (Figura 1) devem ser avaliados. [Dekkers, 1998].

**Figura 1 Componentes de Pontos de Função em relação à aplicação [Dekkers, 1998]**



A técnica de Análise de Pontos de Função divide as funções em dois tipos: *dados* (Arquivos Lógicos Internos – ALI e Arquivos de Interface Externa – AIE) e *transações* (Entradas Externas – EE, Saídas Externas – SE, e Consultas Externas – CE). Segundo Albrecht esses cinco tipos de funções afetam de forma distinta o tamanho de um sistema.

Em 1986 uma comunidade de usuários resolveu efetuar padronizações adicionais nas regras de contagem de Pontos de Função, formando o *IFPUG* [IFPUF, 2000] – *International Function Points Users Group* (Grupo Internacional de Usuários de Pontos de Função), um grupo formalmente constituído e sem finalidades lucrativas [Dekkers, 1998]. O IFPUG tem sido líder no estabelecimento e publicação de documentos relacionados a Pontos de Função, incluindo o *Manual de Práticas de Contagem* (CPM, atualmente na versão 4.1) [IFPUG, 1994], o *Guia para Medição de Software* (atualmente em sua versão 1.1) e diversos Estudos de Casos detalhados.

---

<sup>3</sup> Tamanho funcional é uma medida do tamanho de um software baseada em uma avaliação padronizada dos requisitos lógicos sob o ponto de vista do usuário da aplicação.

Antes da existência do IFPUG muitas das informações sobre pontos de função eram encontradas em vários artigos e livros. Os conceitos e definições eram algumas vezes vagos, tornando as regras difíceis de serem compreendidas e interpretadas, o que levava a uma variabilidade na contagem.

Desde a fundação do IFPUG as regras de contagem evoluíram e melhoraram. A consistência entre os contadores também melhorou. O Manual de Práticas de Contagem do IFPUG define o processo e as regras para contagem de pontos de função, fazendo com que o processo seja documentado para poder ser repetido, melhorado e utilizado em treinamentos. Um estudo realizado em 1994 pelo *Quality Assurance Institute* e pelo IFPUG encontrou uma variância entre contadores treinados da ordem de menos de 11 por cento [Furey, 1997].

Atualmente, o IFPUG permanece como uma organização composta de voluntários (os únicos empregados são a equipe administrativa), participando ativamente do grupo de trabalho da ISO sobre *Medidas Funcionais de Tamanho* (ISO/IEC JTC1 SC7 WG12<sup>4</sup>).

O IFPUG administra a certificação de materiais para treinamento em pontos de função e de especialistas em pontos de função (CFPS – *Certified Function Points Specialist*), mantendo seu site na Internet [IFPUG, 2000], cuidando da instrução em mensuração de software e promovendo pelo menos duas conferências anuais. Atualmente, membros de mais de 13 países participam do IFPUG servindo em comitês, ou comparecendo a conferências e treinamentos.

No Brasil, o *Brazilian Function Points Users Group* [BPFUG, 2000] é um grupo constituído com o objetivo de estimular e divulgar a utilização de métricas no desenvolvimento de sistemas, em particular a Análise de Pontos de Função. Destina-

---

<sup>4</sup> O ISO/IEC/JTC 1/SC7/Working Group 12 (WG12) administra o projeto 7.31 para Medição de Software: Medição Funcional de Software. O projeto é composto por cinco partes:

1. Definição de Conceitos
2. Avaliação da Conformidade de Métodos de Medição de Software com Relação ao padrão ISO/IEC 14143-1:1998
3. Verificação de um Método de Medição Funcional de Tamanho
4. Modelo de Referência para Medição Funcional de Tamanho
5. Determinação de Domínios Funcionais para uso com Medição Funcional de Tamanho.

A Parte 1 deste padrão, ISO/IEC 14143-1:1998 foi publicada em junho de 1998 como um padrão internacional completo e encontra-se disponível através dos organismos nacionais de padrões em cada país.

se aos profissionais interessados em aprender, praticar e divulgar o uso de métricas e de Análise de Pontos de Função.

## 1.2 Usos de Pontos de Função

Até 1984, a técnica era usada basicamente para identificar o tamanho de um sistema e auxiliar na estimativa do esforço de desenvolvimento. Atualmente a utilização de pontos de função também abrange a implantação de programas de métricas para melhorar estimativas, gerenciar a qualidade e para monitorar a produtividade, além de controlar indicadores financeiros, ou seja, melhorar todo o processo de desenvolvimento e manutenção de sistemas.

As organizações podem obter muitos benefícios ao utilizar a técnica de Análise de Pontos de Função. Em medições, a técnica pode servir como um instrumento para acompanhamento de estimativas de custo e recursos requeridos para o desenvolvimento e manutenção de software, ajudando assim na implantação de um programa de métricas.

A técnica de Análise de Pontos de Função também pode ser utilizada para auxiliar na determinação de um fator de normalização para comparação entre softwares, identificando os benefícios que um pacote de software pode oferecer, através da contagem dos pontos de função que refletem as necessidades da empresa e auxiliar na decisão entre a compra de um pacote ou o desenvolvimento do aplicativo *in house*.

Pontos de função também ajudam as organizações a normalizar dados como custo, esforço, duração, defeitos, equipe, etc.. Ao basear-se uma análise apenas no número de defeitos, não se pode objetivamente comparar a qualidade de um sistema com a qualidade de outros. Porém, utilizando pontos de função em combinação com dados de defeitos, pode-se comparar qualidade, em termos de densidade de defeitos do software, entre aplicações e linguagens.

Supondo dois sistemas: um sistema A que contenha 640 defeitos e 800 pontos de função; sua densidade de defeitos é da ordem de 0.8; e um sistema B que contenha 700 defeitos e 2.00 pontos de função; sua densidade de defeitos é da ordem de 0.32. Ao olharmos na raiz do número de defeitos, o sistema B possui mais defeitos. Porém

quando usamos pontos por função para normalizar o tamanho, a densidade de defeitos do sistema B é de fato muito mais baixa e a qualidade global é maior.

### 1.3 Visão Geral do Processo de Análise de Pontos de Função

De acordo com a finalidade a que se destina, pode-se realizar três tipos de contagem: a contagem de pontos de função: a contagem em *projetos de desenvolvimento* de software, a contagem de pontos de função de uma *aplicação já desenvolvida* e implantada, e a contagem de pontos de função em *projetos de manutenção* de software.

Em *projetos de desenvolvimento*, a técnica é utilizada para dimensionar o tamanho provável de uma nova aplicação. O objetivo desse cálculo é quantificar as funções solicitadas, identificadas no modelo lógico do sistema, e que deverão ser entregues ao usuário pela nova aplicação, para que se seja possível obter estimativas de recursos, tempo e pessoal necessários ao desenvolvimento do sistema.

Quando o projeto termina e a *aplicação já se encontra implementada*, ou até mesmo implantada, pode-se obter informações mais reais e físicas do mesmo. Esse cálculo é também referenciado como cálculo de pontos de função instalados.

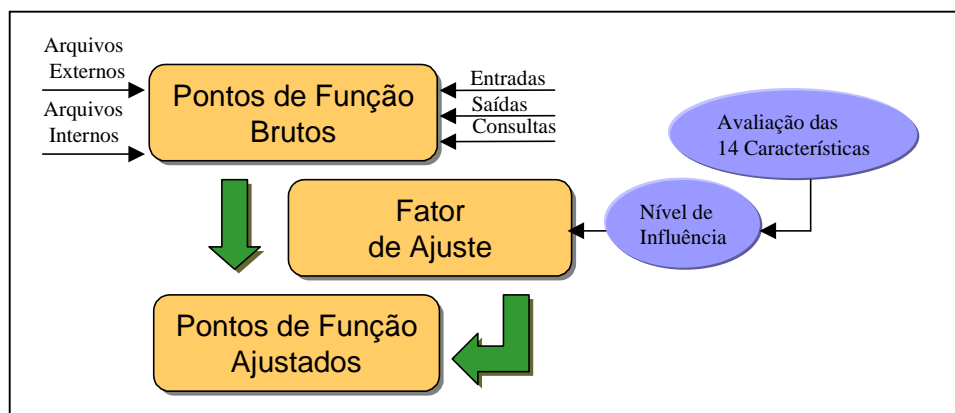
Em *projetos de manutenção*, o cálculo é utilizado para dimensionar o tamanho da modificação que deve ser efetuada em uma aplicação já existente. Em geral, a modificação é uma melhoria ou adição de funcionalidade a uma aplicação existente.

Segundo o IFPUG, o processo de Análise de Pontos de Função (Figura 2) é composto de três etapas: na primeira etapa é realizada a contagem de *Pontos de Função Brutos ou Não Ajustados*, em seguida determina-se o *Fator de Ajuste*, e na última etapa, determina-se os *Pontos de Função Ajustados*.

Os pontos de função brutos, contados na primeira etapa do processo de Análise de Pontos de Função, refletem especificamente a contagem da funcionalidade provida ao usuário pela aplicação. Uma aplicação, vista sob a ótica do usuário, é um conjunto de funções ou atividades de negócio que o beneficiam na realização de suas tarefas.

As funções são identificadas e determinadas após a definição a *fronteira* que separa o projeto ou aplicação, que está sendo contado, de aplicações externas, ou seja, de outros sistemas da organização. A determinação da fronteira é melhor explicada no Capítulo 2.

**Figura 2 Visão Geral do Processo de Análise de Pontos de Função**



As funções identificadas dentro da fronteira da aplicação são classificadas em *funções do tipo dados* e *funções do tipo transação*.

*Funções do tipo dados* representam a funcionalidade provida ao usuário para atender requisitos externos e internos referentes a dados. Funções do tipo dados podem ser *Arquivos Lógicos Internos* ou *Arquivos de Interface Externa*.

Um *Arquivo Lógico Interno* (ALI) é um grupo de dados logicamente relacionados, ou informações de controle, identificados e modificados pelo usuário e mantidos dentro das fronteiras da aplicação que está sendo contada, como por exemplo, um arquivo de clientes.

Um *Arquivo de Interface Externa* (AIE) é um grupo de dados logicamente relacionados, ou informações de controle, utilizados no sistema que está sendo analisado, mas que não são mantidos dentro da fronteira da aplicação que está sendo contada. Um exemplo de AIE poderia ser informações de acesso ao sistema principal, que se encontram em outra aplicação, num suposto arquivo *senhas*. Consequentemente, esse AIE será mantido e alterado por outra aplicação, onde será identificado como ALI.

*Funções do tipo transação* representam a funcionalidade da aplicação provida ao usuário pelo processamento de dados. Funções transacionais são definidas como *Entradas Externas* (EE), *Saídas Externas* (SE) e *Consultas Externas* (CE).

Uma *Entrada Externa* (EE) processa dados ou informações de controle geradas por uma fonte externa à aplicação que está sendo contada, o próprio usuário, por exemplo. São transações que efetuam manutenção nos dados armazenados no sistema, tais como *inclusões*, *alterações* e *exclusões*.

Uma *Saída Externa* (SE) fornece dados ou informações de controle para fora da aplicação que está sendo contada. São transações que extraem informações do sistema para outros aplicativos, como uma função cálculo da folha de pagamento baseado na soma dos totais de horas trabalhadas de funcionários, por exemplo.

Uma *Consulta Externa* (CE) é uma transação que combina transações de entrada e saída, resultando em recuperação de dados. É o caso de um relatório com os dados de um cliente.

Depois de identificadas, cada uma das funções deve ser classificada quanto à sua *complexidade funcional* relativa: *simples*, *média* ou *complexa*. Existem regras e tabelas específicas para auxiliar essa classificação. Em seguida, calcula-se os *pontos de função brutos* através da aplicação de pesos de acordo com tabelas específicas para cada tipo de função.

O *fator de ajuste* determinado na segunda etapa do processo de Análise de Pontos de Função representa a funcionalidade geral da aplicação que é provida ao usuário. O fator de ajuste é composto por 14 *Características Gerais do Sistema* que avaliam a funcionalidade geral da aplicação. Cada característica possui sua própria descrição que auxilia a determinar seu grau de influência. Os níveis de influência variam de zero (nenhuma influência) até cinco (forte influência).

O cálculo dos pontos de função ajustados, efetuado na terceira etapa do processo de Análise de Pontos de Função é, basicamente, a aplicação do fator de ajuste ao resultado apurado na primeira etapa. Os pontos de função ajustados são calculados utilizando-se fórmulas específicas para projeto de desenvolvimento, para projetos de manutenção e para cálculo de aplicação já desenvolvida ou instalada.

## 1.4 Um Exemplo Simplificado

O exemplo a seguir ilustra a contagem de pontos de função não ajustados para um projeto e desenvolvimento de software, sem recorrer ao *Manual Prático de Contagem de Pontos de Função* [IFPUG, 1994] para determinar a complexidade (baixa, média ou alta) dos componentes ou funções identificados no sistema.

---

A seguir são apresentados os requisitos funcionais identificados pelo usuário.



Desenvolver uma aplicação para armazenar e efetuar manutenção em registros de empregados com os seguintes campos de dados: *nome, número, posição, endereço (rua), cidade, estado, código postal, data de nascimento, número do telefone, e data da última alteração dos dados do empregado*. A aplicação deve proporcionar meios para adicionar novos empregados, atualizar informações, excluir empregados, e combinar dados de dois empregados. A aplicação deve prover um relatório semanal em papel que liste quais informações de empregados foram modificadas na semana anterior. A aplicação deve prover meios de pesquisar dados de empregados. Nenhum dado fora da aplicação é referenciado.

Baseado nos requisitos do usuário, os componentes ou funções identificados e que devem ser contados são:

- ❑ **Um Arquivo Lógico Interno** (ALI) para dados de empregados porque trata-se de uma entidade lógica (ou grupo de dados) mantida e modificada pela aplicação. Baseado nas avaliações dos elementos de dados e tipos de registros lógicos (contidos no manual prático de contagem), esse ALI seria considerado baixo. Sua pontuação é 07 pontos de função.
- ❑ **Quatro Entradas Externas** (EE), representadas pelas funções de inclusão, atualização, exclusão, e combinação de registros de empregados. Assumindo que cada uma possui complexidade baixa (cada uma requer apenas um arquivo lógico referenciado – ALI empregados – e menos de 16 elementos de dados). Cada EE deve ser pontuada com três pontos, num total de 12 pontos de função.
- ❑ **Uma Saída Externa** (SE), representada pelo relatório semanal solicitado, categorizada como saída externa de complexidade baixa (um arquivo lógico referenciado – ALI empregados – e menos de 20 elementos de dados). Sua pontuação é 04 pontos de função.
- ❑ **Uma Consulta Externa** (CE), representada pela função de pesquisa de dados de empregados. Baseado no número de elementos de dados (menos de 20) e no número de arquivos lógicos referenciados (ALI empregados), essa CE seria classificada baixa e pontuada com 03 pontos de função.

O total de pontos de função obtidos na contagem é de 26 pontos de função não ajustados. Pontos de função brutos podem ser utilizados para cálculo de estimativas de projeto através do uso do modelo COCOMO (Capítulo 6)

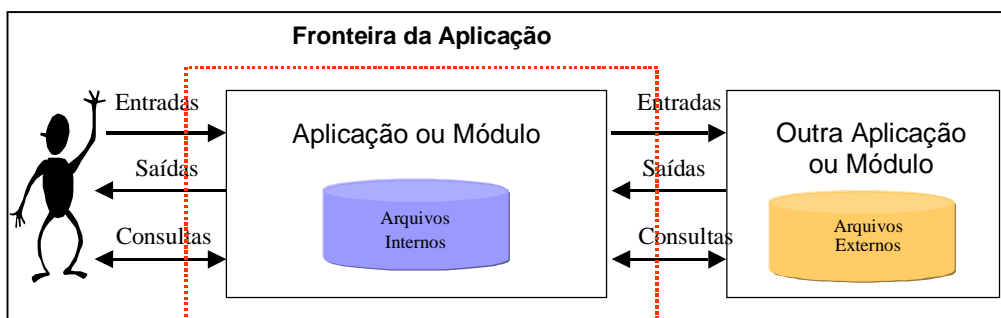
---

## 2 REGRAS PARA IDENTIFICAÇÃO E CONTAGEM DE FUNÇÕES

### 2.1 Identificação da Fronteira da Aplicação

A fronteira da aplicação (Figura 3) é a linha que separa o projeto ou aplicação que está sendo contado de outras aplicações ou sistemas da organização. A determinação da fronteira é importante para que se possa estabelecer o *escopo* do software que está sendo medido, a *propriedade dos dados*, e a *propriedade das funções* requeridos pela contagem (se os dados ou função pertencem à aplicação que está sendo contada ou à outra aplicação).

Figura 3 Identificação da Fronteira da Aplicação



As seguintes regras podem ser aplicadas para auxiliar o estabelecimento das fronteiras:

- ❑ A fronteira é determinada através do ponto de vista do usuário. O foco deve estar no que pode ser compreendido e descrito pelo usuário.
- ❑ A fronteira entre aplicações relacionadas é baseada em funções empresariais separadas, da forma como são vistas pelos usuários, e não através de necessidades tecnológicas.
- ❑ Em projetos de manutenção, a fronteira inicial deve estar em conformidade com a fronteira já estabelecida para a aplicação ou aplicações que estão sendo modificadas.

