Aplicabilidade de Métricas por Pontos de Função em Sistemas Baseados em Web

Marcos David Drach

Trabalho Final de Mestrado Profissional

Universidade Estadual de Campinas

Instituto de Computação

TRABALHO FINAL
MESTRADO PROFISSIONAL

Aplicabilidade de Métricas por Pontos de Função em Sistemas Baseados em Web

Marcos David Drach

janeiro de 2005

Banca Examinadora:

- Profª Drª Ariadne Maria Brito Rizzoni Carvalho (Orientadora)
  Instituto de Computação, UNICAMP
- Profª. Thelma Cecília dos Santos Chiossi (Co-Orientadora)
  Instituto de Computação, UNICAMP
- Profª Drª Ana Cervigni Guerra
  Centro de Pesquisas Renato Archer, MCT
- Prof. Dr. Mário Lúcio Côrtes
  Instituto de Computação, UNICAMP
- Profª Dra. Cecília Mary Fischer Rubira (Suplente)
  Instituto de Computação, UNICAMP

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA DO IMECC DA UNICAMP

Drach, Marcos David

D78a Aplicabilidade de métricas por pontos de função em sistemas baseados em Web / Marcos David Drach -- Campinas, [S.P. :s.n.], 2005.

Orientadores : Ariadne Maria Brito Rizzoni Carvalho; Thelma Cecília dos Santos Chiossi

Trabalho final (mestrado profissional) - Universidade Estadual de Campinas, Instituto de Computação.

1. Engenharia de software. 2. Software – Estimativas. 3. Web sites - Desenho. I. Carvalho, Ariadne Maria Brito Rizzoni. II. Chiossi, Thelma Cecília dos Santos III. Universidade Estadual de Campinas. Instituto de Computação. IV. Título.

Aplicabilidade de Métricas por Pontos de Função em Sistemas Baseados em Web

Este exemplar corresponde à redação
do Trabalho Final submetida à Banca
Examinadora.

Campinas, janeiro de 2005.

Profª Ariadne Maria Brito Rizzoni Carvalho
(Orientadora)

Profª Thelma Cecília dos Santos Chiossi
(Co-Orientadora)

Trabalho Final apresentado ao Instituto de
Computação, UNICAMP, como requisito
parcial para a obtenção do título de
Mestre em Computação na área de
Engenharia de Computação

© Marcos David Drach, 2005
Todos os direitos reservados

Resumo

Métricas de software são padrões quantitativos de medidas de vários aspectos de um projeto ou produto de software, e se constitui em uma poderosa ferramenta gerencial, contribuindo para a elaboração de estimativas de prazo e custo mais precisas e para o estabelecimento de metas plausíveis, facilitando assim o processo de tomada de decisões e a subsequente obtenção de medidas de produtividade e qualidade.

A métrica de Análise por Pontos de Função - FPA, criada no final da década de 70 com o objetivo de medir o tamanho de software a partir de sua especificação funcional, foi considerada um avanço em relação ao método de contagem por Linhas de Código Fonte - SLOC, a única métrica de tamanho empregada na época.

Embora vários autores tenham desde então publicado várias extensões e alternativas ao método original no sentido de adequá-lo a sistemas específicos, sua aplicabilidade em sistemas Web ainda carece de um exame mais crítico.

Este trabalho tem por objetivo realizar uma análise das características computacionais específicas da plataforma Web que permita a desenvolvedores e gerentes de projeto avaliarem o grau de adequação da FPA a este tipo de ambiente e sua contribuição para extração de requisitos e estimativa de esforço.

Abstract

Software metrics are quantitative standards of measurement for many aspects of a software project or product, consisting of a powerful management tool that contributes to more accurate delivery time and cost estimates and to the establishment of feasible goals, facilitating both the decision-making process itself and the subsequent obtention of data measuring productivity and quality.

The metric Function Point Analysis - FPA, created at the end of 70's to measure software size in terms of its functional specification, was considered an advance over the Source Line Of Code - SLOC counting method, the only method available at that time.

Although many authors have published various extensions and alternatives to the original method, in order to adapt it to specific systems, its applicability in Web-based systems still requires a deeper and more critical examination.

This work aims to present an analysis of the specific computational characteristics of the Web platform that allows developers and project managers to evaluate the adequacy of the FPA method to this environment and its contribution to the requirement extraction and effort estimation.

Dedico aos meus pais, que me deram a base para sonhar,
e a minha esposa, por cujo amor continuo sonhando.

Agradecimentos

À minha esposa, Penha, pelo incentivo, carinho e paciência, sem os quais este projeto seria inviável.

Aos meus pais, Abel e Irinéa, pelo apoio infindável e torcida constante pelo sucesso.

Às Professoras Ariadne Carvalho e Thelma Chiossi, pela orientação e compreensão na elaboração deste trabalho.

À Kurt Greiner, pelo incentivo ao longo desta jornada.

Às funcionárias da Secretaria do Instituto de Computação da Unicamp, Cláudia e Olívia, sempre prestativas.

Conteúdo

Capítulo 1 - Introdução 1
1.1 Objetivo 3
1.2 Apresentação 3

Capítulo 2 - Métricas Funcionais de Software 4
2.1 Histórico 4
2.2 Métodos de Contagem 5
2.3 Evolução e Benefícios 7
2.3.1 Estimativas de Esforço 8
2.4 Aplicação no Brasil 13

Capítulo 3 - Análise por Ponto de Função 15
3.1 Método de Contagem 15
3.1.1 Determinação do Tipo de Contagem 16
3.1.2 Determinação do Escopo da Contagem e Identificação da Fronteira da Aplicação 17
3.1.3 Contagem das Funções Tipo Dado 17
3.1.4 Contagem das Funções Tipo Transação 18
3.1.5 Cálculo do Valor de Ponto de Função Não Ajustado 20
3.1.6 Determinação do Fator de Ajuste do Valor 21
3.1.7 Cálculo do Valor de Ponto de Função Ajustado 22
3.1.8 Simplificações 24
3.2 FPA na Extração de Requisitos 25
3.2.1 Verificação de Requisitos 26
3.2.2 Casos de Uso 29
3.2.3 Mudanças de Escopo 29
3.3 Estimativas de Esforço com FPA 29
3.3.1 Uso do Fator de Ajuste 30
3.3.2 Deficiências do Modelo 31
3.3.3 Indicadores de Mercado / Benchmarking 32

Capítulo 4 - Pontos de Função em Aplicações Web 34
4.1 Aplicações Web 34
4.1.1 Arquitetura 34
4.1.2 Particularidades 36
4.2 FPA em Aplicações Web 38
4.2.1 Diretrizes 39
4.2.2 Aplicabilidade 40
4.3 Considerações sobre Estimativas em Aplicações Web 42
4.4 Métodos de Contagem para Web 42
4.4.1 Método de Pontos de Caso de Uso 42
4.4.2 Método Web Points 46
4.4.3 Método Web Objects 47

Capítulo 5 - Estudo de Caso 52
5.1 Descrição do Problema 52
5.1.1 Identificação das Funcionalidades Principais 52
5.1.2 Descrição dos Casos de Uso 53
5.1.3 Interfaces do Usuário 56
5.1.4 Modelo de Dados 58
5.2 Metodologia Empregada para Contagem 60
5.3 Contagem pelo método IFPUG FPA 60
5.3.1 Identificação do Escopo da Contagem e Fronteira da Aplicação 60
5.3.2 Contagem das Funções Tipo Dado 60
5.3.3 Contagem das Funções Tipo Transação 64
5.3.4 Cálculo do Valor Não Ajustado 68
5.3.5 Cálculo do Fator de Ajuste 68
5.3.6 Cálculo do Valor Ajustado 69
5.4 Contagem pelo método de Pontos de Caso de Uso 70
5.5 Contagem pelo método Web Points 72
5.6 Dados Reais 73

Capítulo 6 - Conclusão 74
6.1 Trabalhos Futuros 76

Lista de Abreviações 77
Referências Bibliográficas 79

Apêndice A - Estudo de Caso - Telas do Sistema 82
Apêndice B - Estudo de Caso - Modelo de Dados 90

Lista de Tabelas

Tabela 2-1 Categorias de Modelos de Estimativa 10
Tabela 2-2 Métricas utilizadas para medir a produtividade dos processos de software - 2001 14
Tabela 3-1 FPA - Tabela de Complexidade da Função Tipo Dado 18
Tabela 3-2 FPA - Tabela de Complexidade da Função Tipo Transação EI 19
Tabela 3-3 FPA - Tabela de Complexidade da Função Tipo Transação EO e EQ 20
Tabela 3-4 FPA - Tabela de Conversão Complexidade x UFP 20
Tabela 3-5 FPA - Exemplo de Contagem 21
Tabela 3-6 FPA - Características Gerais do Sistema 22
Tabela 3-7 Exemplo de uso de FPA para validação dos requisitos 28
Tabela 4-1 Características de Projetos de Desenvolvimento Tradicionais vs Web 38
Tabela 4-2 Pontos de Caso de Uso - Peso dos Atores 43
Tabela 4-3 Pontos de Caso de Uso - Peso dos Casos de Uso 44
Tabela 4-4 Pontos de Caso de Uso - Fatores de Complexidade Técnica 45
Tabela 4-5 Pontos de Caso de Uso - Fatores de Ambiente 45
Tabela 4-6 Web Points - Tabela de Complexidade 46
Tabela 4-7 Web Points - Tabela de Conversão Complexidade x WP 47
Tabela 4-8 Web Points - Exemplo de Contagem 47
Tabela 4-9 Web Objects - Tabela de Pesos 49
Tabela 4-10 Web Objects - Exemplo de Contagem 50
Tabela 5-1 Estudo de Caso - FPA - Características Gerais do Sistema 69
Tabela 5-2 Estudo de Caso - UCP - Fatores Técnicos 71
Tabela 5-3 Estudo de Caso - UCP - Fatores de Ambiente 72
Tabela 5-4 Estudo de Caso - Web Points - Páginas HTML 72
Tabela 6-1 Estudo de Caso - Resultados 74

Lista de Figuras

Figura 2-1 Precisão das Estimativas 9
Figura 2-2 Processo de Estimativa 9
Figura 3-1 IFPUG FPA - Passos para Contagem 16
Figura 3-2 Mudança de Escopo com FPA 30
Figura 4-1 Arquitetura em N-Camadas 35
Figura 5-1 Estudo de Caso - Arquitetura do Sistema 54
Figura 5-2 Estudo de Caso - Diagrama de Casos de Uso 55
Figura 5-3 Estudo de Caso - Fluxo de Autenticação 57
Figura 6-1 Estudo de Caso - Mapa da interface 82
Figura 6-2 Estudo de Caso -Tela de Login 82
Figura 6-3 Estudo de Caso - Tela de Acesso a Áreas Restritas 83
Figura 6-4 Estudo de Caso - Tela de Acesso a Áreas Restritas - Assinante 83
Figura 6-5 Estudo de Caso - Tela de Acesso a Áreas de Acesso - Comprador em Banca 84
Figura 6-6 Estudo de Caso - Tela Central de Relacionamento 84
Figura 6-7 Estudo de Caso - Alterar Dados Cadastrais - P. Física 85
Figura 6-8 Estudo de Caso - Tela Alterar Dados Cadastrais - P. Jurídica 86
Figura 6-9 Estudo de Caso - Tela Alterar e-Mail de acesso 87
Figura 6-10 Estudo de Caso - Tela Alterar Senha de acesso 87
Figura 6-11 Estudo de Caso - Tela Recebimento de Newsletters 88
Figura 6-12 Estudo de Caso - Tela Ações de Marketing 88
Figura 6-13 Estudo de Caso - Tela Minhas Áreas de Acesso 89
Figura 6-14 Estudo de Caso - Tela Fale Conosco 89
Figura 6-15 Estudo de Caso - Modelo de Dados - Cadastro 90
Figura 6-16 Estudo de Caso - Modelo de Dados - Newsletter e Marketing 91
Figura 6-17 Estudo de Caso - Modelo de Dados - Autenticação 91
Figura 6-18 Estudo de Caso - Modelo de Dados - Pesquisa 92

Capítulo 1

Introdução

Medição é o processo de atribuir números ou símbolos aos atributos das entidades do mundo real, de acordo com um conjunto de regras claramente definidas. Portanto, uma medição captura informações de atributos de entidades, que são abstrações de objetos ou eventos do mundo real. Para tanto é necessário distinguir as entidades segundo certas propriedades que as identificam. Essas propriedades ou características, conhecidas por atributos, são expressas por números ou símbolos que nos permitem estabelecer comparações e obter conclusões a respeito das mesmas. [FEN 96].

A medição de software tem-se convertido em uma parte essencial da Engenharia de Software. Desenvolvedores de sistemas medem determinadas características do software para saber se os requisitos são consistentes e completos, se os projetos estão corretamente documentados ou se o código está pronto para ser testado. Gerentes de projeto medem atributos de processos e produtos para decidirem quando o software poderá ser entregue e se o projeto será executado dentro do prazo [FEN 96].

Métricas de software, portanto, são padrões quantitativos de medidas de vários aspectos de um projeto ou produto de software, e constituem-se em uma poderosa ferramenta gerencial, contribuindo para a elaboração de estimativas de prazo e custo mais precisas e para o estabelecimento de metas plausíveis, facilitando assim o processo de tomada de decisões e a posterior obtenção de medidas de produtividade e qualidade. Em geral, a aplicação de métricas é muito importante porque favorece três atividades básicas [BER 03]:

1. entendimento do que está ocorrendo durante o processo de desenvolvimento ou manutenção, o que permite estabelecer parâmetros de controle para futuros comportamentos;
2. controle do andamento de projetos com base em metas pré-estabelecidas ou dados históricos, permitindo a rápida atuação no sentido de corrigir eventuais desvios;
3. melhoria dos processos e produtos como, por exemplo, o aumento da quantidade ou tipo de revisões de projeto baseada nas medidas de qualidade da especificação.

Existem diversos tipos de métricas. Métricas de cronograma, por exemplo, medem, dentre outros parâmetros, o número de tarefas completadas no prazo. Métricas de requisitos, por sua vez, podem medir a percentagem de requisitos alterados, e a de testes, a percentagem de

cobertura. Há ainda métricas indiretas que medem atributos de qualidade como funcionalidade, complexidade, eficiência e confiabilidade, dentre outras.

No entanto, uma das métricas que tem recebido especial atenção é a de tamanho de software, considerada um importante direcionador de custo (*cost driver*), sobretudo por sua contribuição como uma das principais variáveis em modelos de estimativa de esforço e na obtenção de parâmetros de produtividade (tamanho / horas trabalhadas).

Existem duas abordagens para métricas de tamanho de software: orientadas à solução e orientadas ao problema [RUH 03b]. A primeira lida com o tamanho do software realmente produzido, enquanto a segunda lida com o tamanho do problema que o futuro software irá solucionar, ou seja, a funcionalidade alvo do sistema.

Métricas orientadas à solução refletem o tamanho dos artefatos produzidos, incluindo a especificação, o projeto ou, mais comumente, a implementação de um sistema, sendo o exemplo típico deste tipo a métrica de contagem por Linhas de Código Fonte - SLOC. Apesar de ser um método relativamente fácil de ser executado, possui diversas desvantagens, como:

- § falta de aceitação universal em relação a definições e regras de contagem de SLOC;
- § dependência de linguagem e estilo de programação;
- § dificuldade de se estimar o número de SLOC antes da implementação;
- § supervalorização da fase de codificação;
- § inadequação do método a novas linguagens e metodologias, como reuso de código e componentes pré-construídos.

Já as métricas orientadas ao problema definem elementos que podem ser contabilizados através dos requisitos do sistema, gerados nos estágios iniciais do ciclo de vida. O exemplo mais comum é a métrica por Análise por Pontos de Função - FPA, desenvolvida por Albrecht na IBM [ALB 79]. Ela captura o tamanho de um software através de suas características funcionais, independente da tecnologia empregada, o que deu origem a uma nova categoria de métricas, denominada Métricas de Tamanho Funcional - FSM, ou simplesmente, Métricas Funcionais. Por não possuir as desvantagens de SLOC, as Métricas Funcionais foram consideradas um avanço na metodologia de contagem, sobretudo por sua aplicabilidade em novas tecnologias [SYM 01].

Contudo, embora o método de Análise por Pontos de Função - FPA exista desde 1979, e vários autores tenham desde então publicado inúmeras extensões e alternativas no sentido de adequar o método a sistemas específicos, conforme detalhado na Seção 2.2, sua aplicabilidade em sistemas baseados em Web ainda carece de um exame mais crítico.

Além de questões como volatilidade de requisitos, componentização e outros, as

aplicações Web diferem daquelas convencionais pela existência de um processo de *authoring*, que corresponde às atividades de desenho, criação e publicação de conteúdos estáticos e componentes multimídia que compõem um Web site, tais como páginas HTML, folhas de estilo, imagens, animações e scripts para a camada de apresentação. Tal processo, que pode demandar uma quantidade considerável do esforço total do projeto, não é contabilizada pelo método de contagem tradicional.

A seguir apresentam-se o objetivo deste trabalho, seu conteúdo e por fim alguns trabalhos relacionados.

1.1 Objetivo

Este trabalho tem por objetivo realizar uma análise das características computacionais específicas da plataforma Web que permita avaliar o grau de adequação e aplicabilidade do método FPA a este tipo de ambiente e sua contribuição para extração de requisitos e estimativa de esforço.

1.2 Apresentação

Inicialmente, no Capítulo 2 serão apresentados os conceitos gerais de Métrica de Tamanho Funcional, seus benefícios e contribuição para a obtenção de estimativas de esforço. Também serão apresentados alguns métodos e discutida a evolução dos mesmos ao longo dos últimos 20 anos.

Em seguida, no Capítulo 3 será detalhado o método de contagem de FPA, delineadas algumas considerações relacionadas a extração de requisitos e estimativa de esforço com FPA, e por fim apresentadas algumas métricas alternativas para a plataforma Web.

As características e particularidades de aplicações Web, bem como a adequação de FPA neste ambiente, serão discutidas no Capítulo 4.

No Capítulo 5 será mostrado um estudo de caso onde serão aplicados alguns métodos de contagem em um aplicação Web para identificação de usuários.

Finalmente, no Capítulo 6 serão analisados os resultados e apresentadas as conclusões e sugestões para trabalhos futuros.

Capítulo 2

Métricas Funcionais de Software

Neste capítulo apresentaremos o conceito de Métricas Funcionais, categoria de métricas de software da qual o método de contagem de Análise por Ponto de Função - FPA faz parte, e descreveremos seus benefícios e sua aplicação no Brasil.

2.1 Histórico

A concepção de Métrica Funcional foi criada originalmente por Albrecht [ALB 79] no final da década de 70, através da introdução da métrica de Análise por Pontos de Função, com o objetivo de medir o tamanho de software a partir da especificação funcional.

A métrica por pontos de função de Albrecht foi impulsionada pela IBM, pois coincidia com suas estratégias de mercado na área de aplicações de negócio e Sistemas de Informação Gerenciais - MIS, o que levou à criação do Grupo Internacional de Usuários de Pontos de Função - IFPUG [IFP 86] em 1986, e à adoção do método pela maioria das grandes empresas e pelo governo americano, como também por tantas outras ao redor do mundo; no final dos anos 80 e início dos anos 90, o método atingiu seu ápice. A partir de então teve início um período de "desilusão" que se prolongou até meados dos anos 90, quando a FPA atingiu seu mínimo de aceitação pelo mercado. Dentre as razões para este declínio podemos citar: a deficiência dos programas de implantação que provocaram a baixa confiabilidade dos dados coletados, a falta de conscientização do nível gerencial e conseqüente dificuldade na compreensão dos resultados, e as ondas contínuas de aquisições e "terceirizações" (*outsourcing*), que resultaram na perda de conhecimento especializado no assunto [SYM 01].

Além destes, outro ponto que levou ao desuso de FPA é a defasagem do método frente aos conceitos dos "modernos" modelos de desenvolvimento, como Orientação a Objeto - OO, Interfaces Gráficas de Usuário - GUI e Casos de Uso. Além da dificuldade em se traduzir conceitos antigos para estes modelos, há ainda a questão dos fatores de ajuste relacionados ao ambiente, idealizados para a tecnologia empregada nos anos 70 e que não se mostram mais adequados ao cenário atual.

A partir de então tem-se notado uma recuperação e um emprego efetivo das Métricas Funcionais. Alguns fatores que contribuíram para esta recuperação foram, sem dúvida, o aparecimento de métodos compatíveis com modernas práticas de desenvolvimento e manutenção de software, e o surgimento de fóruns internacionais de conhecimento responsáveis pela definição de padrões, treinamento, atualização e suporte, o que tem melhorado a

divulgação e contribuído para um aumento no grau de confiabilidade por parte do mercado. O surgimento de empresas do setor privado, especializadas na aplicação de FSM, também tem sido importante para a consolidação da prática de utilização e da base de conhecimento.

2.2 Métodos de Contagem

Vários métodos então evoluíram a partir do método original de Albrecht com o intuito de suprir pontos falhos, ou ainda estender seu campo de aplicação [SYM 01]:

1. IFPUG: basicamente a evolução direta do método original de Albrecht e ainda hoje o método mais utilizado. Teve sua primeira versão lançada em 1990 [IFP 86] e em 2003 teve sua versão 4.1 (não ajustada) publicada como um Padrão Internacional pela ISO (ISO/EIC 20926:2003). Atualmente encontra-se na versão 4.2. Será detalhada na Seção 3.1;

2. Feature Points: proposto por Jones [JON 87], que adicionou ao método original um elemento para contagem de algoritmos matemáticos a fim de abranger sistemas de alta complexidade computacional. Contabiliza como fator de ajuste a complexidade lógica, de código e de dados, porém, a dificuldade do processo de medição impediu sua adoção em larga escala;

3. 3-D Function Point: proposto por Whitmire [WHI 92] da Boeing, que incluiu uma Dimensão de Controle ao método original, a fim de abranger o comportamento dinâmico de sistemas de tempo real; para tanto utiliza conceitos de Máquina de Estado Finito (FSM) para contabilizar o conjunto de estados iniciais e transições do sistema;

4. ISO FSM: proposto pelo Grupo de Trabalho da ISO (ISO/EIC JTC1 SC7 WG12), criado em 1996 para estabelecer princípios básicos e comuns de Métricas Funcionais. Compõe-se de 6 partes, das quais as 5 principais já foram publicadas (ISO/EIC 14143-1:1998,2-2002,3-2003,4-2002,5-2004); a última parte, que servirá como guia para as demais, encontra-se atualmente em desenvolvimento;

5. MkII FPA: proposto por Symons [SYM 98] e adotado pela *United Kingdom Software Metrics Association* - UKSMA, incluiu melhorias no modelo original, substituindo os elementos funcionais (descritos na Seção 3.1) por transações lógicas, compostas de dados de entrada, processos e dados de saída; desta forma utiliza uma granularidade mais fina para produzir resultados consistentes a partir de uma escala de tamanho de software onde

o método original alcança seu limite (cerca de 400 Pontos de Função, segundo Symons). Sua última versão (1.3.1) data de setembro de 1998 [UKS 98] e no final de 2002 foi publicada como um Padrão Internacional pela ISO (ISO/EIC 20968:2002);

6. Full Function Point - FFP V1: proposto por um grupo da Universidade de Quebec [ABR 98], incluiu novos elementos funcionais a fim de abranger softwares de tempo real. Apesar de utilizado com sucesso em várias indústrias, teve sua evolução interrompida em favor do método COSMIC FFP;

7. COSMIC FFP V2.0: criado pelo *Common Software Measurement International Consortium* - COSMIC em 1999 [ABR 99], para abordar igualmente MIS e softwares de tempo real, é baseado em duas fases: a de Mapeamento, na qual um modelo COSMIC-FFP dos Requisitos Funcionais do Usuário - UFR é gerado, e a de Medição propriamente dita, onde as regras de contagem são aplicadas ao modelo UFR para obtenção do tamanho funcional. Propõe, ainda, a decomposição dos requisitos funcionais do usuário em processos funcionais e subprocessos, compostos de quatro tipos de movimento de dados (entrada, saída, leitura e escrita); não contabiliza explicitamente algoritmos nem requisitos técnicos ou de qualidade; porém permite a medição de diferentes *Measurement Viewpoints*, ou seja, a medição dos componentes de uma arquitetura multicamada sob diferentes visões (visão do usuário, do desenvolvedor, etc.). Foi publicada como um Padrão Internacional pela ISO (ISO/EIC 19761:2003) e atualmente se encontra na versão 2003.

Todos os métodos anteriores tinham como base as idéias originais de Albrecht e foram chamados coletivamente, com exceção ao COSMIC FFP V2.0, de Métodos FSM de "1ª Geração". Além destes, alguns métodos alternativos para "novas" plataformas de desenvolvimento também têm sido propostos:

1. GUI Metrics: criada por Jeffery *et al.* [JEF 96], aborda o lado cliente da arquitetura cliente-servidor, utilizando características gráficas da interface do usuário para medir o volume do projeto, expressas em termos de caixas de diálogo, opções de menu, e outros elementos visuais;

2. Object Oriented Function Point - OOFP: proposto por Caldiera *et al.* [CAL 98], visando obter melhores estimativas de tamanho de projetos desenvolvidos segundo o paradigma de Orientação a Objeto. Tem como analogia central em relação ao método original o

mapeamento de Arquivos Lógicos e Transações (descritos na Seção 3.1) em Classes e Métodos;

3. Pontos de Caso de Uso: proposto por Smith [SMI 99], estabelece técnicas para estimativa de esforço baseadas em Casos de Uso. Uma descrição detalhada do método será apresentada na Seção 4.4.1;

4. Web Points: proposto por Cleary [CLE 00] como uma estimativa de tamanho e esforço para Web sites estáticos, realizada através da medição da complexidade de páginas HTML. Uma descrição detalhada será apresentada na Seção 4.4.2;

5. Web Objects: proposto por Reifer [REI 02a] especialmente para aplicações Web, estende a FPA com quatro novos componentes específicos deste ambiente: arquivos multimídia, blocos de construção Web, scripts e links. Após a aplicação em 64 projetos Web completos, a análise resultante demonstrou uma melhor precisão de estimativa do que a FPA tradicional. Uma descrição detalhada será apresentada na Seção 4.4.3;

2.3 Evolução e Benefícios

Ao longo de 20 anos tem-se percebido uma evolução dos métodos e vários fatores tem contribuído para este processo.

A eliminação de ambigüidades, notadamente presentes nos conceitos de MkII, COSMIC FFP e IFPUG, têm contribuído para a consistência dos resultados gerados por diferentes analistas sobre o mesmo documento de requisitos.

A adaptação dos métodos tradicionais a outros tipos de sistemas, como de tempo real e Orientados a Objeto, também tem sido uma prova da contínua evolução destes métodos em face aos novos modelos de desenvolvimento. Neste contexto, merece destaque a inclusão de diferentes "visões" e tamanhos, de acordo com os diferentes propósitos: o método original de Albrecht define a medida funcional sob o ponto de vista do usuário, ou seja, cria um modelo funcional como uma visão simplificada da interação humana com o software. Contudo, o advento da arquitetura N-camadas (ver Capítulo 4) tornou a aplicação exclusiva da visão externa do usuário uma medida de performance insustentável, o que levou ao desenvolvimento do conceito de medição por componente, presente no método COSMIC FFP V2. Desta forma, tornou-se possível distinguir diferentes visões (*e.g.* usuário-externo, principais componentes, componentes elementares), com diferentes tamanhos.

Por último, a mudança na forma de contagem dos requisitos de qualidade e fatores do ambiente tem contribuído para medidas mais realistas. A abordagem de Albrecht contabiliza

estes requisitos através do Fator de Ajuste de Valor - VAF (descrito na Seção 3.1.6), o que afeta diretamente o valor final do tamanho funcional. Há uma tendência, no entanto, de se considerar estes fatores quando da obtenção de estimativas de esforço, e não durante a medição do tamanho funcional, produzindo assim resultados mais consistentes.

De modo geral, as Métricas Funcionais têm proporcionado benefícios significativos como: suporte à produção de requisitos mensuráveis, não ambíguos e completos, apoio a análises de custo/benefício de novos projetos, normalização de dados *e benchmarking* de performance e riscos, possibilitando análises qualitativas e melhor definição de critérios de seleção, medição e controle de performance de fornecedores.

2.3.1 Estimativas de Esforço

Um dos principais benefícios obtidos com o uso de métricas de tamanho é o seu emprego em modelos de estimativa de esforço (prazo) ou custo.

Estimativas são um dos pontos chaves da gerência de projetos de desenvolvimento de software e sua obtenção enfrenta vários obstáculos, como objetivos conflitantes (custo, prazo, qualidade e escopo), exigência de estimativas em estágios iniciais do projeto, surgimento de novos processos de produção de software, falta de requisitos bem definidos e, sobretudo, falta de habilidade em se precisar o tamanho de sistemas.

Segundo Longstreet [LON 04b], a dificuldade na obtenção precisa de estimativas é inversamente proporcional ao tamanho do projeto, porém, da mesma forma também o é em relação à importância da estimativa. Ou seja, projetos pequenos são mais fáceis de estimar porém sem exigência de uma alta precisão, conforme mostra a Figura 2-1 [LON 04b].

O grau de precisão das estimativas também varia de acordo com a fase do projeto. É presumível supor que, à medida que um projeto avança e seus requisitos se tornam mais estáveis, menores serão as incertezas envolvidas e mais realista se torna a estimativa. Este efeito é representado na Figura 2-2 [LON 04b].

Abordagens

O primeiro passo natural para obtenção de estimativas é a definição do tamanho ou volume do programa. Para tanto, pode ser utilizada uma das abordagens descritas na Tabela 2-1, extraída de [PCE 99], e explicadas a seguir.