Observação: Pode haver mais de uma aplicação incluída no escopo de um único projeto. Nesse caso, todas as fronteiras devem ser identificadas.

Pode-se usar as especificações ou o diagrama de fluxo de dados do sistema e desenhar as fronteiras ao redor da aplicação contada, para realçar as funções internas à aplicação.

No processo de análise de pontos de função, os seguintes itens devem estar documentados: a fronteira identificada, o propósito da contagem e outras suposições nas quais a contagem esteja baseada.

## 2.2 Regras de Contagem para *Arquivos Lógicos Internos* (ALI)

Um Arquivo Lógico Interno é um grupo de dados relacionados logicamente (ou informações de controle), identificado pelo usuário, utilizado, mantido (armazenado) e alterado (modificado) *dentro* das fronteiras da aplicação.

A expressão “*grupo de dados logicamente relacionados*” quer dizer que a manifestação lógica dos dados é mais importante que a implementação física. Isto acontece quando um grupo de dados deixa de ser visto como elementos individuais de dados (campos de uma tabela) para satisfazer requisitos da aplicação. Uma base de dados, por exemplo, pode conter várias tabelas; cada qual possui regras específicas. Pode ser o caso em que cada tabela seja um arquivo lógico, porém não necessariamente.

Arquivos Lógicos Internos podem ser referenciados através de transações de entrada, saída e consulta da aplicação analisada, ou ainda, através de transações de outras aplicações.

Para identificar Arquivos Lógicos Internos, todas as questões seguintes devem ser respondidas afirmativamente:

- ✓ O grupo de dados *reflete os requisitos funcionais* definidos pelo usuário?
- ✓ O grupo de dados é *armazenado dentro da fronteira* da aplicação?
- ✓ O grupo de dados é *alterado dentro da fronteira* da aplicação?
- ✓ O grupo de dados é *alterado através de processos elementares*?
- ✓ O grupo de dados *não foi contado* como Arquivo de Interface Externa nessa aplicação?

Grupo de dados ou informações de controle é um conjunto lógico de dados que deve ser identificado pelo usuário e deve refletir todos os requisitos funcionais

definidos por ele. Além disso, o grupo de dados deve ser armazenado e alterado (modificado) dentro da fronteira da aplicação, e *não* pode ser contado também como um Arquivo de Interface Externa (AIE) na mesma aplicação.

Podem ser considerados Arquivos Lógicos Internos: cadastro de clientes, cadastro de produtos, cadastro de funcionários, arquivo de dependentes, arquivo de controle de acesso à aplicação, etc.

Não são considerados Arquivos Lógicos Internos: arquivos temporários, arquivos de trabalho, arquivos de classificação, arquivos incluídos por motivos de tecnologia e arquivos de índices alternativos.

### 2.2.1 Complexidade de Arquivo Lógico Interno

A complexidade de um Arquivo Lógico Interno é calculada a partir da quantidade de *Dados Elementares Referenciados* e de *Registros Lógicos Referenciados*.

*Dado Elementar Referenciado* (DER) é um campo, reconhecido pelo usuário, que está presente em um Arquivo Lógico Interno (ALI) ou em um Arquivo de Interface Externa (AIE). Suas regras de contagem são:

- ❑ Considerar *um* DER para cada campo, reconhecido pelo usuário, presente em cada ALI.
- ❑ Considerar as seguintes implementações técnicas como um DER, para um conjunto completo de campos:
  - Campos que aparecem mais de uma vez em um ALI por causa de tecnologia ou implementação. Exemplo: se um código de funcionário aparecer duas vezes, inicialmente no registro de funcionário e, posteriormente, como chave estrangeira em um registro dependente, contar somente um DER.
  - Campos repetitivos que são idênticos em formato e existem para permitir múltiplas ocorrências de um mesmo dado. Exemplo: se um ALI possui 12 campos de venda mensal e um total anual, considerar apenas dois itens de dados, um para o valor mensal e o segundo para o total anual.

*Registro Lógico Referenciado* (RLR) é um subgrupo de elementos de dados, reconhecido pelo usuário, dentro de um Arquivo Lógico Interno (ALI) ou de um Arquivo de Interface Externa (AIE). Suas regras de contagem são:

- ❑ Caso não haja subgrupo de informações (referentes à outro ALI), contar um RLR para cada ALI.
- ❑ Contar um RLR para cada subgrupo de dados de um ALI independentemente de ser o subgrupo opcional ou mandatório.

O Quadro 1 deve ser utilizado para a definição da complexidade de cada ALI.

**Quadro 1 Complexidade de Arquivos Lógicos Internos**

	<b>1 a 19 DER</b>	<b>20 a 50 DER</b>	<b>51 ou mais DER</b>
1 RLR	Simples	Simples	Média
2 a 5 RLR	Simples	Média	Complexa
6 ou mais RLR	Média	Complexa	Complexa

Exemplo: O usuário deseja manter as informações referentes aos clientes da empresa. Ele relacionou 15 itens de dados (Quadro 2) que deseja utilizar e que devem ser referenciados por funções de inclusão, consulta e emissão de relatório dos clientes da empresa.

O cadastro de clientes foi identificado como Arquivo Lógico Interno porque:

- ✓ É um grupo de dados espelha os requerimentos definidos pelo usuário, é mantido e alterado dentro da fronteira da aplicação, através de processos elementares (inclusão, consulta e emissão de relatório), e não foi contado como Arquivo de Interface Externa na mesma aplicação.

**Quadro 2 Layout do Arquivo Clientes**

<b>Clientes</b>		
<b>Item de dados</b>	<b>Tamanho</b>	<b>Tipo</b>
Código-cliente	6	Numérico
Nome-cliente	12	Alfa
Razão-social	30	Alfa
Tipo-cliente	2	Numérico
Território	3	Alfa
Endereço	30	Alfa
Bairro	12	Alfa
CEP	8	Numérico
Cidade	12	Alfa
CGC	12	Numérico
Inscrição-estadual	12	Numérico
Contato-1	20	Alfa
Telefone-1	10	Numérico
Contato-2	20	Alfa
Telefone-2	10	Numérico

**Contagem:**

*Cadastro de clientes*

DER: 15 (itens de dados)

RLR: 1 (arquivo referenciado: clientes)

Grau da Função: Simples

⇒ função brutos correspondentes ao tipo de ALI identificado (no exemplo, o tipo é simples), utiliza-se a Tabela 1. Portanto, o cadastro de clientes,

analisado e identificado como um Arquivo Lógico Interno, possui 7 pontos de função não ajustados ou brutos.

**Tabela 1 Pontos de Função de Arquivos Lógicos Internos**

<b>ALI - Grau de Complexidade Funcional</b>	
Simple	7
Média	10
Complexa	15

### 2.3 Regras de Contagem para *Arquivos de Interface Externa (AIE)*

Um Arquivo de Interface Externa (AIE) é, segundo a visão do usuário, um grupo de dados relacionados logicamente (ou informações de controle necessárias à aplicação), mantido e alterado *fora* da fronteira da aplicação. Isto significa que um AIE contado em uma aplicação deve estar classificado como Arquivo Lógico Interno (ALI) em outra aplicação.

Supondo uma situação em que uma transação do sistema que está sendo contado necessite acessar uma informação que se encontra em outra aplicação, ou seja, essa informação está em um grupo de dados reconhecido como um ALI em outro sistema. Esse grupo de dados ou informações externas caracteriza um AIE na aplicação que está sendo contada. Porém não são considerados AIE informações de outras aplicações que geram inclusão, alteração ou exclusão de registros. Nesses casos pode se tratar de Entradas Externas.

Para identificar Arquivos de Interface Externa, todas as questões seguintes devem ser respondidas afirmativamente:

- ✓ O grupo de dados *reflete os requisitos funcionais* definidos pelo usuário?
- ✓ O grupo de dados é *armazenado fora da fronteira* da aplicação?
- ✓ O grupo de dados *não é alterado* pela aplicação que está sendo contada?
- ✓ O grupo de dados é *contado como ALI* na outra aplicação?
- ✓ O grupo de dados *não foi contado* como ALI nessa aplicação?

O grupo de dados ou informação de controle é um grupo lógico de dados que deve ser identificado pelo usuário e refletir os requisitos funcionais definidos por ele. Além disso, o grupo de dados deve ser *externo* à aplicação que está sendo contada

(armazenado e alterado – modificado – fora da fronteira da aplicação), e não pode ser contado também como um Arquivo de Lógico Interno (ALI) na mesma aplicação.

### 2.3.1 Complexidade de Arquivo de Interface Externa

A complexidade de um Arquivo de Interface Externa também é calculada em função da quantidade de *Dados Elementares Referenciados* e de *Registros Lógicos Referenciados*.

As regras para contagem para *Dados Elementares Referenciados* e de *Registros Lógicos Referenciados* são as mesmas aplicadas em Arquivos Lógicos Internos, assim como o quadro para a determinação da complexidade.

Exemplo: O desenvolvimento de um novo aplicativo foi solicitado pela empresa, e para ser acessado, deve primeiramente verificar a segurança de acesso aos dados. O cadastro de senhas (Quadro 3) é único e já existe para outros sistemas da empresa.

Essa função foi identificada como Arquivo de Interface Externa na aplicação que vai ser desenvolvida porque:

- ✓ O grupo de dados espelha os requisitos funcionais definidos pelo usuário, é externo à aplicação (mantido e alterado fora da fronteira da aplicação) e foi contado como ALI por outra aplicação.

**Quadro 3 Layout do Arquivo Senhas**

<b>Senhas</b>		
<b>Item de dados</b>	<b>Tamanho</b>	<b>Tipo</b>
Código-usuário	6	Numérico
Senha	6	Alfa
Data-validade	8	Numérico
Data-último-acesso	8	Numérico
Hora-último-acesso	8	Numérico

**Contagem:**

*Arquivo de Senhas*

DER: 5 (itens de dados)

RLR: 1 (arquivo referenciado:

as)

Grau da Função: Simples

Para o cálculo do número de pontos de função brutos correspondentes ao tipo de AIE identificado como simples, utiliza-se a Tabela 2. Portanto, o Arquivo de Interface Externa identificado e analisado possui 5 pontos de função não ajustados ou brutos.

**Tabela 2 Pontos de Função de Arquivos de Interface Externa**

<b>AIE - Grau de Complexidade Funcional</b>	
Simples	5
Média	7
Complexa	10

## 2.4 Regras de Contagem para *Entradas Externas* (EE)

Uma Entrada Externa (EE) é qualquer função ou transação que leva dados que vêm de fora da fronteira para dentro da aplicação. Geralmente esses dados são usados para atualizar informações internas da aplicação. A Entrada Externa é por si só um processo elementar, ou seja, um processo que deixa a aplicação em um estado consistente; uma vez que o processo terminou não existe mais nada a ser feito, nada mais deve ocorrer.

Para identificar as Entradas Externas, deve-se analisar *dados* ou *informações de controle* que estão sendo trazidas para dentro da fronteira da aplicação.

As questões seguintes devem ser aplicadas aos *dados* para serem considerados uma Entrada Externa, e devem ser respondidas afirmativamente:

- ✓ O dado *é recebido de fora da fronteira* da aplicação.
- ✓ O dado em um Arquivo Lógico Interno *é referenciado através de um processo elementar* da aplicação.
- ✓ O processo *é a menor unidade de atividade* com significado para o usuário final.
- ✓ O processo *é autocontido e deixa em estado consistente* o negócio que a aplicação informatiza.

As questões seguintes devem ser aplicadas às *informações de controle* para serem consideradas uma Entrada Externa, e devem ser respondidas afirmativamente:

- ✓ As informações de controle *são recebidas de fora das fronteiras* da aplicação.
- ✓ A informação *é especificada pelo usuário* para garantir aderência às regras do negócio.

Com relação à função uma vez identificada, uma das seguintes opções deve ser verdadeira:

- ✓ A lógica de processamento *é diferente de outras lógicas* processadas nas demais Entradas Externas da aplicação.
- ✓ Os dados elementares identificados *são diferentes das demais Entradas Externas* da aplicação.



Podem ser consideradas Entradas Externas: inclusão, alteração e exclusão de registros em cadastros, e ainda, transações vindas de outra aplicação. *Não* são consideradas Entradas Externas: menus de navegação, telas de consulta, telas de auxílio, telas de *logon*, parte da entrada de uma consulta. A funcionalidade proporcionada por menus, telas de auxílio, etc., é considerada como parte das Características Gerais do Sistema.

#### 2.4.1 Complexidade de Entradas Externas

A complexidade das Entradas, Saídas e Consultas Externas é analisada segundo a quantidade de *Dados Elementares Referenciados* e de *Arquivos Lógicos Referenciados*.

*Dado Elementar Referenciado* (DER) é um único campo, não recursivo, identificado pelo usuário, mantido em um Arquivo Lógico Interno pela Entrada Externa. As regras para contagem para *Dados Elementares Referenciados* são:

- ❑ Contar um DER para cada campo não recursivo, identificado pelo usuário, e mantido em um Arquivo Lógico Interno por uma Entrada Externa.
- ❑ Contar um DER para cada campo pertencente a um Arquivo Lógico Interno que não é digitado pelo usuário, mas é mantido por uma Entrada Externa.
- ❑ Contar as seguintes técnicas de implementação físicas como um único DER para o grupo de campos:
  - Um campo lógico que é armazenado fisicamente em múltiplos campos, mas é requerido pelo usuário como uma peça única de informação. Exemplo: uma data que é fisicamente armazenada em múltiplos campos, mas que o usuário necessita como uma única peça de informação, é contado como um único DER.
  - Campos que aparecem mais de uma vez em um ALI por necessidade da técnica ou tecnologia de implementação deve ser contados apenas uma única vez. Exemplo: se um código de cliente aparecer duas vezes em um ALI, primeiro como chave primária de acesso ao registro de funcionário e depois como chave externa do registro de dependente, contar como um DER somente.
  - Campos que indicam condição de erro durante o processamento ou confirmação de que o processo está completo.

- Contar um único DER para linhas de comando ou teclas de função que provejam a ação a ser tomada pela Entrada Externa.
- Em tela de atualização só contar os campos que possam sofrer atualização.
- Em telas de exclusão contar somente os campos-chave.
- Contar um DER quando uma ou mais mensagens de erro informarem ao usuário, através de um campo ou área da tela, que uma Entrada Externa não pode ser processada por erro de edição, erro de validação, ou ainda, houve uma mensagem de confirmação.

*Arquivo Lógico Referenciado* (ALR) é um ALI, ou um AIE, lido ou mantido por um tipo de função. As regras para contagem para *Dados Elementares Referenciados* são:

- ❑ Contar um ALR para cada ALI mantido.
- ❑ Contar um ALR para cada ALI ou AIE lido durante o processamento da Entrada Externa.
- ❑ Contar um ALR para cada ALI que é lido e mantido por uma Entrada Externa.
- ❑ Contar um ALR caso haja acesso a arquivo de mensagens de erro.

O Quadro 4 deve ser utilizado para a definição da complexidade de cada EE.

**Quadro 4 Complexidade de Entradas Externas**

	<b>1 A 4 DER</b>	<b>5 A 15 DER</b>	<b>16 OU MAIS DER</b>
0 ou 1 ALR	Simples	Simples	Média
2 ALR	Simples	Média	Complexa
3 ou mais ALR	Média	Complexa	Complexa

Exemplo: O usuário definiu como parte das funções do aplicativo solicitado, uma tela (Figura 4) para cadastramento de dados de clientes. Deve ser prevista uma validação de cada campo, e, em caso de erro, uma mensagem deve ser mostrada em um campo específico.

A função definida pelo usuário para cadastramento de dados de clientes foi considerada uma Entrada Externa porque:

- ✓ Os dados serão digitados pelo usuário (fora da fronteira da aplicação), tratando-se de um processo elementar, ou seja, a tela fará o cadastramento. O processo é a menor unidade de atividade com significado para o usuário final, é autocontido, deixa o negócio em estado consistente, nada faz crer que outro processo também efetue o cadastramento e também não existe indícios de que outra Entrada Externa processe os mesmos dados de forma semelhante.

**Figura 4 Layout da Tela de Inclusão dos Dados de Clientes**

Inclusão de Clientes	
Código [ ]	
Nome [ ]	Razão Social [ ]
Tipo [ ]	Território [ ]
CGC [ ]	Inscrição Estadual [ ]
Endereço [ ]	Bairro [ ]
CEP [ ]	Cidade [ ]
Contato-1 [ ]	Telefone [ ]
Contato-2 [ ]	Telefone [ ]

Arquivos referenciados: Clientes

Dados: Todos do arquivo

Regras: Enviar mensagem de erro caso o cliente já exista ou caso o CGC seja inválido.