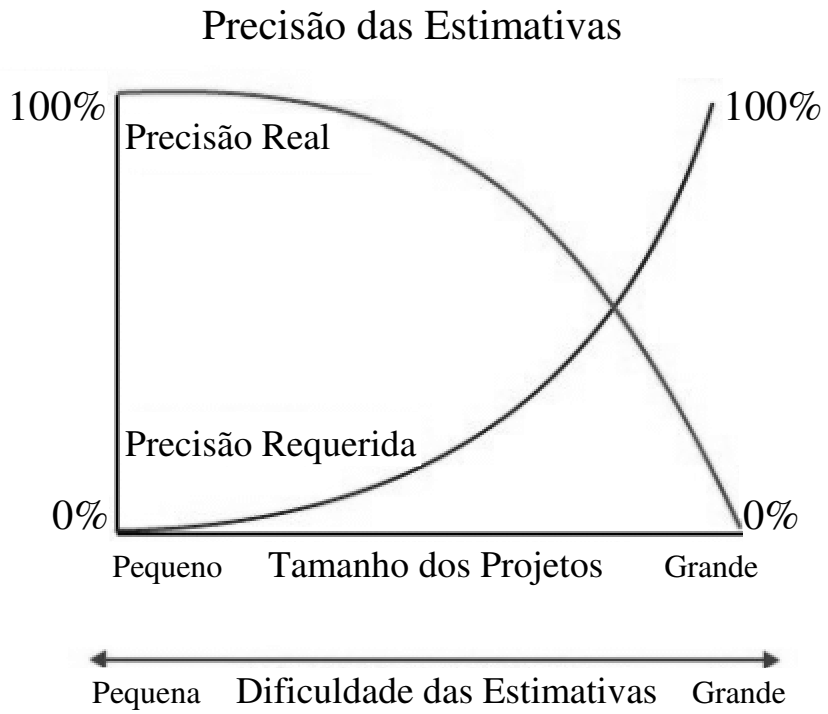

Figura 2-1 Precisão das Estimativas

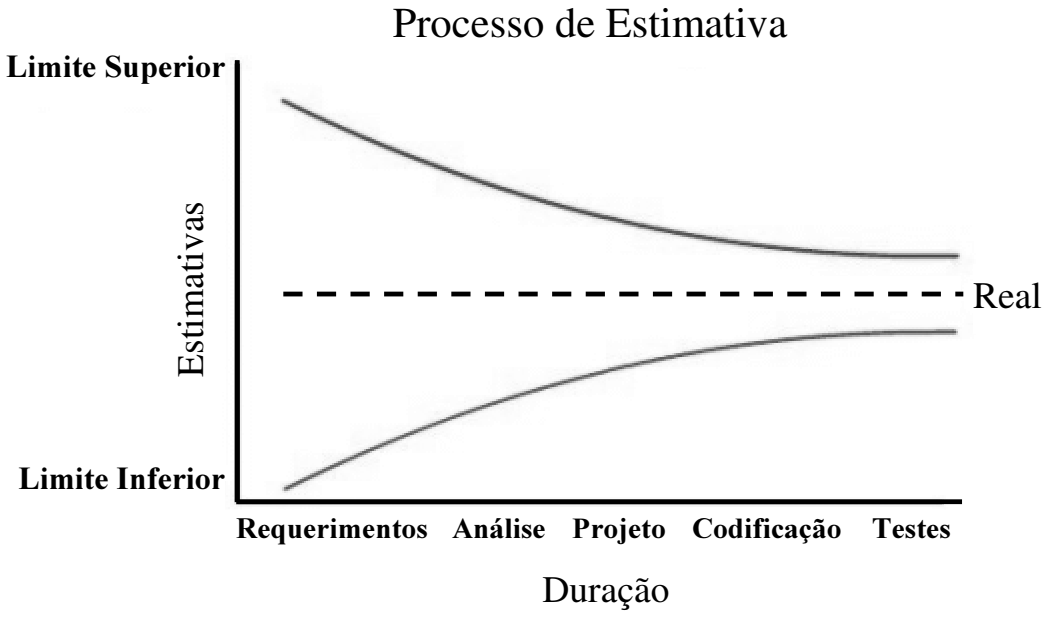

Figura 2-2 Processo de Estimativa

Tabela 2-1 Categorias de Modelos de Estimativa

Categorias	Descrição	Vantagens	Limitações
Analogia	Compara projetos com similares anteriores ou dados históricos	Estimativas são baseadas em experiências reais	Requer alto grau de similaridade entre projetos
Bottom-up	Componentes são avaliados e estimados individualmente e então somados para o cálculo total	Precisão das estimativas	Demanda maior tempo e requer que dados detalhados estejam disponíveis em estágios iniciais
Análise de Especialistas	Obtém informações através de especialistas	Pouco ou nenhum dado histórico necessário	Requer ausência de vícios e alto grau de conhecimento; subjetivo
Modelos Paramétricos	Utiliza parâmetros do projeto em algoritmos matemáticos	Rapidez e facilidade de uso; disponível em estágios iniciais; objetivos e repetitivos	Modelos podem ser imprecisos se não calibrados adequadamente; dados históricos podem não ser úteis para novas linguagens ou tecnologias

Estimativas por Analogia

Estimativas por Analogia consistem de comparações entre o sistema proposto e similares anteriores para os quais dados históricos estão disponíveis. Tem como vantagem a simplicidade e o uso de dados reais, porém requer a existência de uma base histórica. Além disso, o fato de que dois sistemas nunca são exatamente iguais deve ser considerado. Em geral são utilizadas em análises iniciais de viabilidade ou de razoabilidade de outras estimativas.

Estimativas *Bottom-up*

Estimativas *Bottom-up* são em geral muito detalhadas e demoradas. São utilizadas em geral durante a preparação de propostas para posterior acompanhamento dos custos. São as mais precisas e permitem a delegação de responsabilidades individuais para controle dos custos unitários do projeto. Custos associados com atividades de integração, que eventualmente se mostram significativos, podem não ser capturados.

Análise de Especialistas

Análise de Especialistas envolve a consulta de um ou mais *experts* através de técnicas populares, como entrevistas, painéis de discussão, ou ainda da técnica Delphi, que consiste na coleta individual de opiniões de diferentes especialistas sem influência mútua. Apesar de útil para situações novas, onde dados históricos não estão presentes, é pouco utilizada, pois além de muito subjetiva requer conhecimento específico do projeto e ausência de vício (que pode ser minimizada pelo Delphi).

Modelos Paramétricos

No modelo paramétrico, uma vez que o tamanho tenha sido estimado, uma série de fórmulas é usada para converter o tamanho em uma estimativa inicial não ajustada de esforço e duração. Devido ao fato de que cada projeto é único, calibrações e ajustes específicos do projeto devem ser realizados e abrangem: o ambiente de desenvolvimento, complexidade, restrições gerenciais e tolerância ao risco da organização. O resultado será uma estimativa ajustada do esforço e um cronograma ideal de entrega [ROE 00].

Existem muitos modelos paramétricos sofisticados, que utilizam múltiplos parâmetros para cálculos de custo e esforço, sendo os principais

1. COCOMO (*Constructive Cost Model*): criado por Barry Boehm [BOE 81] e publicado em 1981, tem por objetivo a estimativa de esforço (Homens-Hora) para desenvolvimento de sistemas, através de cálculos simples; não é proprietário, embora versões computadorizadas estejam disponíveis no mercado; o esforço nominal é medido em linhas de código fonte entregues (linhas de código sem comentários), e ajustado por alguns fatores relacionados a atributos do projeto e do produto, como necessidade de backup, existência de processamento distribuído e performance, dentre outros. Cada atributo é classificado por uma escala de 0 a 5 pontos e o resultado é convertido em valores quantitativos através de tabelas; o valor final é o Fator de Ajuste de Esforço - EAF, que é usado para calcular o cronograma de desenvolvimento. Tabelas adicionais são usadas para alocação de esforço por fase e atividades. Esse modelo define três diferentes modos de desenvolvimento, cada um com diferente coeficiente para cálculo de esforço, e três diferentes níveis de refinamento - básico (estimativa "grosseira"), intermediário (considera fatores relacionados ao ambiente), e detalhado (por fase). Possui tratamento para reuso de código;

2. COCOMO II: desenvolvido durante os meados de 1990 por um consórcio de organizações, também liderado por Boehm [BOE 95], foi lançado em 1996 com o propósito de desenvolver um modelo de estimativa de prazo e custo voltado às práticas de ciclo de vida mais atuais; enquanto a versão anterior foi largamente utilizada em projetos de desenvolvimento em cascata, COCOMO II é mais compatível com metodologias como Orientação a Objeto; aborda também a disponibilidade de informações em estágios posteriores, os custos não lineares de reuso de componentes e os efeitos de fatores econômicos. Também revisa alguns coeficientes e elimina descontinuidades presentes no modelo anterior. Utiliza três estágios ou modelos de estimativa: Estágio 1, que utiliza o Modelo de Composição de Aplicação e abrange prototipação para resolução de questões

de alto risco, como interface do usuário, interação entre sistemas, performance, etc.; Estágio 2, que utiliza o Modelo de Desenho Inicial e envolve a exploração de arquiteturas e conceitos de operação alternativos; utiliza FPA e *cost drivers* para estimativa; e Estágio 3, que utiliza o Modelo Pós-Arquitetura e abrange estimativas em estágios posteriores (desenvolvimento e manutenção);

3. PRICE S: desenvolvido pela empresa PRICE Systems, LLC [PRI 97], o modelo de custo de software PRICE (PRICE S) utiliza uma equação central baseada no volume de software a ser produzido e calcula o esforço nominal; este é ajustado por fatores não nominais (complexidade da interface, experiência da equipe, etc.) e utilizado para cálculo do cronograma de referência; após ajustes adicionais, o esforço e cronograma finais são calculados. Também possui abordagens para manutenção e reuso. Estudos indicaram uma precisão nas estimativas de 8 a 15 %. O modelo é aplicável a todos os tipos de projetos e considera todas as fases do ciclo de vida, incluindo as fases conceitual e de testes operacionais;

4. COBRA (*Cost Estimation, Benchmarking, and Risk Assessment*): desenvolvido por Briand *et. al* [BRI 98], trata-se de um método de estimativa de custo híbrido que combina a abordagem paramétrica com conhecimento de especialistas; inicialmente utiliza um modelo de relação causa-efeito que, baseado em opiniões de gerentes de projeto, captura os fatores direcionadores de custo mais importantes e quantifica suas relações, a fim de produzir um fator multiplicador a ser empregado sobre o custo nominal; após a repetição do processo em um certo número de projetos, e sua modelagem em curvas de distribuições, obtém-se uma estimativa que, aplicada a curvas de custo e produtividade resultante de projetos anteriores, pode ser utilizada não somente para dimensionamento de esforço, mas também para fins de análise de risco e *benchmarking*.

5. Taxa de Produtividade ou de Entrega: muito utilizada pelo mercado devido a sua simplicidade, consiste na aplicação direta sobre o Tamanho Funcional de uma Taxa de Produtividade [VAZ 04], definida como:

   Taxa de Produtividade = quantidade de Pontos de Função produzida por Homem-Hora de esforço.

   ou ainda do conceito inverso da Taxa de Entrega [VAZ 04], definida como:

Taxa de Entrega = esforço necessário (medido em Homens-Hora) necessário para se produzir um Ponto de Função.

O esforço total então pode ser calculado como:

Esforço = Tamanho Funcional / Taxa de Produtividade, ou
Esforço = Tamanho Funcional * Taxa de Entrega.

A utilização da Taxa de Produtividade (ou de Entrega) fornece uma estimativa de esforço que pode vir a ser muito próximo do real, dependendo do grau de calibração das taxas de acordo com as especificidades de cada caso. Em geral, este processo de calibração é tão mais exato quanto maior o histórico de estimativas da empresa e, assim como o processo de *Backfiring* detalhado na Seção 3.1.8, diversas instituições têm se dedicado a divulgar, através de pesquisas de *benchmarking*, valores médios de Taxa de Produtividade em função do Tamanho Funcional para vários tipos de projetos e empresas.

Especificamente em relação a aplicações Web, podemos citar dois modelos paramétricos desenvolvidos nos últimos anos:

1. WEBMO: proposto por Reifer [REI 02a] a partir de dados de 64 aplicações Web, é uma adaptação do modelo de estimativa COCOMO II para estimativa de esforço e duração para plataforma Web; a métrica utiliza nove fatores de custo e tamanho, medidos em Web Objects (descritos na Seção 4.4.3); utiliza técnica de regressão linear para determinação da estimativa de custo; o modelo tem sido calibrado e possui um nível de previsibilidade de 20% de erro relativo em no mínimo 68% dos casos;

2. WEB-COBRA: adaptação do modelo COBRA por Ruhe *et. al* [RUH 03a] para utilização em aplicações Web a partir da análise de dados de doze projetos, desenvolvidas em uma pequena empresa australiana; o resultado obtido mostrou um erro médio relativo de 17%.

2.4 Aplicação no Brasil

No Brasil a métrica SLOC foi a mais aplicada no passado, quando a utilização de código era dominante nas estimativas de custo. Desde a década de 1990 as Métricas Funcionais vêm ganhando espaço, como mostra a pesquisa "Qualidade no Setor de Software Brasileiro", conduzida pela Secretaria de Política de Informática do Ministério da Ciência e Tecnologia – MCT/SEPIN em 2001 [MCT 01], cujo resultado referente ao uso de métricas está descrito na Tabela 2-2.

No período de 1991 a 1994 foram realizados seis "Encontros Nacionais de Usuários de Ponto de Função - ENUPF"; porém, o interesse consolidou-se apenas quando grandes contratos públicos começaram a ser baseados em FPA. A criação do *Brazilian Function Point Users Group* - BFPUG [BFP 98], e do *chapter* do IFPUG no Brasil, em 1998, também vem contribuindo para a crescente consolidação do método no país.

Tabela 2-2 Métricas utilizadas para medir a produtividade dos processos de software - 2001

Categorias	Nº de organizações	% das organizações
Linhas de Código Fonte *(SLOC)*	46	10,3
Pontos de Função *(Function Point)*	81	18,2
Outras métricas	30	6,7
Não utiliza	312	70,0
Total	**446**	**100**

• • •

Uma vez apresentados os conceitos gerais de métricas funcionais e os métodos de contagem, iremos detalhar, no próximo capítulo, seu principal representante, a Análise por Pontos de Função.

Capítulo 3

Análise por Ponto de Função

Neste capítulo iremos examinar em detalhes a principal Métrica Funcional disponível, conhecida por Análise por Ponto de Função - FPA. Serão apresentados seu método de contagem e algumas simplificações utilizadas. Será explicado como a FPA pode beneficiar o processo de extração de requisitos e por fim serão discutidos importantes aspectos envolvendo FPA e estimativas de esforço.

3.1 Método de Contagem

A métrica de Análise por Pontos de Função - FPA se originou do método criado por Allan Albrecht em 1979 [ALB 79] para medição do Tamanho Funcional de sistemas, a partir de heurísticas derivadas de dados empíricos e aplicadas nas fases iniciais do ciclo de vida do software. Permite o cálculo do tamanho do software de forma independente da tecnologia empregada, tornando possível uma melhor comparação de performance entre projetos e a elaboração de estimativas em fases iniciais do projeto. Diversos métodos derivados têm surgido; porém, o principal e mais utilizado atualmente é aquele mantido e divulgado pelo Grupo Internacional de Usuários de Ponto de Função - IFPUG, baseado nas idéias originais de Albrecht.

Antes que o IFPUG iniciasse suas atividades, as informações sobre FPA se encontravam disseminadas por diversos livros e artigos, o que dificultava a padronização dos conceitos e a correta aplicação das regras de contagem, levando a grandes variações nos resultados. A partir da criação do IFPUG, o método foi devidamente documentado e têm sofrido evoluções e melhorias, permitindo que seja repetido e ensinado. Um estudo realizado em 1994, pelo Quality Assurance Institute e o IFPUG, revelou que a variação na contagem obtida entre profissionais treinados era de no máximo 11% [FUR 97].

Em novembro de 2003 o método foi publicado como um Padrão Internacional da ISO (ISO/IEC 20926:2003) e seu Manual Prático de Contagem - MPC encontra-se atualmente na versão 4.2.

A Figura 3-1, extraída de [VAZ 04], apresenta as etapas do processo de contagem e suas interdependências, de acordo com o MPC, que serão explicadas em detalhe nas próximas seções.

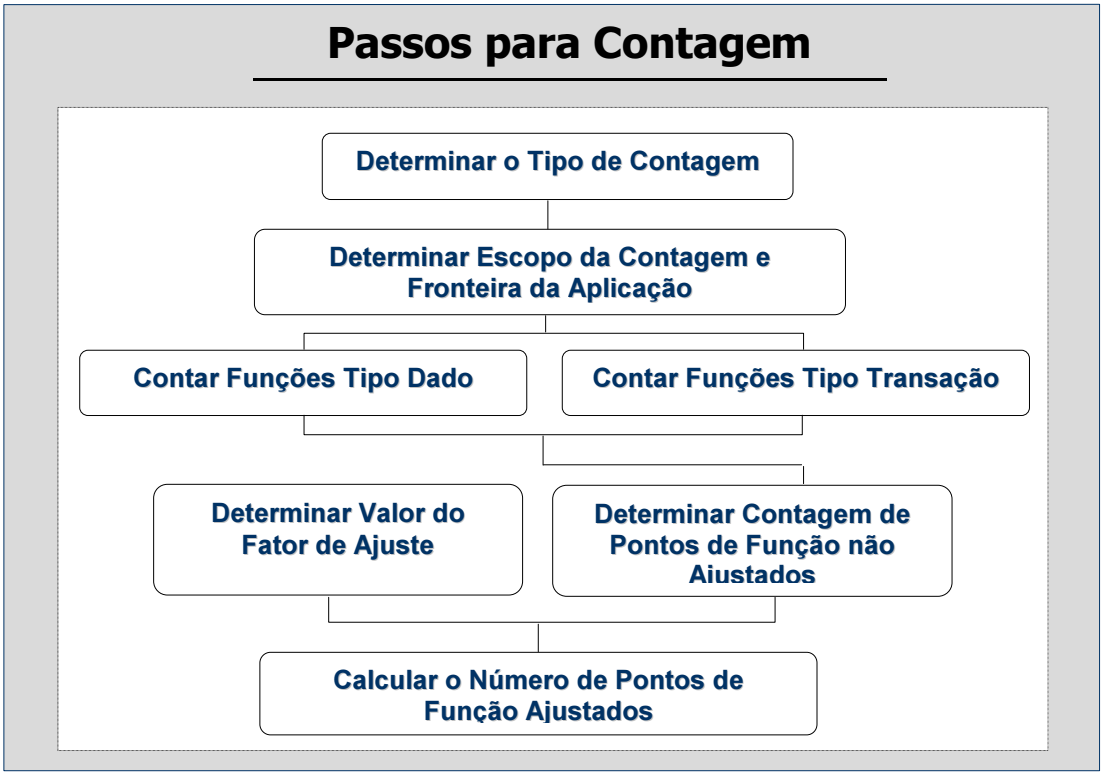

Figura 3-1 IFPUG FPA - Passos para Contagem

3.1.1 Determinação do Tipo de Contagem

O primeiro passo do processo de contagem é a determinação do tipo de contagem que será realizada. Os três tipos são os seguintes:

1. Contagem de Projetos de Desenvolvimento: mede a funcionalidade fornecida aos usuários pela primeira instalação da aplicação; inclui novas aplicações, bem como conversão de dados; subseqüentes medições podem ser realizadas à medida que novas atualizações são incorporadas ao projeto (*scope creep*);

2. Contagem de Projetos de Melhoria: mede a modificação de projetos já existentes, envolvendo adição, alteração ou remoção de funções, além de conversão de dados;

3. Contagem de Aplicações: também conhecida como base de referência (*baseline*) ou contagem instalada, mede uma aplicação já implantada a fim de fornecer um valor para as funcionalidades disponíveis no momento.

Como será visto na Seção 3.1.7, o tipo de contagem determinará a fórmula a ser empregada no cálculo do valor final do Ponto de Função Ajustado.

3.1.2 Determinação do Escopo da Contagem e Identificação da Fronteira da Aplicação

O escopo da contagem é determinado pelo propósito da contagem e especifica sua abrangência, que pode variar de algumas funcionalidades específicas (*e.g.* relatórios) a todas aquelas disponíveis na aplicação.

A fronteira da aplicação é a interface conceitual entre a aplicação sendo medida e os usuários ou as demais aplicações. O IFPUG define as seguintes regras para identificação da fronteira:

- § a fronteira é baseada na visão do usuário;
- § a fronteira entre aplicações relacionadas deve ser baseada nas regras de negócio e não em considerações técnicas;
- § a fronteira inicial já estabelecida para uma aplicação não é influenciada pelo escopo da contagem.

3.1.3 Contagem das Funções Tipo Dado

As funções de dados representam as funcionalidades fornecidas ao usuário que atendem aos requisitos de dados internos e externos.

Elementos da Contagem

Os elementos da contagem para funções tipo Dado são definidos como:

1. Arquivos Lógicos Internos - ILF: grupo de dados logicamente relacionados ou informações de controle identificável pelo usuário e mantidos dentro da fronteira da aplicação, cujo objetivo principal seja o armazenamento de dados mantidos através de um ou mais processos elementares da aplicação sendo medida;

2. Arquivos de Interface Externa - EIF: grupo de dados logicamente relacionados ou informações de controle identificável pelo usuário e referenciado por uma aplicação, mas mantido dentro da fronteira de outra aplicação, cujo objetivo principal seja o armazenamento de dados referenciados através de um ou mais processos elementares dentro da aplicação sendo medida; um EIF contado para uma aplicação deve ser contado como um ILF de uma outra aplicação.

O termo "arquivo" se refere a um grupo lógico de dados e não a sua implementação física. O termo "informação de controle" se refere a dados que influenciam um processo elementar da aplicação sendo contada, ou seja, especificam o que, quando e como os dados serão

processados. O termo "processo elementar" se refere a menor unidade de atividade significativa para o usuário. O termo "mantido" se refere à habilidade de se modificar dados através de processos elementares (*e.g.* inclusão, alteração, remoção, criação).

Procedimento de Contagem

Após a identificação dos ILFs e EIFs, deve-se determinar a complexidade de cada tipo funcional, baseada na contagem dos seguintes tipos elementares:

- § Tipos de Elementos de Dado - DET: campo único não repetido e reconhecido pelo usuário;
- § Tipos de Elementos de Registro - RET: subgrupo único de elementos de dados não repetido e reconhecido pelo usuário dentro de um ILF ou EIF.

Após a contagem de DETs e RETs aplica-se a Tabela 3-1 para obtenção do Grau de Complexidade Funcional.

Tabela 3-1 FPA - Tabela de Complexidade da Função Tipo Dado

Núm. Elementos	1 a 19 DETs	20 a 50 DETs	51 ou mais DETs
1 RET	Baixo	Baixo	Médio
2 a 5 RETs	Baixo	Médio	Alto
6 ou mais RETs	Médio	Alto	Alto

Por exemplo, um IFL com uma contagem de 30 DETs e 3 RETs deve ser classificado como de Complexidade Média.

3.1.4 Contagem das Funções Tipo Transação

As funções de transação representam as funcionalidades fornecidas ao usuário para o processamento de dados de uma aplicação.

Elementos da Contagem

Os elementos da contagem para funções tipo Transação são definidos como:

1. Entrada Externa - EI: processo elementar de dados ou informações de controle que se originam de fora da fronteira da aplicação, cujo objetivo principal seja a manutenção de um ou mais ILFs e/ou alteração do comportamento do sistema;

2. Saída Externa - EO: processo elementar de dados ou informações de controle que são enviados para fora da fronteira da aplicação, cujo objetivo principal seja a

apresentação de informações ao usuário através de lógica de processamento outra que, ou além da, recuperação de dados ou controle de informação; a lógica de processamento deve conter no mínimo uma fórmula ou cálculo matemático ou criação de dados derivados; deve também manter uma ou mais ILFs e/ou alterar o comportamento do sistema;

3. Consulta Externa - EQ: processo elementar de dados ou informações de controle que são enviados para fora da fronteira da aplicação, cujo objetivo principal seja a apresentação de informações ao usuário através de recuperação de dados ou informações de controle de um ILF ou EIF; a lógica de processamento não deve conter fórmulas ou cálculos matemáticos, nem a criação de dados derivados; nenhum ILF é mantido, nem o comportamento do sistema é alterado.

Procedimento de Contagem

Após a identificação dos EIs, EOs e EQs deve-se determinar a complexidade de cada tipo funcional, baseada na contagem dos seguintes tipos elementares:

§ Tipos de Elementos de Dado - DET: campo único não repetido e reconhecido pelo usuário;
§ Tipos de Arquivo Referenciado - FTR: ILF lido ou mantido por uma função transacional, ou um EIF lido por uma função transacional.

Após a contagem de DETs e FTRs aplicam-se as Tabela 3-2 e Tabela 3-3 para obtenção do Grau de Complexidade Funcional.

Tabela 3-2 FPA - Tabela de Complexidade da Função Tipo Transação EI

Entrada Externa			
Núm. Elementos	**1 a 4 DETs**	**5 a 15 DETs**	**16 ou mais DETs**
0 ou 1 FTR	Baixo	Baixo	Médio
2 FTRs	Baixo	Médio	Alto
3 ou mais FTRs	Médio	Alto	Alto

Por exemplo, uma Saída Externa com uma contagem de 15 DETs e 5 FTRs deve ser classificada como de Complexidade Alta.

Tabela 3-3 FPA - Tabela de Complexidade da Função Tipo Transação EO e EQ

Saída Externa ou Consulta Externa			
Núm. Elementos	**1 a 5 DETs**	**6 a 19 DETs**	**20 ou mais DETs**
0 ou 1 FTR	Baixo	Baixo	Médio
2 ou 3 FTRs	Baixo	Médio	Alto
4 ou mais FTRs	Médio	Alto	Alto

3.1.5 Cálculo do Valor de Ponto de Função Não Ajustado

Baseado no Grau de Complexidade dos tipos funcionais obtém-se o valor de Pontos de Função Não Ajustados - UFP para cada elemento funcional, de acordo com os valores apresentados na Tabela 3-4.

Tabela 3-4 FPA - Tabela de Conversão Complexidade x UFP

Grau de Complexidade	UFP				
	ILF_W	EIF_W	EI_W	EO_W	EQ_W
Baixo (B)	7	5	3	4	3
Médio (M)	10	7	4	5	4
Alto (A)	15	10	6	7	6

O valor de UFP de cada elemento funcional é calculado multiplicando-se o número de elementos encontrados para cada complexidade pelo respectivo número de Pontos de Função equivalente. Por fim o valor total de UFP é então calculado somando-se os valores de UFPs obtidos para cada elemento funcional.

Portanto:

$$
\begin{aligned}
UFP_{ILF} &= ILF_{WB}*\textstyle\sum ILF_B + ILF_{WM}*\sum ILF_M + ILF_{WA}*\sum ILF_A \\
UFP_{EIF} &= EIF_{WB}*\textstyle\sum EIF_B + EIF_{WM}*\sum EIF_M + EIF_{WA}*\sum EIF_A \\
UFP_{EI} &= EI_{WB}*\textstyle\sum EI_B + EI_{WM}*\sum EI_M + EI_{WA}*\sum EI_A \\
UFP_{EO} &= EO_{WB}*\textstyle\sum EO_B + EO_{WM}*\sum EO_M + EO_{WA}*\sum EO_A \\
UFP_{EQ} &= EQ_{WB}*\textstyle\sum EQ_B + EQ_{WM}*\sum EQ_M + EQ_{WA}*\sum EQ_A
\end{aligned}
$$

$$UFP = UFP_{ILF} + UFP_{EIF} + UFP_{EI} + UFP_{EO} + UFP_{EQ}$$

Como exemplo, suponha a contagem apresentada na Tabela 3-5.

Tabela 3-5 FPA - Exemplo de Contagem

Tipo de Elemento	Quant. Elementos		
	Baixo	Médio	Alto
ILF	3	2	1
EIF	1	2	3
EI	5	4	2
EQ	3	5	1
EO	3	4	2

Temos, portanto:

$$\begin{aligned} UFP_{ILF} &= 7*3 + 10*3 + 15*1 = 56 \\ UFP_{EIF} &= 5*1 + 7*2 + 10*3 = 49 \\ UFP_{EI} &= 3*5 + 4*4 + 6*2 = 43 \\ UFP_{EQ} &= 3*3 + 4*5 + 6*1 = 35 \\ UFP_{EO} &= 4*3 + 5*4 + 7*2 = 46 \end{aligned}$$

E finalmente:

$$UFP = 56 + 49 + 43 + 35 + 46 = 229.$$

O valor de UFP pode ser considerado a primeira estimativa válida de Pontos de Função. Seu valor poderá ainda ser ajustado para refletir fatores do ambiente e técnicos que influenciam de alguma forma o Tamanho Funcional da aplicação, como será visto nas seções seguintes.

3.1.6 Determinação do Fator de Ajuste do Valor

O Fator de Ajuste do Valor - VAF é baseado em 14 Características Gerais do Sistema (GSC), que medem o Grau Total de Influência - TDI do ambiente. Cada característica está associada a uma descrição que auxilia a determinação do grau de influência. Um peso, que varia de 0 (sem influência) a 5 (forte influência), deve ser atribuído a cada GSC e a soma desses resulta no valor do TDI. As 14 GSCs, apresentadas na Tabela 3-6, são então resumidas no VAF que, quando aplicado, corrige o valor de UFP em cerca de +/-35 %, produzindo o valor de Pontos de Função Ajustado - AFP.

Temos, portanto:

$$TDI = \sum_{i=1}^{14} peso_i \text{ (CGS)}$$

O valor de VAF é então calculado pela fórmula:

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

Por exemplo, se cada uma das 14 GSCs forem avaliadas com um Grau de Influência de peso 3, então temos que:

$$TDI = 3 * 14 = 42$$
$$VAF = (42 * 0,01) + 0,65 = 1,07$$

Tabela 3-6 FPA - Características Gerais do Sistema

Características Gerais do Sistema	
1. Comunicação de Dados	8. Atualização *Online*
2. Processamento de Dados Distribuídos	9. Processamento Complexo
3. Performance	10. Reusabilidade
4. Configuração Altamente Utilizada	11. Facilidade de Instalação
5. Taxa de Transações	12. Facilidade de Operação
6. Entrada de Dados *Online*	13. Múltiplos Sites
7. Eficiência do Usuário Final	14. Facilidade de Alteração

3.1.7 Cálculo do Valor de Ponto de Função Ajustado

A fórmula para cálculo do valor do Ponto de Função Ajustado depende do tipo de contagem adotado, conforme visto na Seção 3.1.1.

1. Projetos de Desenvolvimento

Requerem que os pontos de função para as funções disponíveis após a implantação e para as funções de conversão de dados sejam contabilizados de forma separada, conforme a fórmula apresenta a seguir:

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

onde:

DFP: Valor de Pontos de Função Ajustado do projeto de desenvolvimento;

UFP: Valor de Pontos de Função Não Ajustado das funções disponíveis após a implantação;

CFP: Valor de Pontos de Função Não Ajustado das funções de conversão;

VAF: Fator de Ajuste do Valor.

Tomando-se os exemplos anteriores para os valores de UFP (229), VAF (1,07) e considerando um valor de CFP igual a 10, tem-se que:

$$DFP = (229 + 10) * 1,07 = 256.$$

2. Projetos de Melhoria

Nestes casos devem-se contabilizar as funções incluídas, modificadas e as de conversão de dados disponibilizadas após o processo de melhoria, de acordo com os novos fatores de ajuste, e também aquelas removidas segundo os fatores anteriores, conforme a fórmula apresenta a seguir

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

onde:

- EFP: Valor de Pontos de Função Ajustado do projeto de melhoria;
- ADD: Valor de Pontos de Função Não Ajustado das funções incluídas;
- CHGA: Valor de Pontos de Função Não Ajustado das funções modificadas, contabilizadas após o projeto de melhoria;
- CFP: Valor de Pontos de Função Não Ajustado das funções de conversão de dados;
- VAFA: Fator de Ajuste do Valor após o projeto de melhoria;
- DEL: Valor de Pontos de Função Não Ajustado das funções excluídas;
- VAFB: Fator de Ajuste do Valor antes do projeto de melhoria.

3. Aplicação

Para a contagem de aplicações, devem ser consideradas duas situações distintas:

3.1 Contagem Inicial

Utilizada para estabelecer a contagem inicial de uma aplicação, considera apenas as funções incluídas, conforme a fórmula apresenta a seguir

$$AFP = ADD * VAF$$

onde:

- AFP: Valor de Pontos de Função Ajustado inicial da aplicação;
- ADD: Valor de Pontos de Função Não Ajustado das funções incluídas;
- VAF: Fator de Ajuste do Valor.