#### **Contagem:**

*Inclusão de Clientes*

DER: 16 (15 campos de dados, mensagens de erro)

ALR: 1 (arquivo referenciado: clientes)

Grau da Função: Média

De acordo com a Tabela 3, a cada tipo de Entrada Externa (simples, média ou complexa) é atribuído um número de pontos de função brutos. Portanto, a tela de inclusão de dados de clientes, identificada e analisada como uma Entrada Externa, possui 5 pontos de função não ajustados ou brutos.

**Tabela 3 Pontos de Função de Entradas Externas**

<b>EE - Grau de Complexidade Funcional</b>	
Simples	3
Média	4
Complexa	6

## **2.5 Regras de Contagem para Saídas Externas (SE)**

Uma Saída Externa (SE) é um processo elementar que gera ou formata dados ou informações de controle, levando-os *para fora* da fronteira da aplicação. Apesar de a maioria das Saídas Externas estarem relacionadas com relatórios em papel, não se deve considerar isso uma regra. Saídas Externas podem ser mostradas em monitores de vídeo, terminais, etc.

Para identificar uma Saída Externa, deve-se observar os dados ou informações de controle que estão sendo enviados para fora da fronteira e as seguintes regras devem ser respondidas afirmativamente.

- ✓ A função ou transação envia dados, ou informações de controle, para fora da fronteira da aplicação.
- ✓ O dado, ou informação de controle, é enviado através de um processo elementar.
- ✓ A função é a menor unidade de atividade com significado para o usuário.

Com relação ao processo identificado, uma das seguintes opções deve ser verdadeira:

- ✓ A lógica de processamento é diferente de lógicas processadas em outras Saídas Externas da aplicação.
- ✓ Os dados elementares identificados são diferentes dos dados de outras Saídas Externas da aplicação.

Uma SE contém dados derivados. Dados derivados são informações que requerem processamento complementar ao dado acessado em um ALI ou em um AIE. Exemplo: Supondo se uma aplicação deve calcular o pagamento mensal de um funcionário e os dados número de horas trabalhadas e o valor hora armazenados em um ALI. Será necessário um processamento complementar: a multiplicação dos dois campos, para encontrar o valor do pagamento mensal, sendo esse, portanto, um dado derivado.

São exemplos de Saídas Externas: relatório com relação de clientes, transferência de dados para outra aplicação, fatura de um cliente. *Não* são considerados SE: menus de navegação, telas de auxílio, telas de consulta, consultas *ad-hoc*.

### 2.5.1 Complexidade de Saídas Externas

Devemos determinar a complexidade funcional relativa de cada Saída Externa baseando-se no número de *Dados Elementares Referenciados* e de *Arquivos Lógicos Referenciados*.

*Dado Elementar Referenciado* (DER) é um único campo, não recursivo, identificado pelo usuário, e que aparece em uma Saída Externa. Suas regras de contagem são:

- ❑ Contar um DER para cada campo não recursivo identificado pelo usuário, que apareça em uma Saída Externa. Exemplo: contar um DER para cada sumário ou campo de total que apareça em uma SE.
- ❑ *Não* contar literais como DER. Literais incluem: o nome do relatório, nome da tela, cabeçalhos de colunas e nome de campos.
- ❑ *Não* contar número de páginas ou campos automáticos do sistema. Exemplo: número de página, comandos de paginação como Anterior, Próximo ou seta de paginação em uma aplicação *Windows*, campo de data ou hora.
- ❑ Contar as seguintes implementações físicas como um único DER:
  - Um campo lógico que é armazenado como múltiplos campos, mas é requerido pelo usuário como uma única informação. Exemplo: uma data que é armazenada separada em três campos diferentes como dia, mês e ano.
  - Cada tipo de *label* e cada tipo equivalente de número (valor) em um gráfico de saída. Exemplo: em um gráfico de pizza teríamos 2 (dois) DERs: um para designar a categoria (descrição da partição) e um para a percentagem de cada partição.
  - Informação de texto que poderia ser uma única palavra, sentença ou frase. Exemplo: uma mensagem que é incluída em um relatório para indicar por que uma transação de inclusão de produto não foi completada com sucesso.

*Arquivo Lógico Referenciado* (ALR) é um arquivo lido pela lógica de processamento da Saída Externa. A regra de contagem é:

- ❑ Contar um arquivo lógico referenciado para cada ALI, ou AIE, lido durante o processamento da Saída Externa (SE)

O Quadro 5 deve ser utilizado para a definição da complexidade de cada SE.

**Quadro 5 Complexidade de Saídas Externas**

	<b>1 A 5 DER</b>	<b>6 A 19 DER</b>	<b>20 OU MAIS DER</b>
0 ou 1 RLR	Simples	Simples	Média
2 A 3 RLR	Simples	Média	Complexa
4 ou mais RLR	Média	Complexa	Complexa

**Exemplo:** O usuário solicitou, como parte das funcionalidades do aplicativo, o desenvolvimento de um relatório (Figura 5) com os dados de clientes. Um total de nove campos deve aparecer no relatório. O nome do relatório deverá ser *Relação de Clientes*; cada campo deve possuir seu cabeçalho, apresentado como colunas de informação. As páginas devem ser numeradas e o total de clientes cadastrados deve ser mostrado no final do relatório.

A relação de dados de clientes foi considerada uma Saída Externa porque:

- ✓ Os dados serão impressos em papel, portanto serão enviados para fora da fronteira da aplicação. Os dados serão enviados através de um processo elementar, e constitui a menor unidade de atividade com significado para o usuário. O processo é autocontido e deixa o negócio que a aplicação informatiza em estado consistente. Nada faz crer que outro processo de Saída Externa crie um relatório similar, nem existe outra Saída Externa que processe os mesmo dados.

**Figura 5 Layout do Relatório de Dados de Clientes**

<b>Relação de Clientes</b>				Data __/__/__			Página __	
Código	Nome	Razão Social	Tipo	Território	CGC	Inscrição Est.	Contato	Telefone
.....	.....	.....	.....	.....	.....	.....	.....	.....
.....	.....	.....	.....	.....	.....	.....	.....	.....
Total de Clientes __								

**Contagem:**

Somente o arquivo de clientes será acessado, portanto ALR = 1.

O arquivo de clientes possui 15 campos, mas somente 9 serão incluídos no relatório. Literais como nome de relatório, cabeçalhos dos campos ou colunas, datas do relatório e numeração de página não devem ser considerados como DER. O campo *total de clientes* deve ser considerado como informação, portanto o total de DER = 10.

*Relação de Clientes*

DER: 10 (9 campos e total de clientes)

ALR: 1 (Arquivo referenciado: clientes)

Grau da Função: Simples

De acordo com a Tabela 4, para cada tipo de Saída Externa (simples, média ou complexa) é atribuído um número de pontos de função brutos. Portanto, o total de pontos de função da função Relação de Clientes é 4.

**Tabela 4 Pontos de Função de Saídas Externas**

SE - Grau de Complexidade Funcional	
Simples	4
Média	5
Complexa	7

## 2.6 Regras de Contagem para Consultas Externas (CE)

Uma Consulta Externa (CE) é um processo elementar composto por uma combinação de duas partes: entrada e saída, que resulta em recuperação de informação. Uma consulta não altera ou atualiza qualquer dado interno da aplicação, simplesmente respondendo a um pedido de recuperação de dados. A parte de saída, ou seja, a resposta a uma consulta, *não* contém dados derivados.

Para identificar uma Consulta Externa, deve-se observar onde ocorrem as solicitações de recuperação de informação na aplicação. As questões seguintes devem ser respondidas afirmativamente.

- ✓ A solicitação de consulta deve “entrar” na aplicação.
- ✓ Os resultados de saída devem ser disponibilizados para fora das fronteiras da aplicação.
- ✓ O processo resulta em recuperação de dados.
- ✓ Os dados recuperados *não* contém dados derivados.
- ✓ A solicitação de consulta e os resultados fornecidos compõem um processo que é a menor unidade de atividade com significado para o usuário.
- ✓ O processo elementar é autocontido e mantém a aplicação em estado consistente.
- ✓ O processo *não* atualiza um Arquivo Lógico Interno (ALI).

Uma vez identificada a consulta, uma das seguintes regras deve ser verdadeira:

- ✓ A lógica do processo na parte de entrada ou na parte de saída é única quando comparada com outras lógicas de Consultas Externas da mesma aplicação.
- ✓ Os elementos de dados que compõem a parte de entrada ou a parte de saída são diferentes de outras consultas externas da mesma aplicação.

São exemplos de Consultas Externas: consulta ao cadastro de clientes, consultas implícitas, telas de *logon* com funções de segurança, telas de menu se solicitadas pelo usuário, tela de menu que provê seleção de tela e consulta de dados. *Não* são consideradas CE: menus de navegação, telas de consulta, tela de menu que só prove seleção de telas, tela de *logon* que só provê segurança da aplicação.

### 2.6.1 Complexidade de Consulta Externa

A determinação da complexidade funcional de cada Consulta Externa é feita baseando-se no número de *Arquivos Lógicos Referenciados* e no número de *Dados Elementares Referenciados* para a parte de *entrada* e para a parte de *saída*.

Deve-se considerar a maior das duas complexidades funcionais (entrada e saída) da consulta como a complexidade final da consulta.

*Dado Elementar Referenciado* (DER), para a *parte de entrada* da consulta externa, é um campo, não recursivo, identificado pelo usuário, e que aparece em uma Consulta Externa. As regras para contagem de DER da parte de entrada são:

- ❑ Contar um DER para cada campo não recursivo que aparece na parte de entrada de uma Consulta Externa.
- ❑ Contar um DER para cada campo que especifica o critério de seleção de dados.
- ❑ Contar as seguintes técnicas de implementação física como um único DER para todo o grupo de campos:
  - Campos utilizados para indicar que o processamento foi realizado com sucesso.
  - Campos que permitam a capacidade de especificar que a consulta externa deve ser executada. Exemplo: contar como um DER o botão de OK que deve ser apertado pelo usuário para efetivar a consulta.
- ❑ Contar um DER quando uma ou mais mensagens de erro informarem ao usuário que a consulta não foi efetivada por erro de edição ou validação, ou se ainda uma mensagem de confirmação ocorrer.

*Arquivo Lógico Referenciado* (ALR), para a *parte de entrada* da consulta externa, é um arquivo lido quando a consulta externa é processada. As regras de contagem de ALR para a parte de entrada são:

- ❑ Contar a quantidade de Arquivos Lógicos Referenciados na lógica de processamento da Consulta Externa.

O Quadro 6 deve ser utilizado para a definição da complexidade da parte de entrada de uma Consulta Externa.



**Quadro 6 Complexidade da Parte de Entrada de uma Consulta Externa.**

	<b>1 A 4 DER</b>	<b>5 A 15 DER</b>	<b>16 OU MAIS DER</b>
0 ou 1 ALR	Simples	Simples	Média
2 ALR	Simples	Média	Complexa
3 ou mais ALR	Média	Complexa	Complexa

*Dado Elementar Referenciado*, para a *parte de saída* da consulta externa, é um campo não recursivo, identificado pelo usuário, que aparece em uma Consulta Externa. As regras para a parte de saída são:

- ❑ Contar como um DER um cada campo não recursivo, identificado pelo usuário, que aparece na parte de saída da consulta.
- ❑ Não contar literais como DER. Literais incluem: nome do relatório, nome da tela, cabeçalho de coluna e nome dos campos.
- ❑ Não contar variáveis de paginação ou variáveis automáticas do sistema, tais como número de páginas, informações de posicionamento do cursos, comandos de paginação (próximo, anterior e setas de página em aplicações de interface gráfica como Windows), campos de data ou hora.
- ❑ Contar as seguintes técnicas de implementação física como um único DER, para o inteiro conjunto de campos:
  - Um campo lógico que é armazenado fisicamente como múltiplos campos, mas é requerido pelo usuário como uma única informação.
  - Campos, que por causa da tecnologia empregada no desenvolvimento do sistema, aparecem mais de uma vez em um Arquivo Lógico Interno.

*Arquivo Lógico Referenciado* (ALR), para a *parte de saída* da consulta externa, é um arquivo lido quando a consulta externa é processada. A regra de contagem para a parte de saída é:

- ❑ Contar a quantidade de Arquivos Lógicos Referenciados na lógica de processamento da Consulta Externa

O Quadro 7 deve ser utilizado para a definição da complexidade da parte de saída de uma Consulta Externa.

**Quadro 7 Complexidade da Parte de Saída de uma Consulta Externa**

	<b>1 A 5 DER</b>	<b>6 A 19 DER</b>	<b>20 OU MAIS DER</b>
0 ou 1 RLR	Simples	Simples	Média
2 A 3 RLR	Simples	Média	Complexa
4 ou mais RLR	Média	Complexa	Complexa

**Exemplo:** Foi solicitado pelo usuário, como parte do aplicativo a ser desenvolvido, uma tela (Figura 6) que permitisse a consulta a dados do arquivo de cliente partir da entrada do código do cliente. A consulta deve abranger sete campos e a existência do cliente deve ser validada. Além disso, em caso de erro, deve ser mostrada uma mensagem em um campo de erro.

A consulta a dados de clientes foi considerada uma Consulta Externa porque:

- ✓ Os dados serão recuperados de um Arquivo Lógico Interno da aplicação. Os resultados da saída serão recuperados e disponibilizados para fora da fronteira da aplicação através de uma saída de vídeo. Não há dados derivados no resultado da consulta. As partes de entrada e saída compõem um processo que é a menor unidade de atividade com significado para o usuário. O processo elementar é autocontido e deixa a aplicação em estado consistente. O processo não atualiza um ALI; somente efetua uma consulta. Nada leva a crer que exista uma Consulta Externa similar, nem que exista outra Consulta Externa com os mesmos dados.

**Figura 6 Layout da Tela de Consulta ao Cadastro de Clientes**

<b>Consulta - Cadastro de Clientes</b>	
Código do Cliente [ ]	
Nome [ ]	Razão social [ ]
Tipo [ ]	Território [ ]
CGC [ ]	Inscrição Est. [ ]
Endereço [ ]	Bairro [ ]
CEP [ ]	Cidade [ ]
Contato-1 [ ]	Telefone [ ]
Total compra ano [ ]	Total compra mês atual [ ]

**Contagem:**

- ❑ ALR:

O arquivo de *clientes* será acessado através do código do cliente e também o arquivo *histórico-cliente*, portanto ALR = 2

- ❑ DER:

Parte de Entrada: Será utilizado o código do cliente como campo de entrada e um campo adicional para uma possível mensagem de erro. DER = 2

Parte de Saída: Serão mostrados os 14 campos de informação, além do código do cliente, que não é contado como DER. O campo de mensagem de erro e a própria mensagem de erro não deve ser considerados aqui, já que se relacionam com a entrada e não com a saída. Portanto DER = 14.

Consulta a Clientes*Parte entrada*DER: 2ALR: 2Grau da parte de entrada: Simples**Grau da Função: Média***Parte saída*DER: 14ALR: 2Grau da parte de saída: Média

A Tabela 5 deve ser utilizada para traduzir a complexidade de uma Consulta Externa em pontos de função. A função Consulta a Clientes obteve complexidade média e portanto recebe 4 pontos de função brutos.

**Tabela 5 Pontos de Função de Consultas Externas**

<b><i>CE - Grau de Complexidade Funcional</i></b>	
Simples	3
Média	4
Complexa	6

### 3 CÁLCULO DO FATOR DE AJUSTE

Assim que o total de pontos de função não ajustados for obtido, o fator de ajuste deve ser determinado. O cálculo do fator de ajuste representa a funcionalidade geral da aplicação provida ao usuário pela aplicação e é obtido através da análise de 14 Características Gerais do Sistema (Quadro 8).

O nível de influência de cada uma das características varia de zero até cinco: *Nenhuma influência* (0), *Influência mínima* (1), *Influência moderada* (2), *Influência média* (3), *Influência significativa* (4) e *Influência forte* (5). Quando totalizados, os níveis podem alterar a contagem de pontos de função numa amplitude de -35% até +35%. O valor final do fator de ajuste pode variar de 0,65 até 1,35.

**Quadro 8 As 14 Características Gerais do Sistema**

<b>Características Gerais do Sistema</b>	
1. Comunicação de Dados	8. Atualização On-line
2. Funções Distribuídas	9. Processamento Complexo
3. Performance	10. Reusabilidade
4. Configuração do Equipamento	11. Facilidade de Implantação
5. Volume de Transações	12. Facilidade Operacional
6. Entrada de Dados On-Line	13. Múltiplos Locais
7. Interface com o Usuário	14. Facilidade de Mudanças

#### 3.1 Determinação do nível de influência (NI)

De acordo com o manual de contagem de pontos de função do IFPUG, cada uma das Características Gerais do Sistema possui uma descrição que permite determinar com exatidão seu grau de influência. O Nível de Influência de cada característica deve ser classificado conforme sua própria tabela.

A seguir são apresentadas as 14 características acompanhadas de uma breve descrição.

### 3.1.1 Comunicação de dados

Para a determinação dessa característica, analisar como os dados e informações de controle usados na aplicação são enviados e recebidos de outros terminais, estações de trabalho ou micros, através da utilização de recursos de comunicação de dados. Para que a comunicação aconteça são utilizados protocolos, um conjunto de padrões que permitem a transferência ou troca de informações entre sistemas ou periféricos. Todos os dispositivos de comunicação de dados utilizam algum tipo de protocolo de comunicação.

#### *Grau Descrição*

- 0 O processamento da aplicação é puramente "batch" ou executado em um PC isolado.
- 1 A aplicação é "batch" mas tem entrada de dados remota ou impressão remota.
- 2 A aplicação é "batch" mas tem entrada de dados remota e impressão remota.
- 3 Captura de dados "on-line", via terminal, rede de micros ou "front-end", para alimentar processos em "batch" ou sistemas de consultas.
- 4 Mais de um "front-end", mas a aplicação suporta apenas um tipo de protocolo de comunicação.
- 5 Mais de um "front-end", e a aplicação suporta vários tipos de protocolo de comunicação.

### 3.1.2 Processamento distribuído ou Funções distribuídas

Processamento distribuído ou funções distribuídas são características gerais que podem influenciar a complexidade da aplicação. A aplicação tem por característica o fato de que dados ou o processamento são distribuídos entre várias Unidades Centrais de Processamento (CPUs)

#### *Grau Descrição*

- 0 A aplicação não efetua a transferência de dados ou de processamento entre as CPUs da organização.
- 1 A aplicação prepara dados para o usuário final processar em outra CPU da instalação utilizando-se de software genérico (planilhas eletrônicas, editores de texto, bancos de dados).
- 2 Os dados são preparados, transferidos e processados em outra CPU da instalação. (Transferência de arquivos).
- 3 Processamento distribuído e transferência de dados "on-line" mas em uma única direção. (Processa numa CPU e transfere para outra CPU).
- 4 Processamento distribuído e transferência de dados "on-line" em ambas as direções. (Processamento cooperativo).
- 5 A aplicação a ser desenvolvida deve decidir dinamicamente qual a CPU mais apropriada para executar a função.

### 3.1.3 Performance

Devem ser analisados os objetivos de performance do sistema, estabelecidos e aprovados pelo usuário, em termos de tempo de resposta, que influenciam (ou influenciarão) o projeto, o desenvolvimento, a implantação e o suporte da aplicação.

<i>Grau</i>	<i>Descrição</i>
-------------	------------------

- |   |   |
|---|---|
| 0 | Nenhum requerimento especial de performance foi solicitado pelo usuário.  |
| 1 | Requerimentos de performance e de projeto foram estabelecidos e revistos, mas nenhuma ação especial foi requerida.  |
| 2 | Tempo de resposta e volume de processamento são itens críticos durante horários de pico de processamento. Nenhuma determinação especial para utilização do processador foi estabelecida. A data limite para disponibilidade de processamento é sempre o próximo dia útil. |
| 3 | Tempo de resposta e volume de processamento são itens críticos durante todo o horário comercial. Nenhuma determinação especial para a utilização do processador foi estabelecida. A data limite necessária para a comunicação com outros sistemas é um item importante.   |
| 4 | Os requerimentos de performance estabelecidos requerem tarefas de análise de performance na fase de análise e desenho da aplicação.   |
| 5 | Além do descrito no item anterior, ferramentas de análise de performance foram usadas nas fases de desenho, desenvolvimento e/ou implementação para atingir os requerimentos de performance estabelecidos pelo usuário.   |

### 3.1.4 Configuração do equipamento

Nessa característica, busca-se identificar o grau de utilização do(s) computador (es) em que será utilizado o sistema, que requeira considerações especiais de análise e *projeto* do aplicativo. A avaliação deve ser feita visando identificar a carga de trabalho onde a aplicação que está sendo analisada irá entrar em produção.

<i>Grau</i>	<i>Descrição</i>
-------------	------------------

- |   |   |
|---|---|
| 0 | Nenhuma restrição operacional explícita ou mesmo implícita incluída.  |
| 1 | Existem restrições operacionais leves. Não é necessário esforço especial para resolver as restrições.                   |
| 2 | Algumas considerações de ajuste de performance e segurança são necessárias.   |
| 3 | São necessárias especificações especiais de processador para um módulo específico da aplicação.                         |
| 4 | Restrições operacionais requerem cuidados especiais no processador central ou no processador dedicado.                  |
| 5 | Além das características do item anterior, há considerações especiais na distribuição do sistema e em seus componentes. |

### 3.1.5 Volume de transações

O volume de transações, é de tal magnitude que tem influência no projeto, desenvolvimento, implantação e manutenção da aplicação.

*Grau Descrição*

- |   |  |
|---|--|
| 0 | Não estão previstos períodos de picos de volume de transação   |
| 1 | Estão previstos picos de transações semanalmente, trimestralmente, anualmente ou em certo período do ano.  |
| 2 | São previstos picos semanais.  |
| 3 | São previstos picos diários.   |
| 4 | Alto volume de transações foi estabelecido pelo usuário, ou o tempo de resposta necessário atinge nível alto o suficiente para requerer análise de performance na fase de desenho. |
| 5 | Além do descrito no item anterior, é necessário utilizar ferramentas de análise de performance nas fases de desenho, desenvolvimento e/ou implantação.                             |

### 3.1.6 Entrada de dados *on-line*

Essa característica quantifica a entrada de dados *on-line* provida pela aplicação.

*Grau Descrição*

- |   |   |
|---|---|
| 0 | Todas as transações são processadas em modo <i>batch</i> .        |
| 1 | De 1% a 7% das transações entradas de dados <i>on-line</i> .      |
| 2 | De 8% a 15% das transações entradas de dados <i>on-line</i> .     |
| 3 | De 16% a 23% das transações entradas de dados <i>on-line</i> .    |
| 4 | De 24% a 30% das transações entradas de dados <i>on-line</i> .    |
| 5 | Mais de 30% das transações são entradas de dados <i>on-line</i> . |

### 3.1.7 Interface com o usuário

As funções da aplicação executadas "on-line" enfatizam que o projeto da aplicação foi voltado para a eficiência do usuário final, e é refletido pelo número e tipo de facilidades disponibilizadas:

- ✓ Auxílio à navegação (teclas de função, acesso direto e menus dinâmicos).
- ✓ Menus.
- ✓ Menus pop-up.
- ✓ Documentação e *help* (auxílio) *on-line*.
- ✓ Movimento automático do cursor.
- ✓ Seleção de cursor em campos da tela.
- ✓ Rolagem vertical e horizontal.
- ✓ Impressão remota (através de transações *on-line*).
- ✓ Teclas de função preestabelecidas.
- ✓ Processos batch submetidos a partir de transações *on-line*.
- ✓ Utilização intensa de campos com vídeo reverso, intensificados, sublinhados, coloridos e outros indicadores.
- ✓ Impressão da documentação das transações *on-line* através de *Hard-copy*.
- ✓ Utilização de mouse.
- ✓ O menor número possível de telas para executar as funções de negócio.

- ✓ Suporte bilíngüe ( o suporte a duas línguas; contar como quatro itens).
- ✓ Suporte multilíngüe ( o suporte a mais de duas línguas; contar como seis itens).

#### *Grau Descrição*

- 0 A aplicação não apresenta nenhum dos itens descritos.
- 1 A aplicação apresenta de um a três dos itens descritos.
- 2 A aplicação apresenta de quatro a cinco dos itens descritos.
- 3 A aplicação apresenta mais de cinco dos itens descritos, mas não há requerimentos específicos do usuário quanto à amigabilidade do sistema.
- 4 A aplicação apresenta mais de cinco dos itens descritos, e foram estabelecidos requerimentos quanto à amigabilidade fortes, tais como minimização da digitação para mostrar inicialmente os valores utilizados com mais frequência.
- 5 A aplicação apresenta mais de cinco dos itens descritos, e foram estabelecidos requerimentos quanto à amigabilidade fortes o suficiente para requerer ferramentas e processos especiais para demonstrar antecipadamente que os objetivos foram alcançados.

### **3.1.8 Atualização on-line**

Verificar se aplicação provê atualização *on-line* dos Arquivos Lógicos Internos.

#### *Grau Descrição*

- 0 Nenhum.
- 1 Atualização *on-line* de um a três Arquivos Lógicos Internos. O volume de atualização é baixo e a recuperação de dados é fácil.
- 2 Atualização *on-line* de mais de três Arquivos Lógicos Internos. O volume de atualização é baixo e a recuperação de dados é fácil.
- 3 Atualização *on-line* da maioria dos Arquivos Lógicos Internos.
- 4 Em adição ao item anterior, é necessário proteção contra perda de dados que foi desenhada e programada no sistema.
- 5 Além do item anterior, altos volumes trazem considerações de custo no processo de recuperação. Processos para automatizar a recuperação foram incluídos minimizando a intervenção do operador.

### **3.1.9 Processamento complexo**

Processamento complexo é uma das características da aplicação. Deve ser analisada a presença dos seguintes componentes:

- ✓ Controle sensitivo (processamento especial de auditoria) e/ou processamento de segurança específica da aplicação.
- ✓ Processamento lógico extensivo.
- ✓ Processamento matemático extensivo.
- ✓ Processamento gerando muitas exceções, resultando em transações incompletas que devem ser processadas novamente. Exemplo: transações de auto-atendimento bancário interrompidas por problemas de comunicação ou com dados incompletos.
- ✓ Processamento complexo para manipular múltiplas possibilidades de entrada/saída, como por exemplo, multimídia.



*Grau Descrição*

- 0 Nenhum dos itens descritos.
- 1 Apenas um dos itens descritos.
- 2 Dois dos itens descritos.
- 3 Três dos itens descritos.
- 4 Quatro dos itens descritos.
- 5 Todos os cinco dos itens descritos.

### 3.1.10 Reusabilidade

Analisar se a aplicação e seu código serão/foram projetados, desenvolvidos e mantidos visando utilização em outras aplicações.

*Grau Descrição*

- 0 Nenhuma preocupação com reutilização de código.
- 1 Código reutilizado foi usado somente dentro da aplicação.
- 2 Requerimentos de performance e de projeto foram estabelecidos e revistos, mas nenhuma ação especial foi requerida.
- 3 Menos de 10% da aplicação foi projetada prevendo utilização posterior do código por outra aplicação.
- 4 10% ou mais da aplicação foi projetada prevendo utilização posterior do código por outra aplicação.
- 5 A aplicação foi especificamente projetada e/ou documentada para ter seu código facilmente reutilizado por outra aplicação e a aplicação é customizada para uso através de parâmetros que podem ser alterados pelo usuário.

### 3.1.11 Facilidade de implantação

Facilidade de conversão e de implantação são características da aplicação. Verificar a existência de um plano de conversão e de implantação e/ou a existência de ferramentas de conversão providas e testadas durante a fase de teste da aplicação.

*Grau Descrição*

- 0 Nenhuma consideração especial foi estabelecida pelo usuário e nenhum procedimento especial é requerido na implantação.
- 1 Nenhuma consideração especial foi estabelecida pelo usuário, mas procedimentos especiais são necessários na implantação.
- 2 Requerimentos de conversão e implantação foram estabelecidos pelo usuário e roteiros de conversão e implantação foram providos e testados. O impacto da conversão no projeto não foi considerado importante.
- 3 Requerimentos de conversão e implantação foram estabelecidos pelo usuário e roteiros de conversão e implantação foram providos e testados. O impacto da conversão no projeto é considerado importante.
- 4 Além do item 2, conversão automática e ferramentas de implantação foram providas e testadas.
- 5 Além do item 3, conversão automática e ferramentas de implantação foram providas e testadas.

### 3.1.12 Facilidade operacional

Facilidade de operação é uma característica da aplicação. As especificações da aplicação podem demandar que sejam providos procedimentos automatizados que minimizem a intervenção manual de operadores. Procedimentos de inicialização, salvamento e recuperação do sistema podem ser providos e testados durante a fase de teste do sistema. Verificar se a aplicação minimiza a necessidade de atividades manuais, tais como montagem de fitas, manuseio de papel e intervenção do operador.

*Grau Descrição*

- |     |   |
|-----|---|
| 0   | Nenhuma consideração especial de operação, além do processo normal de salva foi estabelecido pelo usuário.  |
| 1-4 | <p><i>Verifique quais das seguintes afirmativas podem ser identificadas na aplicação. Selecione as que forem aplicadas. Cada item vale um ponto, exceto se definido explicitamente.</i></p> <ul style="list-style-type: none"> <li>✓ Foram desenvolvidos processos de inicialização, salva e recuperação, mas a intervenção do operador é necessária.</li> <li>✓ Foram estabelecidos processos de inicialização, salva e recuperação, e nenhuma intervenção do operador é necessária (contar como dois itens).</li> <li>✓ A aplicação minimiza a necessidade de montar fitas magnéticas.</li> <li>✓ A aplicação minimiza a necessidade de manuseio de papel.</li> </ul> |
| 5   | A aplicação foi desenhada para trabalhar sem operador; nenhuma intervenção do operador é necessária para operar o sistema além de executar e encerrar a aplicação. A aplicação possui rotinas automáticas para recuperação em caso de erro.   |

### 3.1.13 Múltiplos locais

Verificar se a aplicação foi especificamente projetada, desenvolvida e modificada para ser instalada em múltiplos locais ou para múltiplas organizações.

*Grau Descrição*

- |   |  |
|---|--|
| 0 | Os requerimentos do usuário não consideram a necessidade de instalação em mais de um local.  |
| 1 | A necessidade de múltiplos locais foi considerada no projeto, e a aplicação foi desenhada para operar apenas sobre o <i>mesmo</i> ambiente de software e hardware.       |
| 2 | A necessidade de múltiplos locais foi considerada no projeto, e a aplicação está preparada para trabalhar apenas em ambientes <i>similares</i> de software e hardware.   |
| 3 | A necessidade de múltiplos locais foi considerada no projeto, e a aplicação está preparada para trabalhar sob <i>diferentes</i> ambientes de hardware e/ou software.     |
| 4 | Plano de documentação e manutenção foram providos e testados para suportar a aplicação em <i>múltiplos</i> locais; além disso, os itens 1 ou 2 caracterizam a aplicação. |
| 5 | Plano de documentação e manutenção foram providos e testados para suportar a aplicação em <i>múltiplos</i> locais; além disso, o item 3 caracteriza a aplicação.         |

### 3.1.14 Facilidade de mudanças (flexibilidade)

A aplicação foi especificamente projetada, desenvolvida e suporta manutenção, visando facilidade de mudanças através de capacidade de consultas e relatórios flexíveis, bem como a parametrização dos dados de controle do negócio, de forma que o cliente possa modificá-los a qualquer momento.

#### *Grau Descrição*

- |   |  |
|---|--|
| 0 | Caso a aplicação não disponibilize ferramentas será atribuído o Nível de Influência igual a 0 (zero) para esta característica da aplicação.  |
| 1 | É fornecido recurso flexível de consulta e de emissão de relatório capaz de manipular solicitações simples, com lógica de "and/or" aplicada somente a um Arquivo Lógico Interno.                 |
| 2 | É fornecido recurso flexível de consulta e de emissão de relatório capaz de manipular solicitações de média complexidade, com lógica de "and/or" aplicada a mais de um Arquivo Lógico Interno.   |
| 3 | É fornecido recurso flexível de consulta e de emissão de relatório capaz de manipular solicitações de alta complexidade, com lógica de "and/or" aplicada a um ou mais Arquivos Lógicos Internos. |
- Caso os dados de controle do negócio da aplicação sejam mantidos em tabelas ou arquivos, que possam ser atualizadas pelo usuário através de processos "on-line" e interativos, acrescentar ao Nível de Influência determinado acima:
1. Se estas alterações só são efetivadas no próximo dia útil;
  2. Se estas alterações são efetivadas imediatamente.

## 3.2 Cálculo do Fator de Ajuste e dos Pontos de Função Ajustados

Os seguintes passos devem ser executados para o cálculo do fator de ajuste:

1. Avaliar o impacto de cada uma das 14 Características Gerais do Sistema no aplicativo que está sendo contado, atribuindo peso de 0 a 5 para cada uma das 14 características.
2. Calcular o nível de influência (NI) através da soma dos pesos de cada uma das 14 características.
3. Calcular o fator de ajuste (FA) através da Equação 1.

#### **Equação 1 Equação do Fator de Ajuste**

$$FA = (NI * 0,01) + 0,65$$

4. Calcular pontos de função ajustados através da Equação 2.

#### **Equação 2 Equação de Cálculo de PFs Ajustados**

$$FPA = PF \text{ Brutos} * FA$$

---

Exemplo: A contagem dos pontos de função não ajustados resultou em 79 e a soma dos pesos das 14 Características Gerais do Sistema resultou em 16 (nível de influência = 16).

$$\mathbf{Fator\ de\ Ajuste} = (NI * 0,01) + 0,65$$

$$\text{Fator de Ajuste} = (16 * 0,01) + 0,65 = 0,81$$

$$\mathbf{FP\ Ajustados} = PF\ Brutos * Fator\ de\ ajuste$$

$$PF\ Ajustados = 79 * 0,81 = 63,99\ PF$$

$$PF\ Ajustados = 64\ \text{pontos de função}$$

---

## 4 CONTAGEM DE PONTOS DE FUNÇÃO EM PROJETOS

### 4.1 Contagens e Fases de Desenvolvimento

A contagem de Pontos de Função *nas fases iniciais do projeto* é talvez a contagem mais difícil, pois pode ocorrer a ausência de vários elementos que compõem o processo de contagem. Assumindo que a fase de *Business Design* (Análise de Sistema) tenha terminado, provavelmente todos os elementos necessários a uma contagem efetiva de pontos de função estarão presentes.