3.2 Contagem Após Projeto de Melhoria

Utilizada para a atualização da contagem de uma aplicação após um projeto de melhoria. Visa corrigir o valor de Pontos de Função Ajustado anterior considerando as novas funções incluídas, modificadas e removidas, de acordo com os novos fatores de ajuste, conforme a fórmula apresentada a seguir:

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

onde:

- AFP: Valor de Pontos de Função Ajustado da aplicação;
- UFPB: Valor de Pontos de Função Não Ajustado contabilizado antes do projeto de melhoria;
- ADD: Valor de Pontos de Função Não Ajustado das funções incluídas;
- CHGA: Valor de Pontos de Função Não Ajustado das funções modificadas, contabilizadas após o projeto de melhoria;
- CHGB: Valor de Pontos de Função Não Ajustado das funções modificadas, contabilizadas antes do projeto de melhoria;
- DEL: Valor de Pontos de Função Não Ajustado das funções excluídas;
- VAFA: Fator de Ajuste do Valor após o projeto de melhoria.

3.1.8 Simplificações

A contagem original, apesar de detalhada, demanda um tempo considerável para sua realização, além de requerer um nível considerável de especificação dos requisitos do projeto. A fim de simplificar o processo de obtenção do Tamanho Funcional, várias técnicas têm sido propostas. Nesta seção abordaremos algumas delas.

Contagem Estimada

Desenvolvido pela *Nederlandse Software Metrieken Associatie* - NESMA [NES 98] como uma técnica para estimativa de tamanho em estágios iniciais do ciclo de vida do sistema, consiste na classificação de toda função de dados como de Grau de Complexidade Baixa e de toda função de transação como de Complexidade Média. A única diferença em relação à contagem original é que a complexidade não é estimada individualmente e sim por *default*.

Contagem Indicativa

Também desenvolvida pela NESMA para aplicação em estágios iniciais, consiste na utilização apenas do número de funções de dados (ILFs e EIFs) para cálculo do número de Pontos de Função Não Ajustado, através da seguinte fórmula:

$$UFP = 35 * \sum ILFs + 15 * \sum EIFs.$$

Pesquisas realizadas pela NESMA detectaram uma baixa dispersão entre as duas contagens propostas e a contagem original (detalhada), sendo a Contagem Estimada a que apresenta melhores resultados.

Complexidade Média

O *International Software Benchmarking Standards Group* - ISBSG, uma organização sem fins lucrativos formada pela cooperação de diversas associações nacionais de métricas de software, mantém um repositório de dados relativo a milhares de projetos de software, através do qual realiza periodicamente diversas análises publicadas em relatórios chamados de "The Benchmark". Dentre outras informações, a versão 5, publicada em 1998 [ISB 98], apresenta um número médio de UFPs para cada elemento funcional, cujo resultado é o seguinte: ILF (7,4), EIF (5,5), EI (4,3), EO (5,4) e EQ (3,8). Com base nestas informações é possível calcular o número de Pontos de Função Não Ajustado total através da fórmula:

$$UFP = 7{,}4*\sum ILFs + 5{,}5*\sum EIFs + 4{,}3*\sum EIs + 5{,}4*\sum EOs + 3{,}8*\sum EQs.$$

Backfiring

Backfiring é o processo de derivar o número de Pontos de Função a partir do número de linhas de código, através de um fator de conversão específico para a linguagem em questão. A técnica, apesar de ser aparentemente atrativa pela rapidez e facilidade de utilização, apresenta alguns inconvenientes. Além das desvantagens inerentes ao próprio método SLOC (elencados na Seção 2.1), podemos citar a considerável margem de erro existente no processo, uma vez que a técnica assume uma relação linear entre Tamanho Funcional e tamanho físico. Outra desvantagem é a variação encontrada nas mais diversas fontes em relação aos valores dos fatores de conversão, conforme explicitado em [VAZ 04]. Por exemplo, para a linguagem COBOL, temos os seguintes valores de LOC/FP: 77 (fonte: *Quantitative Software Management*), 105 (Caper Jones), 177 (David Consulting Group) e 91 (Donald Reifer). Por último, vale mencionar que o surgimento de novas (e complexas) linguagens têm tornado difícil o processo de obtenção de valores de conversão consistentes.

Portanto, o uso desta técnica é arriscado e sujeito a uma grande margem de erro, sendo indicado apenas para uso em situações onde a contagem pelo método detalhado seja inviável, como no caso de sistemas legados [VAZ 04], ou em modelos de estimativa que utilizam, em suas fórmulas, o tamanho em linhas de código, como ocorre com o COCOMO (descrito na Seção 2.3.1).

3.2 FPA na Extração de Requisitos

Um dos grandes problemas que desenvolvedores de software têm encontrado em seu

trabalho é a habilidade de julgar se os requisitos estão suficientemente completos e corretos antes de iniciar formalmente o desenho e a codificação.

O desafio das equipes de desenvolvimento tem sido capturar o máximo de requisitos conhecidos nos estágios iniciais do projeto e gerenciar o impacto dos requisitos desconhecidos (*Unknow Unknows*), quando estes surgem [DEK 99].

A Análise por Pontos de Função pode ser utilizada como uma ferramenta adicional para identificar requisitos omissos, assegurar a totalidade dos mesmos e revelar potenciais defeitos. De fato, os benefícios obtidos pela aplicação de FPA podem ser mais valiosos do que meramente a obtenção do Tamanho Funcional do software.

3.2.1 Verificação de Requisitos

Os três principais tipos de requisitos de software, conforme descrito por [DEK 01], são:

1. Requisitos Funcionais: abrange lógica de negócio ou funções do usuário; inclui processos elementares que devem ser suportados para permitir a entrada, processamento, manipulação, saída e interface de dados para, de e dentro das fronteiras da aplicação;

2. Requisitos Não Funcionais: abrange restrições de usuário/negócio independentes de tecnologia; inclui requisitos de qualidade e performance, tais como portabilidade, usabilidade, segurança, dependências, confiabilidade e velocidade;

3. Requisitos Técnicos: abrange configurações específicas de hardware/software; FPA não aborda este tipo de requisito.

FPA fornece uma técnica adicional para checagem da totalidade dos Requisitos Funcionais, pois fornece uma visão única da perspectiva do usuário segundo o seu entendimento, em oposto à implementação física destas mesmas funções medidas por SLOC.

Quando se realiza uma contagem de FP's, todos os Requisitos Funcionais conhecidos são analisados, ponderados e contados, utilizando-se o método de contagem padrão. É durante a análise que a maioria dos erros e omissões são descobertos, conforme os passos a seguir [DEK 01]:

1. Determinação do propósito e escopo da contagem
   A análise decorrente da determinação do propósito da contagem e do seu escopo pode contribuir para um melhor esclarecimento das premissas do projeto, bem

como da identificação dos eventuais erros nos documentos de requisito;

2. Identificação das fronteiras da aplicação

Um diagrama de contexto do projeto, por ser visual e independente de tecnologia, permite a descoberta de interfaces que possam não constar dos requisitos escritos, mesmo que tenham sido discutidas nas fases iniciais do projeto; além disso, este passo, junto com outros subseqüentes, permite uma identificação clara das funções contidas dentro de cada aplicação;

3. Contagem das funções de dados

O registro dos tipos de elementos de dados conhecidos e ainda não identificados nos requisitos podem fornecer, em conjunto com as funções de transação, uma listagem das entidades de dados que deverá ser avaliada quanto à consistência e totalidade;

4. Contagem das funções de transação

Algumas regras de validação, como as que são apresentadas a seguir, podem auxiliar na identificação de omissões e inconsistências:

- √ se uma entidade lógica persistente foi identificada como um Arquivo Lógico Interno, ou seja, mantida através de uma função padrão de manutenção da aplicação, mas não existe nenhuma Entrada Externa associada, então existe um ou mais descasamentos nos requisitos, ou seja, a entidade é realmente um Arquivo de Interface Externa, ou existe no mínimo uma omissão nos requisitos para manter a entidade, tal como uma função de inclusão, alteração ou remoção.
- √ se existem funções de administração de dados mas não existe nenhuma entidade lógica persistente para armazená-los (ILF), então o modelo de dados pode estar incompleto; isto indicaria a necessidade de revisar os requisitos de dados da aplicação.
- √ se existe uma função de atualização de dados presente para uma entidade identificada como apenas de referência (EIF), então isto pode indicar que a entidade é na verdade um ILF; neste caso, os requisitos de dados estão inconsistentes e necessitam ser revisados.
- √ se existem funções de consulta ou saída que especificam campos de dados a serem apresentados, mas que não possuem uma origem definida (ILF/EIF ou *harded-coded*), então há um descasamento entre as funções de dados e transação. Isto indica que os requisitos de dados necessitam ser revisados.

Da mesma forma, FPA fornece algum suporte para verificação dos Requisitos Não Funcionais. Tais requisitos, apesar de impactantes no projeto como um todo, em geral somente são avaliados em estágios posteriores do ciclo de vida de desenvolvimento. Através de FPA, é possível se obter, pela análise das Características Gerais do Sistema (que incluem performance e volume de transações, dentre outras), informações valiosas, tais como a necessidade de uma disponibilidade total do sistema (24 horas por dia, 7 dias na semana) devido à carga de transações ou a proteção especial contra perda de dados ainda em estágios iniciais do projeto.

A Tabela 3-7, extraída de [DEK 01], traz um exemplo de como FPA pode ser utilizada para descobrir defeitos nos requisitos. Por exemplo, uma mesma função de dado "Empregado" referenciada simultaneamente como ILF e EIF[1], ou ainda sem nenhuma função de transação para manipulação do mesmo, são indicativos de alguma inconsistência nos requisitos funcionais.

Tabela 3-7 Exemplo de uso de FPA para validação dos requisitos

Funções de Dados	ILF	EIF	Funções de Transação	EI	EO	EQ	Indicador de Problema	Problema Potencial
Entidade "Empregado" (mantido)	x	x	-				sim	1. Mesma entidade mantida e referenciada 2. Nenhuma função de manutenção
Vendas Mensais	x		Incluir Vendas Remover Vendas	x x			sim	1. Função de atualização ? 2. Não há relatórios (consultas) ?
-			Relatório Contábil			x	sim	Nenhuma fonte de dados identificada
% dos componentes FPA	50	0		10		7	possível	1. Nenhum arquivo externo ? 2. Nenhuma consulta ? 3. Todas as funções transacionais estão identificadas ?

Ter um conjunto documentado de requisitos funcionais (e não funcionais) do usuário tais quais endereçados pelo FPA vai muito além da fase de requisitos. Este simples conjunto de funções documentadas minimiza questões como "quem disse o que e quando" e reduz a discussão se as funções foram ou não incluídas na especificação submetida. Além disso, quando mudanças de escopo emergem posteriormente no projeto, como inevitavelmente ocorre, clientes e desenvolvedores se encontram em condições de ajustar o tamanho do FPA e rapidamente

[1] Descartando-se a possibilidade de erros no processo de contagem.

avaliar o impacto da mudança no projeto.

Portanto enquanto outras técnicas de revisão e rastreamento podem também se mostrar de valor, FPA é um método simples que pode, além de calcular o Tamanho Funcional do software e permitir a obtenção de estimativas de esforço, apoiar o processo de extração de requisitos [DEK 01].

3.2.2 Casos de Uso

A abordagem empregada em Casos de Uso envolve a captura dos requisitos sob a perspectiva de como o usuário (chamado neste contexto de *Ator*) irá realmente interagir com o sistema, ao invés da perspectiva de quais características o sistema deverá comportar. Cada Caso de Uso descreve uma ação que é necessária para que um ator realize uma determinada função do sistema. É utilizada no início do ciclo de vida para validação do desenho proposto e garantia de conformidade com os requisitos, independentemente da tecnologia empregada na solução.

Este conceito implica que os diagramas de Caso de Uso podem também ser utilizados para a contagem de funções, permitindo desta forma melhores estimativas de prazos e custos[LON 04a].

3.2.3 Mudanças de Escopo

FPA pode facilitar o processo de tomada de decisões frente a mudanças de escopo. Sempre que mudanças são introduzidas no projeto, ou novos requisitos são identificados, esboços das novas Especificações de Requisitos (SRS) e Casos de Uso podem ser gerados, de forma que FP's podem ser recalculados e introduzidos em modelos de estimativas de esforço a fim de quantificar o impacto das mudanças em termos de recursos, prazo e custo [DEK 99].

Se as mudanças são subseqüentemente aceitas no projeto, SRS e Casos de Uso são formalmente revisados, e novos Casos de Teste são escritos de acordo com as funcionalidades introduzidas, conforme o processo descrito na Figura 3-2, extraído de [DEK 99].

O gerenciamento de mudanças de escopo, conforme descrito, mostra-se similar ao gerenciamento de uma construção - mudanças de pedidos são avaliadas em termos de impacto no projeto e então uma decisão é tomada de acordo com as estimativas geradas. Desta forma, as mudanças podem ser introduzidas de um modo planejado, estimado e controlado, ao invés de simplesmente executadas.

3.3 Estimativas de Esforço com FPA

Conforme exposto na Seção 2.3.1, existem várias abordagens para obtenção de estimativas de esforço. Nesta seção abordaremos alguns fatores importantes ligados ao uso de FPA.

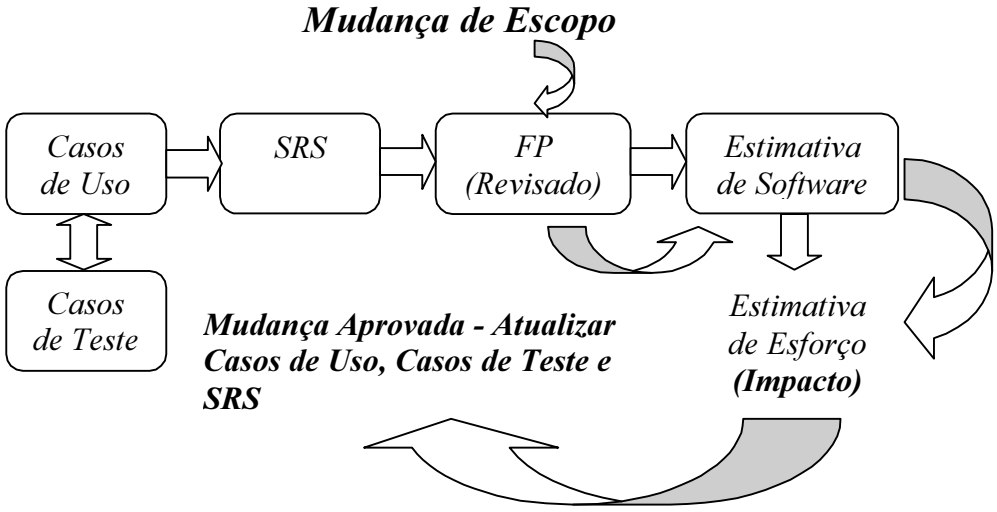

Figura 3-2 Mudança de Escopo com FPA

3.3.1 Uso do Fator de Ajuste

O Fator de Ajuste de Valor - VAF (descrito na Seção 3.1.6), introduzido originalmente por Albrecht de forma subjetiva e desde a revisão do método em 1984 calculado através das 14 Características Gerais do Sistema - GSCs, tem recebido muitas críticas e vem sofrendo um processo de desuso, basicamente por dois motivos: (a) a defasagem das GSCs em relação ao cenário atual; e (b) a variedade de interpretação na definição do nível de influência de cada GSC. Segundo [VAZ 04], algumas das deficiências são as seguintes:

1. algumas GSCs são inter-relacionadas ou de natureza similar, como por exemplo Performance e Volume de Transações;

2. existem requisitos importantes não contemplados;

3. não é apropriado que todas as GSCs possuam o mesmo peso, nem que o limite máximo de influência seja de 5%;

4. a variação de ± 35% é insuficiente para representar as funcionalidades gerais da aplicação.

Em função dos problemas apresentados foi criado um grupo de trabalho do comitê de práticas de contagem do IFPUG dedicado ao assunto. Ao final de 2002 o uso do VAF tornou-se opcional para que pudesse ser considerado apto a tornar-se um Padrão Internacional pela ISO e, desde então, praticamente não se utiliza o valor de Pontos de Função Ajustado para efeito de

obtenção de estimativas ou outras análises. O modelo COCOMO II, para citar um exemplo, utiliza apenas o valor não ajustado em seus cálculos, sendo o fator de ajuste ignorado.

Atualmente, há uma tendência do mercado de se contabilizar os fatores técnicos e de ambiente durante o cálculo de estimativa, através da utilização da Taxa de Produtividade (ou de Entrega), conforme exposto na Seção 2.3.1, aplicado sobre o total de Pontos de Função Não Ajustado. Considerando-se, por exemplo, um Tamanho Funcional de 200 Pontos de Função Não Ajustado (DFP) e uma Taxa de Produtividade média de 0,25 UFP/HH, tem-se:

Esforço = 200 (UFP) / 0,25 (UFP/HH) = 800 HH.

No entanto, pode ser que seja mais adequado definir um valor de 1/6 UFP/HH em função do grau de performance exigido, obtendo-se assim um valor de esforço ajustado de 1.200 HH.

Transferindo-se do Tamanho Funcional para a Produtividade os ajustes necessários em função de fatores técnicos/ambientais, consegue-se manter o valor do tamanho do sistema apenas como uma função exclusiva de suas funções, sem deixar de considerar os impactos dos ajustes no cálculo final de esforço.

3.3.2 Deficiências do Modelo

Duas propriedades importantes do modelo proposto para o método de FPA, expostos a seguir, devem ser levadas em consideração ao se realizar estimativas de esforço.

A primeira refere-se ao Axioma da Ascendência, que diz: se uma aplicação a é uma instância da aplicação a' ($a \subseteq a'$), então o Tamanho Funcional de a' deve ser no mínimo igual à de a, ou

$$a \subseteq a' \Rightarrow \text{UFP}(a) \leq \text{UFP}(a').$$

Contudo, FPA viola este axioma. Vejamos um exemplo, extraído de [FET 99]: uma aplicação para processamento de pedidos mantém três grupos de dados (clientes, produto e pedidos). Uma consulta possui os seguintes campos: nome do cliente, data de início e término do pedido, status do pedido e categoria de produto. Um relatório é obtido a partir dos três grupos de dados e os campos apresentados são: nome do cliente, id do produto, descrição do produto e quantidade. Baseado nestas informações e de acordo com as regras de contagem FPA, a transação (t_a) se refere a uma Consulta Externa com um total de 6 Pontos de Função Não Ajustado. Contudo, se um campo de valor total do pedido é acrescentado ao relatório ($t_{a'}$), então a transação passa a ser uma Saída Externa, com uma contagem total de 4 Pontos de Função Não Ajustado. Ou seja, apesar de $t_{a'} > t_a$, tem-se que UFP (t_a) > UFP ($t_{a'}$). Portanto, um acréscimo de funcionalidade, que certamente implicará em um maior esforço, pode não ser capturado pelo

valor do Tamanho Funcional (ao contrário, este pode indicar um esforço requerido menor). Esta deficiência (superada em outros métodos, como COSMIC FFP), deve ser levada em consideração ao se analisar projetos de melhoria.

A segunda se refere ao Axioma da "Monotonicidade" (*Monotonicity*), que diz: para três aplicações ***a***, ***a′*** e ***a″***, tem-se que:

$$\text{UFP}(\boldsymbol{a}) \leq \text{UFP}(\boldsymbol{a'}) \Leftrightarrow \text{UFP}(\boldsymbol{a} \cup \boldsymbol{a''}) \leq \text{UFP}(\boldsymbol{a'} \cup \boldsymbol{a''}).$$

Ou seja, se uma aplicação ***a*** é menor em Tamanho Funcional que ***a′***, então uma extensão de ambas as aplicações com a mesma funcionalidade ***a″*** não pode inverter esta relação. Contudo, também neste caso observa-se que FPA viola o axioma. Suponha, através de outro exemplo extraído de [FET 99], que uma aplicação ***a*** referencia um grupo de dados de empregados, com um total de 19 elementos de dados e 3 subgrupos. Neste caso, trata-se de um Arquivo de Interface Externa com uma contagem de 5 de Pontos de Função Não Ajustado. Suponha também uma aplicação ***a′*** operando sobre uma base de dados similar, com um total de 17 elementos de dados e 2 subgrupos, porém com uma função de atualização. Neste caso, trata-se de um Arquivo Lógico Interno com uma contagem de 7 Pontos de Função Não Ajustado. Tem-se portanto que UFP (***a***) < UFP (***a′***). Agora suponha que ambas as aplicações sejam estendidas por uma funcionalidade ***a″***, que consiste de um grupo adicional de dados e um conjunto de funções de transação para administração da base de dados. Agora, ambas as aplicações são Arquivos Lógicos Internos, porém ***a*** U ***a″*** passam a contabilizar 10 Pontos de Função Não Ajustado, o mesmo valor de ***a′*** U ***a″***. Ou seja, UFP (***a*** U ***a″***) = UFP (***a′*** U ***a″***). Portanto, conclui-se que, se uma aplicação for desenvolvida em partes, não se pode prever o Tamanho Funcional (e o esforço) da aplicação inteira baseado nestas partes. Da mesma forma que no caso anterior, esta deficiência (que também foi superada em outros métodos) deve ser levada em consideração ao se realizar estimativas de esforço [FET 99].

3.3.3 Indicadores de Mercado / Benchmarking

Diversos indicadores de mercado estão disponíveis para utilização por parte de gerentes de sistemas, a fim de facilitar análises de benchmarking para fins de otimização de performance, prover uma base de dados para obtenção de estimativas, sobretudo para empresas que não possuem um histórico próprio que possa ser utilizado como referência, ou ainda dar subsídios para processos de tomada de decisão, sobretudo no que diz respeito à terceirização (*outsourcing*) de serviços e acompanhamento daqueles já terceirizados. Estes indicadores fornecem parâmetros como Tamanho Funcional médio (em Pontos de Função) ou Taxa de Produtividade (em Pontos de Função por Homem-Hora) dos mais diversos tipos de projetos

(desenvolvimento, manutenção evolutiva, otimização de performance, etc.) e segmentos de softwares (escritório, gestão, indústria, etc.).

Historicamente, alguns pesquisadores como Capers Jones e Howard Rubin vêm se dedicando ao assunto e já disponibilizam os resultados de seus trabalhos através de publicações científicas ou em forma de serviços. Entretanto, à medida que a demanda da indústria por dados mais representativos, consistentes e padronizados foram crescendo, diversas outras fontes têm surgido, sobretudo através de organizações como ISBSG, The Gartner Group e META Group, apenas para citar algumas.

Futuramente espera-se uma maior padronização e agilidade no processo de coleta de dados, de forma a permitir que, a partir de repositório comum, os dados possam ser disponibilizados ao público pela Internet, através de ferramentas automatizadas que farão a captura e análise estatística das informações [GAR 01].

• • •

Neste capítulo vimos em detalhes os passos para contagem de FPA, além de algumas simplificações. Vimos também como o processo de extração de requisitos pode se beneficiar com a utilização de FPA e ainda algumas questões relacionadas à estimativa de esforço. Por último apresentamos alguns métodos alternativos que podem ser empregados em aplicações Web.

Antes, porém, de apresentarmos uma demonstração prática neste sentido, devemos discutir com mais propriedade o que é exatamente uma aplicação Web, bem como suas características e particularidades, que deverão ser levadas em consideração durante um processo de contagem, para que possamos avaliar corretamente a aplicabilidade dos métodos expostos anteriormente neste ambiente. Este estudo é o assunto do próximo capítulo.

Capítulo 4

Pontos de Função em Aplicações Web

Uma aplicação baseada em Web é definida como uma coleção de funções interativas, orientadas por página, construída em interface HTML sobre uma arquitetura em camadas, e que atende à comunidade de usuários Internet. Por suas particularidades, requerem novas abordagens de projeto e desenvolvimento, mas apresentam os mesmos desafios que sistemas tradicionais. Portanto, as mesmas técnicas de engenharia de software ainda são necessárias, mas o processo de obtenção do Tamanho Funcional e estimativa de esforço deve considerar estas diferenças, como veremos a seguir.

4.1 Aplicações Web

O uso comercial da Internet tem crescido exponencialmente nos últimos cinco anos. Neste tempo a Internet evoluiu de um simples meio de comunicação (correio, arquivos, *newsgroup*, salas de bate-papo, etc.) para um veículo de distribuição de informação e mais recentemente para um canal de comércio eletrônico (*e-commerce*). Web sites, que antes simplesmente distribuíam informações para visitantes, tornaram-se sistemas interativos e altamente funcionais que permitem a interação de muitos tipos de usuários com muitos tipos de negócio.

4.1.1 Arquitetura

A arquitetura de um sistema descreve como o mesmo é construído em termos de seus componentes, como estes são conectados e integrados entre si.

Apenas alguns anos atrás Web sites eram basicamente compostos de arquivos HTML estáticos, usualmente criados por um único *Webmaster*, que utilizava HTML, JavaScript e scripts CGI para apresentar informações e obter dados dos visitantes através de formulários. Este modelo de operação podia suportar sites relativamente simples, que utilizavam software não escalável, pouca segurança e funções limitadas. Eram chamados sistemas de duas camadas, devido a sua configuração cliente-servidor.

Web sites são atualmente sistemas altamente funcionais, que fornecem serviços Business-to-Commerce (B2C), Business-to-Business (B2B) e muitos outros. Ao invés de visitantes, agora se utiliza o termo usuário, implicando muito mais interação. No lugar de *Webmasters*, grandes sites empregam gerentes que coordenam uma equipe de diversos tipos de profissionais, que inclui programadores, administradores de banco de dados,

administradores de rede, engenheiros de usabilidade, designers gráficos, especialistas em segurança, etc. Esta equipe utiliza diversas tecnologias, incluindo uma variedade de Java (Java, servlets, applets, Java Beans, EJB), HTML, JavaScript, XML, UML, e várias outras. O crescimento do uso de componentes de terceiros representou uma das grandes mudanças. Ao invés do modelo simples cliente-servidor, a configuração se expandiu inicialmente para um modelo três camadas (Figura 4-1), e agora mais genericamente N-camadas (*N-tier*).

A fim de aumentar os atributos de qualidade, como segurança, confiabilidade e disponibilidade, a lógica de negócio tem sido transferida para um computador separado - os servidores de aplicação. Tipicamente operando em paralelo, estes servidores interagem com um ou mais servidores de banco de dados através de *Middlewares*, *i.e.*, softwares que tratam da comunicação, conversão de dados e distribuição de processos. Novas linguagens, como Java, a habilidade de separar a camada de apresentação (tipicamente provida por um servidor Web) da camada da lógica de negócio, e novos modelos de design como JSPs e EJBs tornaram mais fácil a manutenção corretiva e evolutiva. O modelo N-camadas também permite camadas de segurança adicional. A computação distribuída, particularmente para servidores de aplicação, permite a aplicações Web tolerarem falhas e suportarem mais usuários. Mecanismos de *dispatcher*, por exemplo, redirecionam as requisições provenientes de servidores Web para componentes de hardware/software apropriados na camada de aplicação, contribuindo assim para uma melhor distribuição da carga de trabalho. Tais estratégias de projeto, aliadas a utilização de melhores práticas de programação, têm contribuído para o desenvolvimento de softwares Web mais confiáveis e escaláveis.

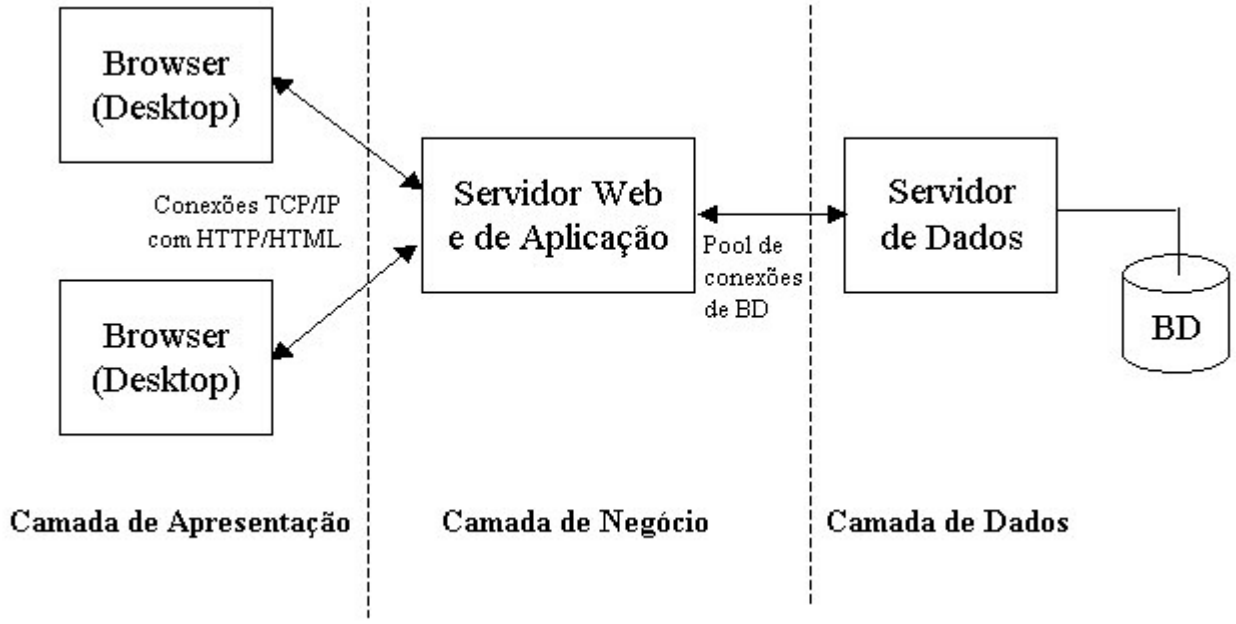

Figura 4-1 Arquitetura em 3 Camadas

4.1.2 Particularidades

Algumas características fazem da Web uma plataforma bem particular. Em primeiro lugar, aplicações Web possuem uma forte predominância de aspectos visuais, sendo em muitos casos um dos componentes mais importantes do sistema, principalmente se o objetivo estiver relacionado às estratégias de marketing ou institucionais. Este fato explica em parte porque os requisitos tornam-se muito variáveis neste ambiente, uma vez que mudanças nas interfaces do usuário são freqüentes; tais mudanças geralmente requerem algum esforço, cuja intensidade dependerá da natureza da mudança e da forma como o sistema foi implementado.

A arquitetura Web possui ainda uma natureza escalar, onde o emprego de camadas com baixo acoplamento tornam o escopo limitado e específico. Também possui um alto grau de componentização, incluindo componentes distribuídos e de fontes desconhecidas.

Um Web site pode fornecer acesso a múltiplas aplicações Web. Os tipos de função de negócio que são fornecidos podem variar consideravelmente, desde uma coleção de páginas HTML estáticas até aplicações online complexas. Soluções cada vez mais elaboradas estão sendo disponibilizadas, facilitadas pelo emprego de plataformas de desenvolvimento sofisticadas (J2EE, .NET, etc.), servidores de aplicação, e ainda pela disseminação de melhores práticas de desenvolvimento. Além disso existem muitas tecnologias distintas (HTML, XML, linguagem Java, etc.) e, portanto, muitos tipos diferentes de recursos envolvidos (editores, desenvolvedores, desenhistas gráficos, etc.) que devem ser gerenciados.