A documentação mínima para que o processo possa ser executado adequadamente é: o Modelo de Dados da Aplicação, *Layout* dos relatórios, *Layout* das telas de entrada de dados, *Layout* das telas de consulta, *Layout* dos arquivos de interface e *Layout* dos arquivos que são entrada para o sistema.

Contagens de pontos de função realizadas nas fases iniciais do desenvolvimento de sistemas são estimativas da funcionalidade desejada. À medida que o projeto caminha, o escopo do sistema vai sendo validado, e suas funções vão sendo desenvolvidas, de forma que é perfeitamente normal identificar funções adicionais que não foram previstas inicialmente.

A contagem de pontos de função *durante o desenvolvimento do projeto* acontece então quando uma mudança significativa no escopo da aplicação acontece. Três casos típicos que podem ocorrer são: a *inclusão de um novo processo ou função*, a *eliminação de um processo ou função*, ou ainda a *alteração de parte de processo ou função*.

Se houver a *inclusão de um novo processo ou função* após a contagem inicial, a documentação mínima, conforme descrito anteriormente, deve estar disponível, bem como o primeiro total obtido de pontos de função não ajustados. Para a análise e contagem dessa nova função deve-se observar somente a documentação mínima do que foi incluído. Conta-se os pontos de função brutos da nova função e soma-se aos pontos de função não ajustados já obtidos, aplicando-se posteriormente o fator de ajuste.

No caso de *exclusão de uma função*, efetua-se um processo semelhante à inclusão de uma nova função. Conta-se os pontos de função referentes ao processo ou

função excluída, excluindo-os do total de pontos de função não ajustados, aplicando-se posteriormente o fator de ajuste.

A alteração de parte de processo ou função de um projeto de aplicação é o cenário mais complexo. Deve-se estudar a melhor forma de fazer o novo cálculo, porém o ideal para garantir a consistência final da contagem é que toda a aplicação seja recontada através da documentação mínima já descrita anteriormente.

*Após o término do projeto*, quando a aplicação já se encontra instalada, efetua-se o cálculo dos pontos de função instalados para obtenção de informações reais e físicas da mesma. Nesse caso, toda a aplicação deve ser contada novamente e como documentação mínima para auxiliar na contagem deve-se ter em mãos o *Layout* dos arquivos mantidos pela aplicação (para contagem de Arquivos Lógicos Internos), o *Layout* dos arquivos de interface externa (para contagem de Arquivos de Interface Externa), o impressão das telas de entrada de dados (para contagem de Entradas Externas e de Consultas Externas), impressão das telas de consulta (para contagem de Consultas Externas e Saídas Externas) e exemplos de cada tipo de relatório do sistema (para contagem de Consultas Externas e Saídas Externas).

## **4.2 Exemplo de Análise de Pontos de Função em Projeto de Nova Aplicação**

---

As seguintes etapas serão cumpridas para o projeto de desenvolvimento:

1. Descrição do sistema a ser desenvolvido
2. Levantamento das principais funções do sistema
3. *Layout* dos Arquivos Internos e Externos do sistema
4. *Layout* das telas do sistema
5. *Layout* Relatórios do sistema
6. Identificação das funções quanto a complexidade
7. Classificação das funções do sistema
8. Cálculo dos pontos de função não ajustados
9. Cálculo do fator de ajuste e dos pontos de função ajustados

A documentação mínima exigida para a contagem é: Modelo de Dados da Aplicação, *Layout* dos relatórios, *Layout* das telas de entrada de dados, *Layout* das telas de consulta, *Layout* dos arquivos de interface e *Layout* dos arquivos que são entrada para o sistema.

### **4.2.1 Descrição do sistema a ser desenvolvido**

O sistema de clientes tem como objetivo apoiar o Departamento de Vendas e o Departamento de Marketing.

O sistema de clientes será desenvolvido em uma linguagem de Quarta Geração usando banco de dados e será implantado em um equipamento Unix ou em uma rede de micros. O sistema será utilizado por cerca de 60 funcionários em um ambiente on-line, através de microcomputadores. Os usuários conhecem pouco de informática; assim sendo, as telas devem ser de fácil utilização, além de terem telas de auxílio associadas. O volume de transações não deverá ser alto, entretanto o fim de ano é um período crítico.

#### 4.2.2 Levantamento das principais funções do sistema

As principais funções do sistema são: cadastramento, alteração e consulta a dados de cliente, cadastramento, alteração e consulta no cadastro de produtos, emissão de relatórios, pedidos de clientes e segurança no acesso.

#### 4.2.3 Layout dos Arquivos Internos e Externos do sistema

**Quadro 9 Layout do Arquivo de Clientes**

<i>Clientes</i>		
<i>Item de dados</i>	<i>tamanho</i>	<i>tipo</i>
Código-cliente	6	Númerico
Nome-cliente	12	Alfa
Razão-social	30	Alfa
Tipo-cliente	2	Númerico
Território	3	Alfa
Endereço	30	Alfa
Bairro	12	Alfa
CEP	8	Númerico
Cidade	12	Alfa
CGC	12	Númerico
Inscrição-estadual	12	Númerico
Contato-1	20	Alfa
Telefone-1	10	Númerico
Contato-2	20	Alfa
Telefone-2	10	Númerico

**Quadro 10 Layout do Arquivo de Produtos**

<i>Produtos</i>		
<i>item de dados</i>	<i>tamanho</i>	<i>tipo</i>
Código-produto	12	Númerico
Nome-produto	12	Alfa
Descrição-produto	30	Alfa
Tipo-produto	2	Númerico
Unidade-produto	3	Alfa
Custo-médio-unitário	12	Númerico
Preço-médio-unitário	12	Númerico
Principal-fornecedor	12	Alfa
Preço-último-fornecimento	12	Númerico
Principal-cliente	12	Alfa
Data-último-pedido	8	Númerico
Quantidade-último-pedido	12	Númerico
Preço-unit-último-pedido	12	Númerico

**Quadro 11 Layout do Arquivo de Pedidos**

<i>Pedidos</i>		
<i>item de dados</i>	<i>tamanho</i>	<i>tipo</i>
Número-pedido	10	Númerico
Código-produto	12	Númerico
Código-cliente	6	Númerico
Quantidade	12	Númerico
Preço-unitário	12	Númerico
Data-pedido	8	Númerico

**Quadro 12 Layout do Arquivo de Senhas**

<i>Senhas</i>		
<i>item de dados</i>	<i>tamanho</i>	<i>tipo</i>
Código-usuário	6	Númerico
Senha	6	Alfa
Data-validade	8	Númerico
Data-último-acesso	8	Númerico
Hora-último-acesso	8	Númerico

**Quadro 13 Layout do Arquivo de Histórico de Clientes**

<i>Histórico de Clientes</i>		
<i>item de dados</i>	<i>tamanho</i>	<i>Tipo</i>
Código-cliente	6	Númerico
Produto-1-mais-comprado-94	12	Númerico
Valor-compra-produto-1-94	12	Númerico

#### 4.2.4 Layout das telas do sistema

A cada tela do sistema deve ser associada uma tela de auxílio, visando agilizar a utilização do sistema por novos usuários e aumentando a produtividade de utilização.

**Figura 7 Tela de Inclusão de clientes**

Inclusão de Clientes		
Código [ ]	Nome [ ]	Razão Social [ ]
Tipo [ ]	Território [ ]	
CGC [ ]	Inscrição Estadual [ ]	
Endereço [ ]	Bairro [ ]	
CEP [ ]	Cidade [ ]	
Contato-1 [ ]	Telefone [ ]	
Contato-2 [ ]	Telefone [ ]	

Arquivos referenciados: Clientes

Dados: Todos do arquivo.

Regras: Enviar mensagem de erro caso seja digitado CGC inválido.  
Enviar mensagem de erro caso cliente já exista.

**Figura 8 Exclusão de clientes**

Exclusão – Cadastro de Clientes
Código do cliente [ ]

Arquivos referenciados: Clientes

Dados: Somente é necessário digitar o código do cliente.

Regras: Verificar se cliente existe; se não existir, enviar mensagem de erro.



**Figura 9 Alteração dos dados de cliente**

Alteração dos Dados - Cadastro de Clientes		
Código [ ]	Nome [ ]	Razão Social [ ]
Tipo [ ]	Território [ ]	
CGC [ ]	Inscrição Estadual [ ]	
Endereço [ ]	Bairro [ ]	
CEP [ ]	Cidade [ ]	
Contato-1 [ ]	Telefone [ ]	
Contato-2 [ ]	telefone [ ]	

Arquivos referenciados: Clientes

Dados: Todos podem ser alterados, com exceção do código do cliente.

Regras: Enviar mensagem de erro caso o cliente não exista.

**Figura 10 Consulta a Clientes**

Consulta - Cadastro de Clientes	
Código do Cliente [ ]	
Nome [ ]	Razão social [ ]
Tipo [ ]	Território [ ]
CGC [ ]	Inscrição Estadual [ ]
Endereço [ ]	Bairro [ ]
CEP [ ]	Cidade [ ]
Contato-1 [ ]	Telefone [ ]
Total compra ano [ ]	Total compra mês atual [ ]

Arquivos referenciados: Clientes

Dados: 12 do Cadastro de Clientes e dois do Histórico de Clientes.

Regras: Deverá ser digitado o código do cliente.

Verificar se cliente existe, se não existir, enviar mensagem de erro.

O arquivo Histórico de Clientes pertence a outro aplicativo.

**Figura 11 Inclusão de Produtos**

Inclusão de Produtos			
Código [ ]	Nome [ ]	Descrição [ ]	
Tipo [ ]	Unidade [ ]	Custo médio [ ]	Preço Médio [ ]
Principal Fornecedor [ ]			Preço Médio [ ]
Principal Cliente [ ]			
Data última compra [ ]	Quantidade última compra [ ]		Preço Unitário. [ ]

Arquivos referenciados: Produtos

Dados: Todos do arquivo

Regras: Envie mensagem de erro caso o produto já exista.

**Figura 12 Alteração dos Dados de Produto**

Alteração do cadastro de Produtos			
Código [ ]	Nome [ ]	Descrição [ ]	
Tipo [ ]	Unidade [ ]	Custo médio [ ]	Preço Médio [ ]
Principal Fornecedor [ ]	Preço Médio [ ]		
Principal Cliente [ ]			
Data última compra [ ]	Quantidade última compra [ ]	Preço Unitário. [ ]	

Arquivos referenciados: Produtos

Dados: Todos podem ser alterados, menos o código do produto.

Regras: Envie mensagem de erro caso o produto não exista.

**Figura 13 Consulta a Produtos**

Consulta ao cadastro de Produtos		
Código [ ]	Nome [ ]	Descrição [ ]
Tipo [ ]	Unidade [ ]	
Custo médio [ ]	Preço Médio [ ]	
Principal Fornecedor [ ]	Preço Médio [ ]	
Principal Cliente [ ]	Data última compra [ ]	
Quantidade última compra [ ]	Preço Unitário. [ ]	

Arquivos referenciados: Produtos

Dados: 13 campos

Regras: Verificar se produto existe.

**Figura 14 Exclusão de Produtos**

Exclusão de Produtos
Código do Produto [ ]

Arquivos referenciados: Produtos

Dados: Somente é necessário digitar o código do produto.

Regras: Verificar se o produto existe.

**Figura 15 Pedido de Clientes**

Pedido de Clientes	
Número do pedido [ ]	Data pedido [ ]
Código cliente [ ]	
Código produto [ ]	Quantidade [ ] Preço unitário [ ]

Arquivos referenciados: Clientes, produtos e pedidos

Dados: 6 campos

Regras: Verificar se pedido existe, se produto existe e se cliente existe.

Atualizar informações sobre o pedido no cadastro de produtos.

**Figura 16 Tela de Identificação**

Tela de Identificação
Código do usuário [ ]

Arquivos referenciados: Senhas

Dados: É necessário informar o usuário e a senha

Regras: O usuário e a senha deverão ser validados para acesso ao sistema.  
Uma mensagem de erro deverá ser enviada quando necessário.

**Figura 17 Menu do Sistema**

Menu de Seleção	Auxílio [ ]
1. Inclusão de clientes	
2. Alteração de dados de clientes	
3. Consulta ao cadastro de clientes	
4. Exclusão de clientes	
5. Inclusão de produtos	
6. Alteração de dados de produtos	
7. Consulta ao cadastro de produtos	
8. Exclusão de produtos	
9. Pedidos de clientes	
10. Relação de clientes inativos	Data [ ]
Opção desejada [ ]	

Arquivos referenciados: nenhum

Dados: É necessário informar o número da tela desejada.

Regras: Caso o usuário solicite a opção de relação de clientes inativos, ele deverá informar também uma data e serão considerados como resposta os clientes que tiverem seu último pedido realizado antes da data especificada. Se nenhum registro atender à opção de execução, o relatório saíra em branco.

**Figura 18 Telas de auxílio**

Tela de auxílio – Menu de seleção
Essa tela permite que o usuário escolha a função que deseja utilizar:
❖ Cadastro de clientes – permite a inclusão de novos clientes
❖ Alteração de dados de clientes – permite a alteração de dados de um certo cliente
❖ Consulta a clientes – consulta informações sobre um certo cliente
❖ Exclusão de clientes – exclui um determinado cliente
❖ Cadastro de produtos – permite a inclusão de novos produtos
❖ Alteração de dados de produtos – permite a alteração de dados de um certo produto
❖ Consulta a produtos – consulta informações sobre um certo produto
❖ Exclusão de produtos – exclui um determinado produto
Retorna à tela anterior [ ]

Arquivos referenciados: Nenhum

Dados: Nenhum

Regras: A respectiva tela de auxílio deve ser apresentada sempre que o usuário solicitar através de preenchimento do campo de auxílio

#### 4.2.5 Layout dos Relatórios do sistema

**Figura 19 Relação de Clientes**

Relação de Clientes				Data __/__/__			Página __	
Código	Nome	Razão Social	Tipo	Território	CGC	Inscr. Est.	Contato	Telefone
.....	.....	.....	.....	.....	.....	.....	.....	.....
.....	.....	.....	.....	.....	.....	.....	.....	.....
Total de Clientes __								

**Figura 20 Relação de Clientes Inativos**

Relação de Clientes Inativos					Data __/__/__		Página __	
Código	Razão social	Tipo	Endereço	Bairro	CEP	Cidade	Último Pedido	
.....	.....	.....	.....	.....	.....	.....	.....	

**Figura 21 Relação de Produtos**

Relação de Produtos				Data __/__/__		Página __	
Código	Nome	Descrição	Unidade	Custo médio	Preço médio	Principal Fornecedor	
.....	.....	.....	.....	.....	.....	.....	

#### 4.2.6 Identificação das funções do sistema

Depois de observada e analisada a documentação mínima, foram determinadas as seguintes funções do sistema (Quadro 14):

**Quadro 14 Funções Identificadas no Sistema para o Exemplo**

<b>Arquivos Lógicos Internos</b> (ALI)	<i>Cadastro de clientes, Cadastro de produtos, Cadastro de pedidos</i>
<b>Arquivos de Interface Externas</b> (AIE)	<i>Histórico de clientes, Cadastro de senhas</i>
<b>Entradas Externas</b> (EE)	<i>Inclusão de clientes, Alteração de clientes, Exclusão de clientes, Inclusão de produtos, Alteração de produtos, Exclusão de produtos, Pedido de clientes</i>
<b>Saídas Externas</b> (SE)	<i>Relação de clientes, Relação de produtos</i>
<b>Consultas Externas</b> (CE)	<i>Consulta a clientes, Consulta a produtos, Telas de auxílio, Relação de clientes inativos</i>

Observações:

1. A tela de identificação não é contada como função, já que não provê segurança da aplicação, apenas segurança de acesso.
2. A tela menu do sistema, como só provê seleção de telas, não é considerada como função. É considerada, entretanto, como parte de entrada da consulta de relação de clientes inativos.
3. A função relação de clientes inativos, apesar de ter sido considerada um relatório durante o levantamento inicial, não é uma saída externa, mas sim uma consulta externa, pois não possui dados derivados e sua execução é feita através de tela de menu.

#### 4.2.7 Classificação das funções quanto a complexidade

As funções identificadas foram analisadas (Quadro 15) segundo as quantidades de Dados Elementares Referenciados (DER), Arquivos Lógicos Referenciados (ALR) e Registros Lógicos Referenciados (RLR).