No mundo Internet a escala de tempo para entrega de novos sistemas foi encurtada de alguns anos para alguns meses, o que tem levado a uma necessidade de integração muito maior e mais ágil entre profissionais das áreas de marketing/publicidade, negócios e sistemas, e ao surgimento de novas práticas como o *Extreme Programming* - XP [BEC 99].

Softwares baseados em Web levantam novas questões econômicas. Empresas que operam através da Web dependem dos clientes usando, e mais importante, retornando aos seus sites. E diferentemente de vendedores de software, se uma nova empresa lança um site competitivo de maior qualidade, os clientes irão quase que imediatamente migrar seus negócios para o novo site. Portanto, a qualidade do produto tem sido um aspecto importante neste ambiente.

Aplicações Web possuem ainda outras características específicas do ponto de vista do produto [OFF 02], tais como:

Confiabilidade

A base de usuários Web espera que as aplicações funcionem de forma tão confiável

quanto uma loja tradicional. Mais ainda, caso isto não aconteça, os usuários não tem que se dirigir à outra loja, basta apontar seus *browsers* para outro endereço virtual.

Usabilidade

Usuários Web esperam encontrar transações fáceis, tão simples quanto comprar um produto numa loja tradicional. Além disso, a comunidade de usuários é extremamente diversificada, com diferentes graus de conhecimento. Estes fatos, aliado ao comportamento pouco leal exibido pelos usuários, demonstram a importância da usabilidade neste tipo de ambiente.

Segurança

Web sites têm sido violados e as informações privadas dos usuários distribuídas, contribuindo para a sensação de insegurança que ainda permeia o mundo Web. O desenvolvimento de aplicações Web atualmente enfrenta uma grande escassez, tanto em conhecimento disponível quanto de pessoal especializado. Aplicações devem, portanto, manusear dados e outras informações eletrônicas dos usuários da forma mais segura possível.

Disponibilidade

Web sites tem abrangência global e, portanto, acessos realizados em uma região do planeta durante o dia podem ocorrer de forma simultânea a outros durante a noite em outra. Feriados nacionais ou religiosos ocorrem em diferentes datas em diferentes países. Na Web, usuários esperam uma disponibilidade total (24 horas por dia, 7 dias por semana e 365 dias por ano). Inclui também acessabilidade via diferentes *browsers*.

Escalabilidade

Virtualmente um número ilimitado de clientes pode visitar um Web site. Aplicações Web devem estar preparadas para um rápido crescimento, em termos tanto de usuários requisitando serviços quanto do número de serviços sendo oferecidos. É interessante observar que, à medida que os sites crescem, fraquezas que antes haviam passado despercebidas podem efetivamente levar a falhas em disponibilidade, usabilidade e segurança.

Manutenibilidade

Novas versões de softwares tradicionais envolvem marketing, venda, entrega ou mesmo instalação local, o que acarreta em alto custo. Por isso, fabricantes de software normalmente coletam manutenções corretivas ao longo do tempo e as distribuem de uma só vez. Aplicações Web, no entanto, fornecem aos usuários acesso imediato às atualizações, reduzindo o ciclo de manutenção para dias ou mesmo horas. Outra ramificação do aumento

da taxa de atualização está relacionada à compatibilidade. Como usuários nem sempre atualizam seus softwares, os fabricantes devem assegurar a compatibilidade entre versões, o que naturalmente não é necessário em aplicações Web (exceto para versões de *browsers* em determinados casos).

Na Tabela 4-1, extraída de [REI 02a], são apresentadas outras características de desenvolvimento de projetos Web comparadas a projetos tradicionais.

Tabela 4-1 Características de Projetos de Desenvolvimento Tradicionais vs Web

Características	Desenvolvimento Tradicional	Desenvolvimento Web
Objetivo primário	menor custo	menor prazo
Tamanho típico (equipe)	médio à grande (centenas)	pequeno (3-5, máx. 30)
Prazo típico	12-18 meses	3-6 meses
Custo típico	$ milhões	$ milhares
Abordagens de desenvolvimento	clássica, baseada em requisitos, por fases ou incremental, casos de uso, direcionadas por documentação	ágil, XP, baseado em blocos, direcionado por demos e protótipos, RUP, múltiplas linguagens
Técnicas de engenharia	OO, C++, ferramentas CASE	baseado em componentes, linguagens de 4ª / 5ª geração (html, java, etc)
Processos empregados	CMM	*Ad hoc*
Produtos desenvolvidos	baseado em código, novo, muitas interfaces externas, aplicações complexas	baseado em objetos, uso de componentes reutilizáveis, em camadas
Pessoas envolvidas	engenheiros de software	programadores, designers gráficos
Processo de estimativa	por analogia + lições aprendidas de projetos anteriores.	*Ad hoc* baseado em experiência pessoal.

4.2 FPA em Aplicações Web

Os fatores expostos anteriormente tornam a contagem de pontos de função de aplicações Web extremamente peculiar. Inicialmente deve ser observado que, devido à existência de múltiplas linguagens e tecnologias disponíveis neste ambiente, faz-se ainda mais necessário o uso de métricas de Tamanho Funcional, uma vez que a contagem de linhas de código fonte pode produzir resultados bem diferentes sobre o mesmo escopo de contagem, dificultando assim processos de tomada de decisão.

A possibilidade de componentização e o emprego de arquitetura em N-camadas neste ambiente habilitam o uso de diferentes visões do escopo de contagem, bem como uma contagem mais delimitada e próxima das funcionalidades efetivamente incluídas ou

alteradas.

O alto grau de variabilidade dos requisitos aliado ao fato de que em geral os mesmos são obtidos conforme se processa o diálogo com usuários do negócio, muitas vezes sem um processo formal de registro, tornam difíceis os processos de identificação dos grupos lógicos de dados, processos elementares e componentes transacionais.

Devido a grande variedade de categorias de aplicações Web, é vital determinar se o site contém funções de negócio, processos lógicos e algoritmos, ou simplesmente a entrega de dados armazenados, para efeito de contagem.

É importante salientar ainda que, embora quem determine quais são as funcionalidades a serem construídas seja o usuário do negócio e não o usuário final de um Web site, este tem forte influência sobre o projeto, sobretudo em função de aspectos visuais e de navegabilidade. Como conseqüência, o Tamanho Funcional de uma aplicação Web pode variar de acordo com seu público-alvo e, em alguns casos, pode se tornar um fator considerável do tamanho total. De forma similar, o marketing (publicidade, audiência, etc.) é uma grande preocupação na maioria dos projetos de desenvolvimento Web, e muitas das idéias devem ser incorporadas na aplicação, afetando seu Tamanho Funcional.

4.2.1 Diretrizes

Algumas diretrizes devem ser observadas para a correta aplicação das regras de contagem do Manual Prático de Contagem em aplicações Web [GAR 01]:

Fronteira da aplicação e escopo de contagem

- observar o proprietário do site para a determinação da fronteira;
- observar as visões dos usuários relacionadas às áreas funcionais do negócio para determinação de diferentes fronteiras;
- considerar o escopo da *home page* com suas funções do negócio dentro da mesma aplicação;
- excluir da fronteira funcionalidades fornecidas pelos *browsers* Web;
- excluir agentes de busca não desenvolvidos pelo site;
- incluir funções de recuperação e apresentação de dados (Saída Externa ou Consulta Externa), incluindo carga de listas de valores (*drop-down*);
- incluir funções de navegação para outro Web site que inclua envio de dados (Consulta Externa ou Saída Externa);
- incluir funções de segurança (*e.g.* autorização, autenticação);
- contar separadamente funções de diferentes componentes da arquitetura Web;
- considerar todos os meios de armazenamento de dados (*e.g. cookies*) como Arquivos Lógicos Internos.

Usuários e Visões

§ considerar usuários Web mais avançados que operadores tradicionais;

§ considerar eventualmente administradores de aplicações e Webmasters.

Funcionalidade de Dados

§ considerar todos os dados mantidos pela aplicação e se certificar de que os mesmos não são apenas recuperados por outra aplicação (EIF's).

Funcionalidade Transacional

§ seguir as regras de processos elementares para identificação correta das transações; não considerar a recuperação de documentos similares como transações separadas;

§ atentar para múltiplas implementações de uma única visão do usuário;

§ considerar sempre a visão lógica; por exemplo, links que fornecem apenas funções de navegação não devem ser reconhecidos como Pontos de Função;

§ certificar-se de que todas as funcionalidades estão corretamente identificadas;

§ considerar transações que permitam ao usuário salvar dados como um EI; por exemplo, dados enviados por correio eletrônico (e não mantidos dentro da aplicação) devem ser contados como uma Consulta Externa;

§ atentar para a contagem de processos elementares, mesmo que haja diversas páginas físicas para atendimento dos requisitos.

4.2.2 Aplicabilidade

A seguir examina-se a aplicabilidade de método FPA frente aos diferentes tipos de aplicações Web.

1. Páginas HTML estáticas

Web sites simples em geral são compostos de uma coleção de páginas HTML estáticas contendo tanto texto quanto elementos gráficos, conectados via *hyperlinks*. As habilidades envolvidas estão relacionadas à edição, projeto gráfico e publicidade. Pode incluir algum desenvolvimento em JavaScript, entretanto não é considerado como lógica e sim interface do usuário. O principal esforço está na criação de componentes multimídia, como imagens, áudio, vídeo e outros. As consultas suportadas por estes sites são similares às consultas de ajuda do sistema. De uma perspectiva lógica cada consulta é do mesmo tipo de uma Consulta Externa. O identificador da página é a entrada e os textos a saída. De forma similar às telas de ajuda, apenas uma Consulta Externa deve ser contada. O esforço envolvido no desenvolvimento deste tipo de aplicação está relacionado diretamente ao número de páginas criadas, ao volume de texto por página, ao número de *hyperlinks*

estabelecido, a estrutura do *menu* fornecida e, principalmente, a complexidade dos componentes *hypermedia.* Todos estes componentes não são considerados pela FPA tradicional e por isso esta técnica não é apropriada para medição deste tipo de aplicação.

2. Aplicações online

Aplicações online incluem desde aplicações desenvolvidas especificamente para Web até aquelas que atuam como *front-end* ou extensões de aplicações já existentes ("aplicações hospedadas"). Podem ser simples, cujo principal objetivo é fornecer acesso e armazenar dados através de páginas dinâmicas acessadas por link direto entre a camada de apresentação e o banco de dados, com nenhuma camada lógica envolvida, ou aplicações de negócio complexas, que fornecem funções completas envolvendo manipulação de dados, lógica e algoritmos, implementadas em linguagens como Java e executadas através de servidores de aplicação.

Em ambos os casos há a presença de funções de dados e transações, e portanto a FPA se mostra como uma técnica apropriada para medição. Arquivos mantidos via formulários, se desenvolvidos especificamente para a aplicação, ou por funções de negócio (não temporários), devem ser contados como ILF, e EIF caso se refiram a aplicações hospedadas. Arquivos referenciados, utilizados para recuperação de dados de negócio e que satisfazem as consultas disponíveis, devem ser contados como EIF. Arquivos de controle. os quais armazenam links de *hypertexto*, não são contados.

3. Outros componentes da arquitetura Web

As principais questões que devem ser endereçadas para contagem destes componentes são: quais funções representam requisitos lógicos de usuário e quais funções foram construídas pela equipe de desenvolvimento da organização ?

Por exemplo, uma função de *download* de arquivos é normalmente fornecida pelos *browsers,* e não deve ser contada. Funções de *e-commerce*, envolvendo manipulação de "carrinho de compras", ou serviços de cobrança, devem ser contados. Já *hyperlinks* podem incluir:

- § links de imagens em páginas estáticas HTML - neste caso não são contados;
- § links para agências de publicidade - neste caso é contado como uma Consulta Externa;
- § links para outros sites/aplicações Web - tratados da mesma forma que seria em outros sistemas, ou seja, simples acessos a outras aplicações; devem ser tratados como navegação e não são contados. Entretanto, se como parte da transferência, dados são enviados, então neste caso a contagem deve ser do tipo Saída Externa.

4.3 Considerações sobre Estimativas em Aplicações Web

Conforme já mencionado na Seção 2.3.1, alguns modelos de estimativa de esforço para Web têm sido propostos. Entretanto alguns fatores tornam tais modelos imprecisos, se não considerados durante a análise.

O principal deles diz respeito ao processo de criação de componentes de mídia, tais como gráficos, imagens, vídeos, áudios, animações, etc. Estes componentes, coletivamente chamados de multimídia, ou mais especificamente, *hypermedia*, requerem um esforço de desenvolvimento que em pouco se assemelha a aqueles despendidos no desenvolvimento de sistemas tradicionais. Sistemas de multimídia requerem ferramentas de criação (*Authoring*), e técnicas específicas com forte ênfase na qualidade visual. Apesar de ser possível medir a complexidade de uma página em função dos componentes nela contidos, como medir a complexidade de um gráfico ? [STE 98]. Alguns autores, como Rodriguez [ROD 02], defendem a classificação do processo de desenvolvimento de aplicações Web em dois subprocessos quase independentes: um voltado para criação da *hypermedia* (processo de *Authoring*), e outro encarregado do desenvolvimento das interfaces com bancos de dados, segurança e outros aspectos de infraestrutura.

Vale mencionar ainda o uso crescente de ferramentas que facilitam a construção, montagem e integração de sites (editores Web, publicadores, etc.), bem como o uso de componentes pré-construídos. Tais recursos provocam um considerável aumento na produtividade, e certamente devem ser considerados nas análises de estimativa de esforço neste ambiente.

4.4 Métodos de Contagem para Web

Várias métricas funcionais derivadas de FPA estão disponíveis atualmente (ver Seção 2.2). Para efeito deste trabalho, no entanto, serão apresentadas outras três métricas: duas voltadas especificamente a aplicações Web - Web Points e Web Objects, e também a métrica conhecida por Pontos de Caso de Uso, que apesar de não ser exclusiva de ambientes Web, obteve resultados satisfatórios em estudos que analisaram sua eficácia para este fim [AND 01].

4.4.1 Método de Pontos de Caso de Uso

Desenvolvido por Karner [KAR 93] da Objective Systems SF AB (atualmente Rational Software) como uma tese de mestrado em 1993 sob a supervisão de Ivar Jacobson, o método consiste não somente de uma extensão de IFPUG FPA, mas também de MkII FPA.

Os passos necessários para a geração da estimativa são descritos a seguir [BAN 01]:

1. Contagem dos Atores: o Peso total dos Atores Não Ajustado - UAW é calculado como sendo o somatório do produto de cada Tipo de Ator pelo respectivo Peso, de acordo com a classificação apresentada na Tabela 4-2.