**Quadro 15 Classificação de complexidade das funções identificadas**

<i><b>Tipo de função</b></i>	<i><b>Função</b></i>	<i><b>Grau</b></i>
Arquivos Lógicos Internos	Cadastro de clientes DER: 15 RLR: 1	Simples
	Cadastro de produtos DER: 13 RLR: 1	Simples
	Cadastro de pedidos DER: 6 RLR: 1	Simples
Arquivos de Interface Externa	Histórico de Clientes DER: 3 (código do cliente, total-compra-ano, total-compra-mês-atual) RLR: 1	Simples
	Arquivo de Senhas DER: 3 (usuário, senha e data de validade) RLR: 1	Simples
Entradas Externas	Inclusão de clientes DER: 16 (15 campos, mensagens de erro) ALR: 1	Média
	Alteração de clientes DER: 16 (15 campos, mensagens de erro) ALR: 1	Média
	Exclusão de clientes DER: 2 (código do cliente e mensagens de erro) ALR: 1	Simples
	Inclusão de produtos DER: 14 (13 campos, mensagens de erro) ALR: 1	Simples
	Alteração de produtos DER: 14 (13 campos, mensagens de erro) ALR: 1	Simples
	Exclusão de produtos DER: 2 (código de produto e mensagens de erro) ALR: 1	Simples

<i><b>Tipo de função</b></i>	<i><b>Função</b></i>	<i><b>Grau</b></i>
------------------------------	----------------------	--------------------

Entradas Externas	Pedido de clientes DER: 7 (6 campos e mensagens de erro) ALR: 3 (clientes, produtos, pedidos)	Complexa
Saídas Externas	Relação de clientes DER: 10 (9 campos e total de clientes) ALR: 1	Simples
	Relação de produtos DER: 7 ALR: 1	Simples

**Observações:** A função relação de clientes é uma saída externa pois possui dados derivados (campo total-de-clientes). A função relação de produtos não possui dados derivados, entretanto é uma saída externa pois sua execução é feita através de processo regular de execução através de *tarefas*.

<b><i>Tipo de função</i></b>	<b><i>Função</i></b>	<b><i>Grau</i></b>
Consultas Externas	Consulta de clientes: Grau da parte de entrada: Simples DER: 2 (código do cliente e mensagens de erro) ALR: 2 (clientes e histórico de clientes) Grau da parte de saída: Média DER: 14 (12 de clientes e 2 de histórico de clientes) ALR: 2	Média
	Consulta a produtos: Grau da função: Grau da função: Simples DER: 2 (código do produto e mensagens de erro) ALR: 1 (cadastro de produtos) Grau da função: Simples DER: 12 (código do produto não é considerado) ALR: 1	Simples
	Relação de clientes inativos: Grau da função: Grau da função: Simples DER: 3 (data, opção e mensagens de erro) ALR: 0 Grau da função: Média DER: 8 ALR: 2 (clientes e produtos)	Média
	Telas de Auxílio <i>O conjunto de telas de auxílio é considerado por definição, uma consulta externa de grau simples, independentemente de sua quantidade.</i>	

#### 4.2.8 Cálculo dos pontos de função não ajustados

A Tabela 6 é utilizada para determinar o total geral de pontos de função não ajustados do sistema.

Tabela 6 Cálculo dos Pontos de Função não ajustados

<i>Tipo de função</i>	<i>Complexidade Funcional</i>		<i>Total Complexidade</i>	<i>Total tipo função</i>
Arquivo Lógico Interno	3 Simples	x 7 =	21	21
	0 Média	x 10 =	0	
	0 Complexa	x 15 =	0	
Arquivo de Interface Externa	2 Simples	x 5 =	10	10
	0 Média	x 7 =	0	
	0 Complexa	x 10 =	0	
Entradas Externas	4 Simples	x 3 =	12	26
	2 Média	x 4 =	8	
	1 Complexa	x 6 =	6	
Saídas Externas	2 Simples	x 4 =	8	8
	0 Média	x 5 =	0	
	0 Complexa	x 7 =	0	
Consultas Externas	2 Simples	x 3 =	6	14
	2 Média	x 4 =	8	
	0 Complexa	x 6 =	0	
<b>Total de pontos de função não ajustados</b>				<b>79</b>

#### 4.2.9 Cálculo do Fator de Ajuste e dos Pontos de Função Ajustados

As 14 Características Gerais do Sistema devem ser analisadas considerando-se a descrição do projeto efetuada pelo usuário. Deve-se evitar a dedução das influências das características, buscando informações complementares com o usuário principal do projeto ou da área à qual a dúvida está relacionada.

Depois de avaliar o nível de influência de cada uma das 14 Características Gerais do Sistema no aplicativo que está sendo contado (Quadro 15), calcula-se o nível de influência (NI) através da soma dos pesos de cada uma das 14 características.

Quadro 16 Cálculo do Fator de Ajuste

<i>Grau</i>	<i>Característica</i>	<i>Análise</i>
4	Comunicação de Dados	A aplicação não é <i>batch</i> . Ela inclui entrada de dados on-line, porém não se resume a entrada de dados. Nada foi mencionado quanto à quantidade de protocolos de comunicação. Considera-se que seja apenas 1 protocolo como resposta mais provável. É o caso de buscar esclarecimentos com o usuário.
0	Funções Distribuídas	Nada foi comentado quanto à execução em outros processadores. A aplicação pode ser executada em ambiente Unix ou em rede de micros. Não há preocupação com a distribuição das funções.
1	Performance	Segundo o usuário, o volume de transações não deverá ser alto. O período crítico é no fim de ano. Não se pode colocar que não há nenhum requerimento de performance.
1	Configuração Carregada	Nada foi informado para deduzir de que a configuração do equipamento está carregada. Nenhuma consideração de análise ou desenho foi requerida. O período de fim de ano é crítico.
1	Volume das Transações	Estão previstos picos de transações anualmente.
<i>Grau</i>	<i>Característica</i>	<i>Análise</i>

5	Entrada de dados <i>on-line</i>	Pelo descrito, as transações são processadas somente on-line.
1	Interface com o Usuário	Foram especificados menus de navegação e interface através de microcomputadores. Portanto, poucas características dos itens especificados.
3	Atualização <i>on-line</i>	Atualização <i>on-line</i> de todos os Arquivos Lógicos Internos; entretanto, não foi especificada proteção especial contra perda de dados ou processos automáticos para recuperação.
0	Processamento Complexo	A aplicação não envolve processamento complexo.
0	Reusabilidade	Não houve preocupação com reutilização de código.
0	Facilidade de Instalação	Nenhuma consideração foi estabelecida visando facilitar a instalação do aplicativo.
0	Facilidade de Operação	Nenhuma consideração específica foi definida envolvendo procedimentos de salva, recuperação ou operador do equipamento.
0	Múltiplos Locais	Os requerimentos do usuário não consideraram a necessidade de instalação em mais de um local.
0	Facilidade de mudanças	Não houve nenhum tipo de preocupação específica visando facilitar a flexibilização de mudanças do aplicativo.
<b>16</b>	<b>Total</b>	

O nível de influência obtido foi 16. O passo seguinte é o cálculo do fator de ajuste (Equação 1) e em seguida os pontos de função ajustados (Equação 2).

$$\text{Fator de ajuste} = (NI * 0,01) + 0,65$$

$$\text{Fator de ajuste} = (16 * 0,01) + 0,65 = 0,81$$

$$PF \text{ Ajustados} = PF \text{ Brutos} * \text{Fator de ajuste}$$

$$PF \text{ Ajustados} = 79 * 0,81 = 63,99 \text{ PF}$$

$$PF \text{ Ajustados} = 64 \text{ pontos de função}$$

Portanto, o sistema solicitado pelo usuário é da ordem de grandeza de 64 pontos de função. A partir desse dado, pode-se calcular o esforço e o tempo necessários para execução do projeto baseando-se em comparações com dados obtidos de projetos concluídos anteriormente.



## 5 CONTAGEM DE PF EM PROJETOS DE MANUTENÇÃO DE APLICAÇÕES

Após a liberação do software para o usuário, é muito comum a ocorrência de pedidos de alterações [Pressman, 1997]. Casos típicos de modificações que ocorrem tem como objetivo identificar e corrigir erros, adaptar o software a alguma mudança no ambiente ou ainda melhorar a manutenibilidade ou confiabilidade futuras.

Outros cenários que se apresentam freqüentemente são decorrentes dos pedidos do usuário para *alterar ou excluir funções existentes, incluir novas funções e efetuar melhoramentos gerais*.

Num cenário de *inclusão de nova funcionalidade*, a análise dos pontos de função é vista como um novo desenvolvimento. A documentação mínima para auxiliar o processo de contagem é:

- ✓ Modelo de dados da melhoria
- ✓ *Layout* dos relatórios
- ✓ *Layout* das telas de entrada de dados
- ✓ *Layout* das telas de consulta
- ✓ *Layout* dos arquivos de interface
- ✓ *Opcionalmente* deveria também ser obtida a documentação da parte da aplicação que possa ser afetada pela melhoria.

No caso de *alteração ou exclusão da funcionalidade existente*, é preciso ter disponível a documentação do sistema existente, que vai ser afetado, bem como a especificação do que vai ser alterado. Como documentação necessária a esse cenário deve-se ter em mãos os mesmos citados na inclusão de funcionalidade com acréscimo dos seguintes documentos:

- ✓ *Layout* dos arquivos afetados mantidos pela aplicação
- ✓ *Layout* dos arquivos de interface afetados
- ✓ Impressão das telas de entrada de dados afetadas
- ✓ Impressão das telas de consulta afetadas
- ✓ Exemplos de cada relatório do sistema afetado

*Projetos de melhoria* normalmente possuem um escopo determinado, sendo possível, mesmo nas fases iniciais, obter informações necessárias para uma boa

contagem de pontos de função. Se esse não for o caso, é preferível esperar até que o projeto chegue a esse ponto.

Os projetos de *expansão e inclusão de campos em arquivos, mudança em telas para melhoria da visualização, mudança no Layout dos relatórios, adição de quebras e totalizações em relatórios e consultas*, são denominados *Production Support* e em geral não oferecem oportunidade de utilizar a técnica de pontos de função para sua estimativa, devido à natureza das mudanças, e também, muitas vezes, da sua simplicidade.

Para esse tipo de projeto, o ideal é a utilização da contagem de linhas de código e a experiência de quem está fazendo a manutenção. Caso o tamanho final da aplicação deva ser calculado após a manutenção, então a documentação mínima é a seguinte:

- ✓ *Layout dos arquivos internos afetados e/ou incluídos pela manutenção*
- ✓ *Layout dos arquivos de interface afetados e/ou incluídos pela manutenção*
- ✓ *Impressão das telas de entrada de dados afetados e/ou incluídas pela manutenção*
- ✓ *Impressão das telas de consulta afetadas e/ou incluídas pela manutenção*
- ✓ *Exemplos de cada relatório do sistema afetados e/ou incluídas pela manutenção*

Os projetos de *expansão e inclusão de campos em arquivos, alteração da aplicação para melhorar performance (uso de tabelas internas, bufferização, etc.), alteração da aplicação em função de novo realease (lançamento) de um produto e correção de um problema em produção* são projetos do tipo *Major Maintenance* e apresentam quase que a mesma característica dos projetos do tipo *Production Support*. Esses projetos também apresentam menos oportunidade de uso da análise de pontos de função.

Nesse caso também é ideal a utilização da contagem por linhas de código e a experiência de quem está fazendo a manutenção. Caso o tamanho final da aplicação após a manutenção deva ser calculada, deve-se utilizar a mesma documentação para projetos de *Production Support*.

## 5.1 Exemplo de Análise de Pontos de Função de um Projeto de Manutenção

A Análise de Pontos de Função de um projeto de manutenção envolve a contagem de três componentes de funcionalidade: *a funcionalidade da aplicação*, *a funcionalidade de conversão* e *a aplicação do fator de ajuste*.

A *funcionalidade da aplicação* é composta por pontos de função referentes à funcionalidade que é incluída no sistema, à funcionalidade que sofreu alteração, e à funcionalidade que foi excluída durante a manutenção.

A *funcionalidade de conversão* é o resultado da contagem dos pontos de função entregues na aplicação, como resultado da funcionalidade de conversão solicitada pelo usuário.

A *aplicação do fator de ajuste* é efetuada depois de determinados os números obtidos pela contagem das influências das 14 Características Gerais do Sistema, antes do início e depois do término da manutenção.

Para ilustrar a utilização da Análise de Pontos de Função em projetos de manutenção de aplicações, é apresentado a seguir um exemplo prático. As fórmulas para o cálculo são apresentadas no final do exemplo.

A documentação necessária para a contagem é:

- ✓ *Layout* dos arquivos internos afetados e/ou incluídas pela manutenção *Layout* dos arquivos de interface afetados e/ou incluídas pela manutenção.
- ✓ Impressão das telas de entrada de dados afetados e/ou incluídas pela manutenção.
- ✓ Impressão das telas de consulta afetadas e/ou incluídas pela manutenção.
- ✓ Exemplos de cada relatório do sistema afetados e/ou incluídas pela manutenção.

---

Uma aplicação deve ser alterada e foram previstas alterações, inclusões e exclusões de funcionalidade. As seguintes etapas serão cumpridas para a Análise de Pontos de Função do projeto de manutenção:

1. Descrição do escopo do projeto de manutenção do sistema
2. *Layout* das telas, arquivos e relatórios afetados, e/ou incluídos no sistema
3. Identificação das modificações, inclusões e exclusões no sistema
4. Classificação das funções incluídas, excluídas e modificadas no sistema
5. Cálculo dos pontos de função não ajustados das inclusões, exclusões e modificações no sistema
6. Cálculo do novo fator de ajuste dos pontos de função
7. Cálculo dos pontos de função ajustados do projeto de manutenção
8. Cálculo dos pontos de função ajustados após o projeto de manutenção

### 5.1.1 Descrição do escopo do projeto de manutenção do sistema

Descrição: O usuário solicitou alterações no sistema implantado. O principal objetivo da alteração é a inclusão da figura do vendedor dentro de certas transações e adicionar novas funções, descritas a seguir:

- ✓ Inclusão de um arquivo de vendedores.
- ✓ Alteração da tela de pedido, possibilitando a inclusão de um campo com o código do vendedor.
- ✓ Inclusão de um relatório listando os pedidos dos clientes, em uma data especificada.
- ✓ Sua execução será realizada através da tela de menu.
- ✓ Manutenção do cadastro de vendedores.
- ✓ Exclusão do relatório de clientes inativos do sistema.
- ✓ aplicativo será executado em um equipamento já existente, com vários outros aplicativos em paralelo. Serão necessários ajustes de performance no banco de dados, principalmente para suportar a transação de pedido de cliente, que é a mais utilizada no sistema.
- ✓ Deve ser incluída uma opção de visualização gráfica em vídeo dos cinco maiores clientes através da comparação do percentual das suas compras no ano corrente.

### 5.1.2 *Layout* das telas, arquivos e relatórios afetados, e/ou incluídos no sistema

Os arquivos, telas e relatórios afetados e/ou incluídos no sistema devem ser analisados. Os *layouts* das modificações e inclusões estão apresentados a seguir.

**Quadro 17** *Layout* do cadastro de pedidos (alterado)

<i>Pedidos</i>		
<i>Item de dados</i>	<i>Tamanho</i>	<i>Tipo</i>
Número-pedido	10	Numérico
Código-produto	12	Numérico
Código-cliente	6	Numérico
Quantidade	12	Numérico
Preço-unitário	12	Numérico
Data-pedido	8	Numérico
Código-vendedor	6	Numérico

**Quadro 18 Layout do cadastro de vendedores (incluído)**

<b>Vendedores</b>		
<b>Item de dados</b>	<b>Tamanho</b>	<b>Tipo</b>
Código-vendedor	6	Numérico
Nome-vendedor	30	Alfa
Data-admissão	8	Numérico
Cargo	15	Alfa
Filial	15	Alfa

**Figura 22 Layout da tela Pedido de Clientes (alterado)**

Pedido de clientes		
Número do pedido [ ]	Data pedido [ ]	
Código cliente [ ]	Código vendedor [ ]	
Código produto [ ]	Quantidade [ ]	Preço unitário [ ]

Arquivos referenciados: clientes, produtos, pedidos e vendedores

Dados: 7 campos

Regras: Verificar se existe pedido, produto, cliente e vendedor.

**Figura 23 Layout da tela Cadastramento de Vendedores**

Cadastramento de vendedores	
Operação [ ]	[I]nclusão
Código vendedor [ ]	[A]lteração
Nome vendedor [ ]	[C]onsulta
Data admissão [ ]	[E]xclusão
Cargo [ ]	
Filial [ ]	

Arquivos referenciados: vendedores

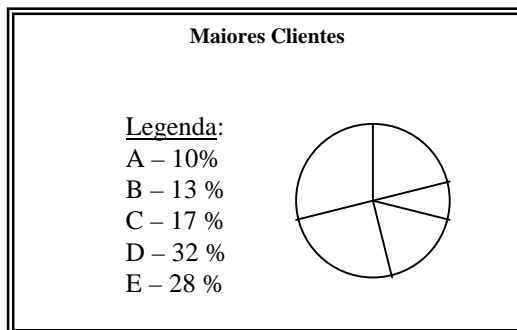
Dados: 5 campos

Regras: Verificar se a operação é válida e se o vendedor existe.

**Figura 24 Layout do Menu do sistema (alterado)**

Menu de Seleção		Auxílio [ ]
1.	Inclusão de clientes	
2.	Alteração de dados de clientes	
3.	Consulta ao cadastro de clientes	
4.	Exclusão de clientes	
5.	Inclusão de produtos	
6.	Alteração de dados de produtos	
7.	Consulta ao cadastro de produtos	
8.	Exclusão de produtos	
9.	Pedidos de clientes	
10.	Relação de pedidos	Data desejada [ ]
11.	Cadastramento de vendedores	
12.	Gráfico comparativo dos 5 maiores clientes	
Opção desejada [ ]		

**Figura 25 Layout da tela de Comparação entre os cinco maiores clientes**



Arquivos referenciados: *histórico de clientes* (para identificar os cinco clientes que possuem o maior volume de compras) e *clientes* (através do campo código do cliente do cadastro histórico de clientes, acessamos o nome do cliente no cadastro de clientes)

**Figura 26 Layout da Relação de Pedidos de Clientes**

Relação de Clientes		Data __/__/__	Página __
Data do pedido [ ]			
Número	Nome Cliente	Código produto	Valor do pedido
.....	.....	.....	.....
.....	.....	.....	.....

### 5.1.3 Identificação das modificações, inclusões e exclusões no sistema

A seguir (Quadro 19), a representação geral das modificações e inclusões de funções no sistema.

**Quadro 19 Modificações e inclusões de funções no sistema**

<b>Arquivos Lógicos Internos (ALI)</b>	Cadastro de clientes, Cadastro de produtos, Cadastro de pedidos (função alterada), Cadastro de vendedores (função incluída)
<b>Arquivos de Interface Externa (AIE)</b>	Histórico de clientes, Cadastro de senhas
<b>Entradas Externas (EE)</b>	Inclusão de clientes, Alteração de clientes, Exclusão de clientes, Inclusão de produtos, Alteração de produtos, Exclusão de produtos, Pedido de clientes (função alterada) Cadastramento de Vendedores – Inclusão (Função incluída), Cadastramento de Vendedores – Alteração (Função incluída), Cadastramento de Vendedores – Exclusão (Função incluída)
<b>Saídas Externas (SE)</b>	Relação de clientes, Relação de produtos, Relação de pedidos de clientes (Função incluída), Comparação entre os 5 maiores clientes (Função incluída)
<b>Consultas Externas (CE)</b>	Consulta de clientes, Consulta de produtos, Telas de auxílio, Relação de clientes inativos (Função excluída), Cadastramento de vendedores – Consulta (Função incluída)

Observações:

1. A tela de *Identificação* não é contada como função já que não provê segurança da aplicação, apenas segurança de acesso.
2. A tela *Menu do sistema*, como só provê seleção de telas, não é considerada como função.
3. A função *Relação de pedidos de clientes* é uma saída externa, pois possui dados derivados (campo valor-pedido, cujo valor é calculado pela multiplicação dos campos Quantidade e Preço-unitário).
4. A função *Comparação entre os cinco maiores clientes* possui dados derivados e mesmo com saída em vídeo é considerada uma Saída Externa.

### 5.1.4 Classificação das funções incluídas, excluídas e modificadas no sistema

As funções incluídas, excluídas e modificadas foram analisadas (Quadro 20) segundo as quantidades de Dados Elementares Referenciados (DER), Arquivos Lógicos Referenciados (ALR) e Registros Lógicos Referenciados (RLR).

**Quadro 20 Funções incluídas, excluídas e modificadas no sistema**

<i><b>Tipo de função</b></i>	<i><b>Função</b></i>	<i><b>Grau</b></i>
Arquivos Lógicos Internos	Cadastro de clientes DER: 15 RLR: 1	Simples
	Cadastro de produtos DER: 13 RLR: 1	Simples

<i><b>Tipo de função</b></i>	<i><b>Função</b></i>	<i><b>Grau</b></i>
------------------------------	----------------------	--------------------

Arquivos Lógicos Internos	Cadastro de pedidos (função alterada) DER: 7 RLR: 1	Simple
	Cadastro de vendedores (função incluída) DER: 5 RLR: 1	Simple
Arquivos de Interface Externas	Histórico de Clientes DER: 3 (código do cliente, total-compra-ano, total-compra-mês-atual) RLR: 1	Simple
	Arquivo de Senhas DER: 3 (usuário, senha e data de validade) RLR: 1	Simple
Entradas Externas	Inclusão de clientes DER: 16 (15 campos, mensagens de erro) ALR: 1	Média
	Alteração de clientes DER: 16 (15 campos, mensagens de erro) ALR: 1	Média
	Exclusão de clientes DER: 2 (código do cliente e mensagens de erro) ALR: 1	Simple
	Inclusão de produtos DER: 14 (13 campos, mensagens de erro) ALR: 1	Simple
	Alteração de produtos DER: 14 (13 campos, mensagens de erro) ALR: 1	Simple
	Exclusão de produtos DER: 2 (código de produto e mensagens de erro) ALR: 1	Simple
	Pedido de clientes (função alterada) DER: 8 (6 campos, mensagens de erro, código do vendedor) ALR: 4 (clientes, produtos, pedidos e vendedores)	Complexa
	Cadastramento de vendedores – Inclusão (função incluída) DER: 7 (5 campos, operação, mensagens de erro) ALR: 1 (vendedores)	Simple
	Cadastramento de vendedores – Alteração (função incluída) DER: 7 (5 campos, operação, mensagens de erro) ALR: 1 (vendedores)	Simple
	Cadastramento de vendedores – Exclusão (função incluída) DER: 3 (Código do vendedor, operação, mensagens de erro) ALR: 1 (vendedores)	Simple
Saídas Externas	Relação de clientes DER: 10 (9 campos e total de clientes) ALR: 1	Simple
	Relação de produtos DER: 7 ALR: 1	Simple
	Relação de pedidos de clientes DER: 5 (pedido-clientes e clientes, para acessar o nome-cliente) ALR: 2	Simple
	Comparação entre os cinco maiores clientes (função incluída) DER: 2 (nome-cliente, percentual-participação) ALR: 2 (clientes, histórico-clientes)	Simple

<b><i>Tipo de função</i></b>	<b><i>Função</i></b>	<b><i>Grau</i></b>
------------------------------	----------------------	--------------------



Consultas Externas	Consulta de clientes: Grau da parte de entrada: Simples DER: 2 (código do cliente e mensagens de erro) ALR: 2 (clientes e histórico de clientes) Grau da parte de saída: Média DER: 14 (12 de clientes e 2 de histórico de clientes) ALR: 2	Média
	Consulta a produtos: Grau da parte de entrada: Simples DER: 2 (código do produto e mensagens de erro) ALR: 1 (cadastramento de produtos) Grau da parte de saída: Simples DER: 12 (código do produto não é considerado) ALR: 1	Simples
	Relação de clientes inativos (função excluída)	Média
	Cadastramento de vendedores – Consulta (função incluída) Grau da parte de entrada: Simples DER: 3 (código do vendedor, mensagens de erro, operação) ALR: 1 Grau da parte de saída: Simples DER: 4 (nome-vendedor, data-admissão, cargo e filial) ALR: 1	Simples
	Telas de Auxílio <i>O conjunto de telas de auxílio é considerado por definição, uma consulta externa de grau simples, independentemente de sua quantidade.</i>	

Resumindo as funções modificadas e incluídas, temos o seguinte panorama:

#### ***Funções incluídas***

ALI	Cadastramento de vendedores	Grau Simples
EE	Cadastramento de vendedores – Inclusão	Grau Simples
	Cadastramento de vendedores – Alteração	Grau Simples
	Cadastramento de vendedores – Exclusão	Grau Simples
SE	Relação de pedidos	Grau Simples
	Comparação entre os cinco maiores clientes	Grau Simples
CE	Cadastramento de vendedores - Consulta	Grau Simples

#### ***Funções alteradas***

ALI	Cadastro de pedidos de clientes	Grau Simples
EE	Pedido de clientes	Grau Complexo

#### ***Funções excluídas***

CE	Relação de clientes inativos	Grau Médio
----	------------------------------	------------

### **5.1.5 Cálculo de PFs Brutos das modificações no sistema**

A contagem dos pontos de função não ajustados é efetuada separadamente para cada caso de modificação: funções incluídas, funções excluídas e funções alteradas (antes e depois do projeto de manutenção).

Tabela 7 Total de PFs das funções incluídas

<i>Tipo de função</i>	<i>Complexidade Funcional</i>		<i>Total Complexidade</i>	<i>Total tipo função</i>
<b>Arquivo Lógico Interno</b>	1 Simples	x 7 =	7	7
	0 Média	x 10 =	0	
	0 Complexa	x 15 =	0	
<b>Entradas Externas</b>	3 Simples	x 3 =	9	9
	0 Média	x 4 =	0	
	0 Complexa	x 6 =	0	
<b>Saídas Externas</b>	2 Simples	x 4 =	8	8
	0 Média	x 5 =	0	
	0 Complexa	x 7 =	0	
<b>Consultas Externas</b>	1 Simples	x 3 =	3	3
	0 Média	x 4 =	0	
	0 Complexa	x 6 =	0	
<b>Total de pontos de função não ajustados</b>				<b>27</b>

Tabela 8 Total de PFs das funções excluídas

<i>Tipo de função</i>	<i>Complexidade Funcional</i>		<i>Total Complexidade</i>	<i>Total tipo função</i>
<b>Consultas Externas</b>	0 Simples	x 3 =	0	4
	1 Média	x 4 =	4	
	0 Complexa	x 6 =	0	
<b>Total de pontos de função não ajustados</b>				<b>4</b>

Tabela 9 Total de PFs das funções alteradas – antes projeto de manutenção

<i>Tipo de função</i>	<i>Complexidade Funcional</i>		<i>Total Complexidade</i>	<i>Total tipo função</i>
<b>Arquivo Lógico Interno</b>	1 Simples	x 7 =	7	7
	0 Média	x 10 =	0	
	0 Complexa	x 15 =	0	
<b>Entradas Externas</b>	0 Simples	x 3 =	0	6
	0 Média	x 4 =	0	
	1 Complexa	x 6 =	6	
<b>Total de pontos de função não ajustados</b>				<b>13</b>

Tabela 10 Total de PFs das funções alteradas – após projeto de manutenção

<i>Tipo de função</i>	<i>Complexidade Funcional</i>		<i>Total Complexidade</i>	<i>Total tipo função</i>
<b>Arquivo Lógico Interno</b>	1 Simples	x 7 =	7	7
	0 Média	x 10 =	0	
	0 Complexa	x 15 =	0	
<b>Entradas Externas</b>	0 Simples	x 3 =	0	6
	0 Média	x 4 =	0	
	1 Complexa	x 6 =	6	
<b>Total de pontos de função não ajustados</b>				<b>13</b>

### 5.1.6 Cálculo do novo fator de ajuste dos pontos de função

O projeto de manutenção requer algumas considerações especiais, já que o sistema será executado em um equipamento que suporta outros aplicativos. Das 14 Características Gerais do Sistema, somente o fator Configuração do equipamento foi afetado. O novo grau desse item é 2. *Novo nível de influência = 17.*

$$\text{Fator de ajuste} = (NI * 0,01) + 0,65$$

$$\text{Fator de ajuste} = 17 * 0,01 + 0,65 = 0,82$$

### 5.1.7 Cálculo dos pontos de função ajustados do projeto de manutenção

A Equação 3 é utilizada para o cálculo do tamanho em pontos de função do projeto de manutenção é:

#### Equação 3 Cálculo de Pontos de Função do Projeto de Manutenção

$$PFM = [(INC + ALT + PFC) * FAD] + (EXC * FAA)$$

$$PFM = [(27 + 13 + 0) * 0,82] + (4 * 0,81)$$

$$PFM = 36,04, \text{ onde:}$$

PFM	36,04	Pontos de função do projeto de manutenção.
INC	27	Pontos de função brutos que foram incluídos na aplicação pelo projeto de manutenção. Refletem as funções que foram adicionadas à aplicação.
ALT	13	Pontos de função que foram alterados na aplicação do projeto de manutenção. Refletem as funções que sofreram alteração.
PFC	0	São os pontos de função que foram adicionados pelo processo de conversão.
FAD	0,82	Fator de ajuste da aplicação depois do projeto de manutenção.
EXC	4	Pontos de função brutos que foram excluídos da aplicação pelo projeto de manutenção.
FAA	0,81	Fator de ajuste da aplicação antes do projeto de manutenção.

### 5.1.8 Cálculo dos pontos de função ajustados após o projeto de manutenção

Quando um projeto de manutenção é terminado, as contagens de pontos de função da aplicação devem ser atualizadas para refletir as manutenções realizadas.

A funcionalidade da aplicação pode ser alterada de várias maneiras. Pode ocorrer a adição de novas funções, o que acarreta no aumento do tamanho da aplicação. A alteração de funções pode aumentar, diminuir ou mesmo não alterar o tamanho de uma aplicação. A exclusão de funções diminui o tamanho do sistema. A mudança no fator de ajuste também pode causar o aumento, diminuição ou mesmo não alterar o tamanho de uma aplicação.

A Equação 4 é utilizada para o cálculo dos pontos de função de uma aplicação após o projeto de manutenção.

#### Equação 4 Cálculo após Projeto de Manutenção

$$FPA = [(PFB + INC + ALTD) - (ALTA + EXC)] * FAD$$

$$FPA = [(79 + 27 + 13) - (13 + 4)] * 0,82$$

$$FPA = 102 * 0,82$$

$$FPA = 83,64, \text{ onde:}$$

FPA	83,64	São os pontos de função ajustados da aplicação após o projeto de manutenção.
PFB	79	São os pontos de função brutos da aplicação antes do projeto de manutenção.
INC	27	São os pontos de função brutos que foram adicionados pelo projeto de manutenção.
ALTD	13	São os pontos de função brutos correspondentes às funções que sofreram alteração depois do projeto de manutenção.
ALTA	13	São os pontos de função brutos correspondentes às funções que sofreram alteração antes do projeto de manutenção.
EXC	4	São os pontos de função brutos correspondentes às funções que foram excluídas da aplicação pelo projeto de manutenção.
FAD	0,82	É o fator de ajuste da aplicação verificado depois do projeto de manutenção.

---

## 6 PONTOS DE FUNÇÃO E ESTIMATIVAS

### 6.1 Estimativas de Tamanho de Projeto

O tamanho da aplicação que vai ser desenvolvida é a principal variável de entrada para o cálculo das estimativas de esforço, custo e tempo. Normalmente essa variável é expressa em unidade de milhares de linhas de código (KLOC) e é derivada da estimativa do tamanho dos módulos de software que irão constituir a aplicação ou programa.

Essa métrica é preferida por muitas organizações, pela facilidade com que pode ser contada e compreendida e por facilitar a contagem automatizada, porém é difícil prever o total de linhas de código logo no início do projeto, quando devem ser elaboradas as estimativas. Nesses casos, a alternativa para medir o tamanho do software pode ser através da *Análise de Pontos de Função*. Pontos de função podem ser utilizados como boas estimativas de tamanho porque são baseados em informações que estão disponíveis logo no início do ciclo de vida do projeto.

Técnicas e modelos de estimativa são implementados em software e constituem ferramentas importantes para o planejador estimar custo, esforço e cronograma e realizar análises dos resultados. Atualmente, um grande número de ferramentas de estimativa automatizadas estão disponíveis e representam um bom método para checagem com estimativas derivadas manualmente [Software, 1996].

Exemplos de ferramentas que implementam a *Análise de Pontos de Função* [Vigder, 1994]: *SPQR* (*Software Productivity, Quality, and Reliability*) Sizer/FP (*Function Points*), 1987, *BYL* (*Before You Leap*), 1986, *ASSET-R* (*Analytical Software Size Estimation Technique - Real Time*), 1987 e *QSM* (*Quantitative Software Management, Inc.*) *Size Planner*, 1987.

As ferramentas de software que implementam o modelo COCOMO<sup>5</sup> e os modelos *Early Design* e *Post-Architecture* do modelo COCOMO II permitem que o

---

<sup>5</sup>COCOMO (CONstructive COst Model) é uma hierarquia de modelos de estimativa de software que pode ser aplicado em três classes distintas de projetos de software: projetos de software simples, projetos de software de nível intermediário, e projetos de software com restrições operacionais, de hardware e de software [Boehm, 1981].

tamanho da aplicação possa ser estimado a partir de pontos de função não ajustados, posteriormente convertidos em linhas de código, de acordo com a linguagem de implementação escolhida (Assembly, linguagem de Quarta-Geração, etc.).

Exemplos de ferramentas que implementam os modelos COCOMO e COCOMO II são: REVIC (REVised Intermediate COCOMO) [Vigder, 1994], USC COCOMO [USC COCOMO, 2000], COSTAR [Softstar, 2000].

A Tabela 11 é usada para converter pontos de função não ajustados na quantidade equivalente de linhas de código [COCOMOII, 2000].

**Tabela 11 Conversão de Pontos de Função em Linhas de Código [COCOMOII 1999]**

<i>Linguagem</i>	<i>SLOC/UFP</i>	<i>Linguagem</i>	<i>SLOC/UFP</i>
Ada	71	ANSI Cobol 85	91
AI Shell	49	Fortran 77	105
APL	32	Forth	64
Assembly	320	Jovial	105
Assembly (Macro)	213	Lisp	64
ANSI/Quick/Turbo Basic	64	Modula2	80
Basic Compilado	91	Pascal	91
Basic Interpretado	128	Prolog	64
C	128	Report Generator	80
C++	29	Spreadsheet	6

### 6.1.1 A importância da Base de Dados Histórica

Para que as organizações possam efetuar estimativas mais precisas, é necessário que possuam informações registradas de projetos anteriores. A falta de informações de projetos passados e, às vezes, até mesmo de dados das estimativas iniciais do projeto atual, impede a análise sistemática por analogia entre projetos e qualquer tentativa de melhoria na precisão das estimativas [Vigder, 1994].

Além disso, os principais modelos de predição para software foram desenvolvidos a partir de análises do contexto de um conjunto de dados específico. Portanto, para utilização precisa, é necessário calibrar o modelo, ou mesmo customizá-lo, conforme a realidade da organização [Fenton, 1991], o que somente é possível se a organização possuir uma base histórica de observações.

A Tabela 12 mostra um exemplo de um conjunto de dados com uma série histórica hipotética.

Tabela 12 Base de dados histórico [Fernandes, 1995]

Plataforma de Hardware	: Maniframe			
Plataforma de Software	:			
Gerenciador de Banco de Dados	: DB2			
Monitor de TP	: CICS			
Linguagem de Programação	: CSP			
Processo	: Metodologia Estruturada			
Itens	Projetos			
	A	B	C	D
Tamanho	300	500	800	1200
Esforço H/M	10	16,67	32	60
Prazo (meses)	5	6	8	10
Custo \$	48,8	80,016	153	288
Equipe	2	3	4	6
Produtividade PF/H	30	30	25	20

O procedimento para elaborar estimativas é realizar inferências sobre a base de dados que contenha uma série histórica (observações) com informações acerca de projetos já concluídos, em termos de: plataforma de hardware/software, processo de desenvolvimento (metodologia empregada), prazo, esforço, custo e tamanho.

A cada novo projeto da organização, devem ser registrados os dados de estimativas tamanho, esforço e duração, dados do processo de estimativas e das eventuais suposições assumidas para o projeto, e no final também registrar os resultados reais de cada atividade de desenvolvimento que foi estimada. Essa abordagem é de grande auxílio na elaboração de estimativas mais precisas no futuro [Wieggers, 2000].

O procedimento para realizar inferência, a partir do tamanho do software em Pontos de Função é o seguinte [Fernandes, 1995]:

1. Determinar o Ambiente de Hardware/Software de desenvolvimento.

A inferência deve ser realizada para o mesmo ambiente em que será desenvolvido o software.

2. Determinar o *processo* que será utilizado para o desenvolvimento.

A inferência deve ser realizada para o mesmo processo que servirá de base para o desenvolvimento.

3. Determinar as *estimativas* para o projeto.

Para os valores de tamanho de software diferentes dos existentes no conjunto de dados e que estejam compreendidos entre as amplitudes 300 e 1200 pode-se empregar o método da interpolação aritmética<sup>6</sup>, utilizando-se a Equação 5.

**Equação 5 Interpolação Aritmética**

$$Y = Y_0 + \frac{X - X_0}{X_1 - X_0} (Y_1 - Y_0)$$

## 6.2 Método de Inferência para Estimativa do Prazo do Projeto

A seguir será apresentado um exemplo de inferência na base de dados histórico (Tabela 12), utilizando-se a equação da interpolação aritmética, de acordo com o procedimento especificado anteriormente.

---

Supondo uma estimativa de tamanho de 600 pontos de função (entre 500 e 800 da tabela), as variáveis da fórmula assumem os seguintes valores:  $Y_0 = 6$ ,  $Y_1 = 8$ ,  $X = 600$ ,  $X_0 = 500$  e  $X_1 = 800$ .

Aplicando-se a Equação 5:

$$Y = 6 + \frac{600 - 500}{800 - 500} * (8 - 6)$$

Obtém-se  $Y = 6,6$  meses.

---

## 6.3 Métodos de Inferência para Estimativa do Esforço do Projeto

O esforço estimado para um projeto é medido pela alocação dos recursos humanos em termos de homem/mês ou homem/hora. Pode-se efetuar a inferência na base de dados e obter a estimativa através da *transformação por interpolação aritmética* ou o *método da transformação direta*.

---

<sup>6</sup> A interpolação aritmética é uma forma de determinar valores equidistantes entre dois pontos numa tabela.



### 6.3.1 Transformação por Interpolação

Nesse caso, para cada projeto, deve estar contido no conjunto de dados as taxas de produtividade correspondentes medidas para cada projeto.

### 6.3.2 Transformação Direta

Caso o conjunto de dados possua dados registrados da a produtividade média de desenvolvimento, pode-se realizar transformação direta. A seguir é apresentado um exemplo de transformação direta.

---

Supondo-se que a produtividade média na plataforma *mainframe*, DB2, CICS e CSP seja Igual a 26,25 PFs/H/M e os pontos de função estimados sejam 500. O cálculo esperado é efetuado a partir da Equação 6.

#### Equação 6 Transformação Direta de PFs em Esforço

$$Esforço = \frac{(TamanhoEstimado do Software em PFs)}{(Produtividade Média em PFs / H / M)}$$

$$Esforço = \frac{(500 PFs)}{(26.26 PF / H / M)} \quad \text{ou } 19/h/m \text{ (por arredondamento).}$$

Transformando em horas, considerando a jornada de 168 horas/mês, obtém-se o valor aproximado de 3.192 homens/hora de esforço.

---

## 7 CONCLUSÃO

A partir de um modesto início como um conceito original de Allan Albrecht da IBM em 1979 até a criação de um grupo de trabalho na ISO sobre Medidas Funcionais de Tamanho em 1994, as medidas funcionais de tamanho (que incluem Pontos de Função) tornaram-se um dos principais assuntos da área de métricas de software.

O objetivo principal da *Análise de Pontos de Função* é medir a funcionalidade de um software ou aplicativo, baseando-se primeiramente no projeto lógico e, de acordo com a perspectiva do usuário, para que possa ser independente de tecnologia.

Os esforços promovidos pelo IFPUG para padronização e divulgação da técnica de análise de pontos de função tem contribuído para a evolução e melhoria das regras de contagem consistência entre os contadores.

Existem várias evidências de que pontos de função trazem benefícios às organizações. Com o crescente aumento da complexidade dos sistemas e, consequentemente, dos prazos de desenvolvimento, é necessário estimar, cada vez mais cedo, os prazos para cada fase do ciclo de desenvolvimento de sistemas. O tamanho do software é um parâmetro importante para os modelos de estimativa e pode ser determinado em *pontos de função logo* no início, a partir das especificações dos requisitos funcionais ou da especificação de projeto.

Com o aumento das atividades da área de informática, é reforçada a necessidade de se acompanhar a produtividade das equipes de desenvolvimento, produtividade na manutenção de sistemas e ainda um programa de monitoração da qualidade do software. Pontos de função podem prover uma unidade padrão de medida de software, permitindo a criação de métricas na área de produtividade e qualidade de software.

Furey [Furey, 1997] incentiva o uso de pontos de função., pois acredita-se que o uso repetitivo dessa métrica ajuda a normalizar os dados, possibilitando comparações a nível do âmbito do projeto e expectativas de cliente.

A Análise de Pontos de Função leva as organizações a normalizar seus dados como custo, duração, defeitos, previsão de pessoal, e assim por diante. Ela provê um modo melhor para comparar as organizações e a produtividade de projeto, especialmente em casos de tecnologias diferentes.

Com tempo e orçamento limitados, pontos de função possibilitam que gerentes tomem decisões mais objetivamente. Pontos de função ajudam gerentes de projeto a compreender melhor e articular com clientes os impactos de mudança e melhorias requeridas. Gerentes de projeto podem utilizar dados de pontos de função para negociar as mudanças, possivelmente adiando-as para uma futura entrega ou adicionando mais pessoas para trabalhar com as mudanças aceitas.

Várias instituições acadêmicas de renome têm adotado pontos de função no currículo de seus cursos de pós-graduação, nos programas de ciência da computação e nos cursos de MBA – *Mastering in Business Administration*. As instituições acadêmicas cuja adoção foi verificada pelo IFPUG são:

- ❑ California Polytechnic State University,  
San Luis Obispo, CA  
College of Business: Dr. Patricia McQuaid  
*Software Quality Management course*  
*Systems Analysis & Design course*  
College of Engineering, Computer Science Department: Dr. Sigurd Meldal,  
Dr. Dan Stearns *First course in Software Engineering*
- ❑ University of Pittsburgh  
Katz School: Dr. Chris Kemerer
- ❑ University of Southern California  
COMOCO II: Dr. Barry Boehm
- ❑ University of Tampa, Tampa, Florida  
Dr. Gordon Couturier  
*Graduate & Undergraduate: Systems Analysis & Design*
- ❑ Defense Acquisition University, Fort Lee, VA  
U. S. Army Logistics Management College:  
Major Mike Nelson  
*Software Cost Estimating*
- ❑ Carnegie Mellon University  
School of Computer Science, Masters Degree program: Dr. James Tomayko  
*Managing Software Development*
- ❑ University of Bari, Italy  
Department of Informatics: Dr. Giuseppe Visaggio
- ❑ Carleton University, Ontario, Canada  
Department of Systems and Computer Engineering: Dr. Lionel C. Briand
- ❑ University of Quebec at Montreal, Canada  
Software Engineering Management Research Laboratory: Dr. Alain Abran
- ❑ Monash University, Melbourne, Australia  
School of Business Systems: Mr. Kevin B.G. Luxford  
*Project Management*
- ❑ University of Washington, Seattle  
Computer and Science Engineering:  
Dr. Paul R. Stephens

*Brief component of a Project Management certificate program*

- ❑ UC Berkeley  
*The Art of a Successful Test Plan and S/W Metrics: Dr. Alka Shah-Jarvis*
- ❑ UC Santa Cruz Extension  
*S/w Metrics for S/W Project Mgt & Process Improvement: Mr. Ron Nusenoff*
- ❑ Air Force Institute of Technology  
*Wright-Patterson Air Force Base, Ohio: Mr. Daniel Ferens*

No Brasil também tem havido um crescente interesse por pontos de função e suas aplicações. O SERPRO, uma empresa de processamento de dados do Ministério da Fazenda, terminou recentemente a implantação da Análise de Pontos de Função na SUNAT (Superintendência de Negócios Administração Tributária). Também foi desenvolvida uma ferramenta de apoio ao processo de dimensionamento de sistemas usando a técnica de Análise de Pontos de Função, a qual foi chamada de Estimativa

A Unisys Eletrônica, empresa que promoveu o 1º ENUPF – Encontro Nacional de Usuários de Pontos de Função, em 1991, é outra empresa que se destaca no uso de pontos de função. Outras empresas que adotaram pontos de função são: COAMO, Objetiva, Varig, e WEG [Braga, 1996] e a CELEPAR - Companhia de Informática do Paraná.

Existem na literatura outras métricas funcionais que foram propostas para medir o tamanho funcional do software; entre elas os *Pontos de Característica* e os *Pontos de Função Mark II*.

*Feature Points* ou *Pontos de Característica* foram desenvolvidos em 1986 pelo *Software Productivity Research* (SPR) [FPFAQ, 2000]. Ela surgiu do reconhecimento de que algumas classes de aplicações não se adequavam à Análise de Pontos de Função, como as aplicações de controles de processos em tempo-real, otimização matemática e sistemas embutidos. Essa técnica considera o número de algoritmos usados na aplicação e modifica algumas das medidas que constituem os Pontos de Função tradicionais. A técnica foi documentada por Capers Jones [Jones, 1996]. Pontos de Característica não possuem o mesmo nível de aceitação que os Pontos de Função, nem possuem organizações de padronização tais como o IFPUG.

A técnica de *Pontos de Função Mark II* [FPFAQ, 2000] foi proposta para corrigir deficiências na técnica proposta por Albrecht. Symons desenvolveu a abordagem Mk II como forma de corrigi-las. Essa técnica surgiu do trabalho de

Charles Symons em 1984, contratado como consultor de metodologias da empresa Nolan, Norton & Co., uma parte da empresa *KPMG Management Consulting*, para clientes que queriam melhorar a performance em desenvolvimento de sistemas. Em 1987 a abordagem Mk II se tornou um produto licenciado.

## 8 REFERÊNCIAS BIBLIOGRÁFICAS

- [Albrecht, 1979] Albrecht, Allan J., "Measuring Application Development Productivity", *Proceedings SHARE/GUIDE IBM Applications Development Symposium*, Monterey, CA., Oct 14-17, 1979.
- [Azevedo, 1999] Azevedo, Douglas José Peixoto. Análise de Pontos por Função para Aplicações Orientadas a Documentos. *Dissertação de Mestrado*, UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL, 1999.
- [BFPUG, 2000] Brazilian Function Points Users Group [on-line] [05/02/2000] Disponível na Internet: <[www.bfpug.com.br](http://www.bfpug.com.br)>.
- [Boehm, 1981] BOEHM, B.W. *Software Engineering Economics*. Englewood Cliffs, NJ: Prentice-Hall, 1981.
- [Braga, 1996] BRAGA, Antônio. *Análise de pontos por função*. Rio de Janeiro: Infobook, 1996.
- [COCOMOII, 2000] COCOMO II [on-line] [05/02/2000] Disponível na Internet: <<http://sunset.usc.edu/index.html>>.
- [Conte, 1986] CONTE, S.D.; SHEN V.Y. *Software Engineering Metrics and Models*. Menlo Park, California: Benjamin/Cummings Publishing, 1985.
- [COSMOS, 2000] Software Cost Modeling Systems Version 4.1 *East Tennessee State University* [on-line] [05/02/2000] Disponível na Internet: <<http://csvaxsrv.east-tenn-st.edu/~dsgnstudio/cosmos/download.html>>.
- [Costar, 2000] Costar Version 5 *Softstar Systems* [on-line] [05/02/2000] Disponível na Internet: <<http://www.softstarsystems.com/demo.htm>>.
- [Dekkers, 1998] Carol, A. D. *Managing (the Size of) Your Projects*. A Project Management Look at Function Points. Quality Plus Technologies, Inc. [on-line] [05/02/2000] Disponível na Internet: <[www.bfgup.com.br](http://www.bfgup.com.br)>.
- [Fenton, 1991] Fenton, N. E. *Software Metrics: a rigorous approach*. Londres: Chapman & Hall, 1991.
- [Fernandes, 1995] FERNANDES A. A. *Gerência de Software através de métricas: garantindo a qualidade do projeto, processo e produto*. São Paulo: Atlas, 1995.
- [FPFAQ, 2000] Function Point Frequent Asked Questions [on-line] [05/02/2000] Disponível na Internet: <<http://ourworld.compuserve.com/homepages/softcomp/fpfaq.htm>>.
- [Furey, 1997] FUREY, Sean. Why We Should Use Function Points. *IEEE Software*. Los Alamitos, v.14, n.º 2, p.3, Mar./Apr. 1997.
- [IFPUG, 1994] Function Points Counting Practice Manual. Release 4.0. Westerville, Ohio, 1994 [on-line] [05/02/2000] Disponível na Internet: <[www.ifpug.org](http://www.ifpug.org)>.
- [IFPUG, 2000] International Function Points Users Group [on-line] [05/02/2000] Disponível na Internet: <[www.ifpug.org](http://www.ifpug.org)>.

- [Jones, 1996] JONES, Carpers. *Applied Software Measurement: assuring productivity and quality*. New York: McGraw-Hill, 1996.
- [Pressman, 1997] PRESSMAN, R. *Software Engineering: A Practitioner's Approach*. 4<sup>th</sup>. Ed. New York: McGraw-Hill, 1997.
- [Software, 1996] SOFTWARE, *Software Size, Cost and Schedule Estimation Process V 2.1*. [on-line]. Space and Naval Warfare Systems Center, Software Enginnering Process Office – SEPO, San Diego, 1996. [29/01/99]. Disponível na Internet: <http://sepo.nosc.mil/docs.html>.
- [USC COCOMO II, 1997] USC COCOMO II-2000 Version 0 [on-line] [05/02/2000] Disponível na Internet: [http://sunset.usc.edu/j\\_cocomo/cocomo.html](http://sunset.usc.edu/j_cocomo/cocomo.html).
- [Vigder, 1994] VIGDER, M. R.; Kark, A. W. Kark. *Software Cost Estimation and Control* [on-line]. National Research Council Canada, Institute for Information Technology, 1994. [03/02/99/]. Disponível na Internet: <http://wwwsel.iit.nrc.ca/abstracts/NRC37116.abs>.
- [Wieggers, 2000] Karl, W. Stop Promising Miracles. In: *Software Development*. p. 49-54 February 2000.