Tabela 4-2 Pontos de Caso de Uso - Peso dos Atores

Tipo de Ator	Peso (AW)	Descrição
Ator Simples (As)	1	Outro sistema acessado através de uma API de programação
Ator Médio (Am)	2	Outro sistema interagindo através de um protocolo de comunicação, como TCP/IP ou FTP
Ator Complexo (Ac)	3	Um usuário interagindo através de uma interface gráfica (*stand-alone* ou Web)

Portanto:

$$UAW = AWs * \sum As + AWm * \sum Am + AWc * \sum Ac.$$

2. Contagem dos Casos de Uso: o Peso total dos Casos de Uso Não Ajustado - UUCW é calculado de acordo a complexidade envolvida, conforme as seguintes regras, resumidas na Portanto:

$$UUCW = UCWs * \sum UCs + UCWm * \sum UCm + UCWc * \sum UCc.$$

Tabela 4-3:

√ se o Caso de Uso contém uma simples funcionalidade, como uma interface de usuário simples, manipula apenas uma entidade de banco de dados, ou contém 4 ou menos cenários principais ou caminhos de execução, então o mesmo é classificado como "Fácil" e contado com um valor de 5;

√ se o Caso de Uso contém uma funcionalidade mais elaborada, envolve mais desenho de interface, manipula duas ou mais entidades de banco de dados, ou contém de 5 a 8 cenários principais ou caminhos de execução, então o mesmo é classificado como "Médio" e contado com um valor de 10;

√ se o Caso de Uso contém uma funcionalidade difícil, envolve uma interface de usuário complexa, manipula três ou mais entidades de banco de dados, ou contém 8 ou mais cenários principais ou caminhos de execução, então o mesmo é classificado como "Complexo" e contado com um valor de 15;

Portanto:

$$UUCW = UCWs * \sum UCs + UCWm * \sum UCm + UCWc * \sum UCc.$$

Tabela 4-3 Pontos de Caso de Uso - Peso dos Casos de Uso

Tipo de Caso de Uso	Peso (UCW)	Complexidade
Simples (UCs)	5	Fácil
Médio (UCm)	10	Média
Complexo (UCc)	15	Complexa

3. Cálculo do Peso total Não Ajustado - UUCP: realizado através do somatório entre os pesos de Atores e Casos de Uso, ou seja:

$$UUCP = UAW + UUCW.$$

4. Cálculo do Fator de Ajuste: constituído de duas partes: (a) cálculo de Fator de Complexidade Técnica - TCF, cobrindo uma série de requisitos funcionais do sistema similar ao adotado pelo FPA; (b) cálculo de Fator de Ambiente - EF, requisitos não-funcionais associados ao processo de desenvolvimento, tais como experiência da equipe, motivação e estabilidade do projeto. Estes dois tipos de fatores geram multiplicadores distintos, conforme explicação abaixo, devem ser aplicados ao UUCP calculado anteriormente:

Cálculo de TCF:

$$TCF = 0{,}6 + (0{,}01 * TFactor),$$

onde TFactor corresponde ao somatório dos níveis de influência (valores de 0 a 5, do menos para o mais influente), atribuídos a cada fator (T1 a T13) mostrados na Tabela 4-4, multiplicados pelo seu peso correspondente, conforme a fórmula:

$$TFactor = \sum_{i=1}^{13} peso_i \ (TCF)$$

Cálculo de EF:

$$EF = 1{,}4 + (-0{,}03 * EFactor),$$

onde EFactor corresponde ao somatório dos níveis de influência (valores de 0 a 5, do menos para o mais influente), atribuídos a cada fator (E1 a E8) mostrados na Tabela 4-5, multiplicados pelo seu peso correspondente, conforme a

fórmula:

$$EFactor = \sum_{i=1}^{8} peso_i \ (EF)$$

Tabela 4-4 Pontos de Caso de Uso - Fatores de Complexidade Técnica

Fatores de Complexidade Técnica	Peso	Fatores de Complexidade Técnica	Peso
T1. Sistema Distribuído	2	T8. Portabilidade	2
T2. Tempo de Resposta	2	T9. Facilidade de Mudança	1
T3. Eficiência	1	T10. Concorrência	1
T4. Processamento Complexo	1	T11. Recursos de Segurança	1
T5. Código Reusável	1	T12. Acessível por Terceiros	1
T6. Facilidade de Instalação	0,5	T13. Requer Treinamento Especial	1
T7. Facilidade de Uso	0,5		

Tabela 4-5 Pontos de Caso de Uso - Fatores de Ambiente

Fatores de Ambiente	Peso	Fatores de Ambiente	Peso
E1. Familiaridade com RUP ou outro processo formal	1,5	E5. Motivação	1
E2. Experiência com a Aplicação em desenvolvimento	0,5	E6. Requisitos estáveis	2
E3. Experiência em Orientação a Objetos	1	E7. Desenvolvedores em meio-expediente	-1
E4. Presença de analista experiente	0,5	E8. Linguagem de programação difícil	2

5. Finalmente, cálculo do Ponto de Caso de Uso Ajustado - AUCP:

$$AUCP = UUCP * TCF * EF.$$

Segundo Karner, pode-se estimar o tempo necessário para o desenvolvimento do projeto calculando-se uma média de 20 horas de trabalho por AUCP, sendo que experiências demonstram uma variação neste valor entre 15 e 30 horas.

Em um estudo realizado em 2001 na Universidade de Oslo, Anda *et al.* [AND 01], comparou o método com estimativas realizadas por especialistas e obteve um

resultado satisfatório em termos de erro relativo.

Contudo o método possui algumas desvantagens, sendo a principal delas a inexistência de padrões universais para a construção de Casos de Uso, sobretudo quanto ao nível de detalhamento empregado, ou seja, à quantidade e granularidade dos Casos de Uso [FRE 03].

4.4.2 Método Web Points

Proposto por Cleary [CLE 00] para obtenção do tamanho e esforço de Web sites estáticos, o método Web Points baseia-se no número e complexidade de páginas HTML.

Neste método a complexidade de cada página é classificada em três níveis (Baixa, Média e Alta) de acordo com dois fatores: contagem de palavras e combinação de números de *hyperlinks* internos e externos, mais o número de componentes não textuais de cada página.

Os passos para contagem são definidos como:

1. Determinação do número de páginas;

2. Determinação do Grau de Complexidade de cada página, de acordo com a Tabela 4-6;

Tabela 4-6 Web Points - Tabela de Complexidade

Complexidade de Página HTML				
Contagem de Palavras	Contagem de Links Internos, Externos e Componentes Não Textuais			
	0-5	6-15	> 15	
0-300	*Baixa*	*Baixa*	*Média*	
301-500	*Baixa*	*Média*	*Alta*	
> 500	*Média*	*Alta*	*Alta*	

Por exemplo, uma determinada página HTML que contenha 200 palavras e 30 elementos entre links e outros componentes textuais (*e.g.* imagens), deverá ser classificada com um Grau de Complexidade Médio;

3. Determinação do número de Web Points - WP, calculado através do somatório dos WPs para cada Grau de Complexidade, de acordo com os fatores de conversão apresentados na Tabela 4-7.

Tabela 4-7 Web Points - Tabela de Conversão Complexidade x WP

WP	
Complexidade	**Conversor (W)**
Baixo	13
Médio	5
Alta	11

O valor total de Web Points é então calculado como:

$$WP = W_B * \sum WP_B + W_M * \sum WP_M + W_A * \sum WP_A.$$

Como exemplo, suponha a contagem apresentada na Tabela 4-8.

Tabela 4-8 Web Points - Exemplo de Contagem

Complexidade	Núm. Páginas
Baixo	4
Médio	6
Alta	7

Temos portanto:

$$WP = 4*13 + 6*5 + 7*11 = 159.$$

Web Points podem ser utilizados em conjunto com dados de produtividade para determinação do esforço de desenvolvimento ou melhoria no que diz respeito à análise de requisito, projeto, construção, teste e implementação de páginas HTML; porém, não inclui o desenvolvimento e obtenção de conteúdos ou elementos não textuais e componentes de infraestrutura, que devem ser determinados de forma independente. Também carece de uma maior comprovação empírica.

4.4.3 Método Web Objects

O método Web Objects, proposto por Reifer [REI 02a], estende a contagem de Pontos de Função tradicional e acrescenta quatro outros grupos de elementos funcionais:

- Links (XML, HTML, etc.);
- Componentes Multimídia (vídeo, áudio, imagem, etc.);
- Scripts (animação, etc.);
- Blocos de construção (ActiveX, applets, etc.).

Tais extensões são necessárias pois o método original aborda características de aplicações mais tradicionais, como o número de entradas e saídas. Utilizando estes grupos adicionais é possível então contabilizar os diferentes elementos de uma aplicação, incluindo aqueles específicos para Web.

Os passos para contagem, cujo manual está disponível em [REI 02b], são definidos como:

1. Identificação e contagem do número de elementos funcionais: realizado de acordo com o Manual Prático de Contagem de FPA;

2. Identificação e contagem do número de operandos: a identificação dos operadores pode requerer algum grau de inferência, uma vez que estes podem não se encontrar explicitamente indicados na especificação. Realizado de acordo com as seguintes regras:

   - § Componentes Multimídia: contar cada componente separadamente independente da densidade de *pixels*;
   - § Scripts: contar cada script ou caso de uso separadamente independente do número de atores envolvidos;
   - § Blocos de construção: contar cada componente separadamente independente dos recursos consumidos;

3. Identificação e contagem do número de operações únicas executadas sobre os operadores identificados: a identificação das operações pode requerer algum grau de inferência, uma vez que estas podem não se encontrar explicitamente indicadas na especificação. Realizado de acordo com as seguintes regras:
   - § Componentes Multimídia: contar cada operação separadamente (abrir, fechar, gravar, limpar, etc.);
   - § Scripts: contar cada operação separadamente (abrir, fechar, atualizar, pesquisar, avançar, etc.);
   - § Blocos de construção: contar cada operação separadamente (alinhar, buscar, inserir, remover, excluir, etc.);

4. Aplicação do peso correspondente, conforme apresentado na Tabela 4-9.
5. Cálculo do Total de Web Objects.

Tabela 4-9 Web Objects - Tabela de Pesos

Grupo	Peso		
	Baixo	Médio	Alto
Componentes Multimídia	4 (JPEG)	5 (figura Microsoft)	7 (imagem PCX, streaming)
Scripts	2 (1 a 3 atores)	3 (4 a 6 atores)	4 (mais do que 6 atores)
Blocos de construção	3 (1 a 50)	4 (51 a 250)	6 (mais do que 250)
Links	3 (HTML)	4 ("query lines")	6 (XML)

Como exemplo, extraído de [REI 02b], suponha uma aplicação Web contendo:

- 1 Script invocando um clip de animação e um arquivo de áudio associado;
- 16 linhas de HTML;
- 13 arquivos multimídia (3 gráficos e 10 slides);
- 3 Arquivos Internos (de ajuda, do usuário e de preferências);
- 2 Arquivos de Interface Externa (link para biblioteca de componentes e para o arquivo de registro do sistema operacional);
- 6 Entradas Externas (abrir, fechar, gravar, importar, consultar e pesquisar);
- 3 Saídas Externas (imprimir, exportar e "linkar");
- 15 Blocos de construção (botões, carrinho de compra, etc.);

A Tabela 4-10 resume a contagem para este caso.

No exemplo, foram considerados fatores de complexidade para cada um dos elementos citados na descrição. Por exemplo, dos 3 Arquivos Internos, considerou-se 2 com fator de complexidade médio e 1 com fator de complexidade alto. Aplicando-se os respectivos pesos (10 e 15, para ILF) e repetindo-se o processo para cada um dos elementos, chega-se a um total de 340 Web Objects.

Tabela 4-10 Web Objects - Exemplo de Contagem

Grupos	Baixo	Médio	Alto	Observação
Pontos de Função ♦ Arquivo Lógico Interno ♦ Arquivo de Interface Externo ♦ Entrada Externa ♦ Saída Externa ♦ Consulta Externa		 2*10 2*7 3*5	 1*15 6*6	 Da especificação: 3 ILFs Da especificação: 2 IEFs Da especificação: 6 EIs Da especificação: 3 EOs Da especificação: nenhum
Componentes Multimídia	1*4 3*4	13*5	1*7	Operandos: 1 arq. áudio, 13 multimídia, 1 ajuda Operações (por inferência): abrir, fechar, gravar
Scripts	3*2	1*3		Operandos: 1 script de animação Operações (idem): abrir, avançar, fechar
Blocos de construção	3*3	10*4	5*6	Operandos: 15 componentes Operações (idem): pesquisar, editar, adicionar
Links		16*4		Da especificação: 16 linhas HTML
Total	**31**	**221**	**88**	**= 340 Web Objects**

Uma vez calculado, o número de Web Objects pode ser usado nas equações do modelo paramétrico de estimativa WEBMO (Seção 2.3.1), para cálculo de esforço e duração de projetos Web.

A definição dos elementos adicionais de contagem foi inicialmente baseada em opinião de especialistas. Posteriormente o método foi aplicado durante um ano em 64 projetos Web completos, distribuídos em cinco domínios de aplicação (comércio eletrônico, financeira, B2B, portais e utilidades), durante o qual foram realizadas coletas de dados, análises e refinamentos.

Também foram desenvolvidas de forma empírica taxas de conversão entre Web Objects e SLOC (*e.g.* 32 linhas de código Java por objeto Web), a fim de se permitir o uso de uma versão modificada do modelo COCOMO II (Seção 2.3.1) para estimativas de esforço e duração. Esta versão de modelo de estimativa de custo, chamada de WEBMO (Seção 2.3.1), foi calibrada após muitos experimentos para utilizar Web Objects ao invés de SLOC, em dois dos cinco domínios de aplicação, o que permitiu, segundo Reifer, um incremento no grau de precisão em 12% [REI 02a].

Contudo o método ainda carece de maiores estudos quanto a sua viabilidade prática e não alcançou, até o momento, uma penetração em larga escala tanto em meios acadêmicos

quanto na indústria, como já ocorre com o FPA.

...

Aplicações Web possuem, portanto, características próprias que devem ser consideradas para efeito de cálculo de Tamanho Funcional e obtenção de estimativas de esforço, conforme foi exposto nas Seções anteriores. No último capítulo apresentaremos um estudo de caso através do qual avaliaremos a aplicação da FPA e de alguns métodos de contagem específicos em uma aplicação Web típica, que trata do processo de cadastramento e autenticação de usuários. A partir deste caso poderemos comparar os resultados teóricos frente ao real e com isso obter subsídios para avaliar a aplicabilidade dos métodos em sistemas de plataforma Web.

Capítulo 5

Estudo de Caso

Neste capítulo apresentaremos um estudo de caso de utilização da Análise por Pontos de Função em um sistema desenvolvido exclusivamente para plataforma Web, utilizando-se diferentes métodos de contagem. O objetivo é obter, a partir dos resultados encontrados, subsídios para uma análise comparativa frente ao caso real, a ser apresentada no capítulo final.

5.1 Descrição do Problema

O caso a ser analisado neste trabalho refere-se a um dos sistemas mais comuns presentes em Web sites: o Sistema de Identificação de Usuários. O sistema em questão faz parte da estratégia de uma das maiores empresas do ramo editorial no Brasil de aperfeiçoamento do seu sistema atual de cadastro de usuários, que reúne em uma única base, os dados de usuários de todo o seu portfólio de sites, atualmente em número superior a cinqüenta.

A documentação apresentada compõe-se de:

1. Identificação das Funcionalidades Principais;
2. Descrição dos Casos de Uso;
3. Modelo de Dados;
4. Interfaces do Usuário.

5.1.1 Identificação das Funcionalidades Principais

O sistema consiste basicamente de um grupo de funções para cadastramento e outro para o processo de Autenticação/Autorização, detalhados a seguir:

Cadastramento

- § inserção ou alteração de cadastro de usuários, consistindo de informações pessoais de caráter geral (identidade, endereço e preferências), ou específico para cada site (pesquisas);
- § cadastramento e descadastramento de *newsletters*.

Autenticação / Autorização

- § autenticação (*Login*) e autorização para acesso em áreas restritas e de forma segura, de acordo com o perfil do usuário;

§ integração com o Sistema de Assinaturas a fim de possibilitar a validação do status de assinatura, realizado através de um arquivo de Ticket[2], mantido pelo mesmo;

§ integração com o Sistema de Pagamento Online a fim de possibilitar a validação do status de pagamento, realizado através de um arquivo de Ticket, mantido pelo mesmo;

§ integração com outros sites externos a Empresa ("Parceiros") para fins de troca de cadastros e autenticação mútua.

Além destas, o sistema comportará a seguinte função adicional:

Marketing e Promoções

§ cadastramento e descadastramento de opções para recebimento de mala-direta com conteúdo de promoções, publicidade, renovação e outras ações.

Neste estudo de caso não serão contempladas as funções internas de administração do sistema, como gerenciamento de perfis de usuário, emissão de relatórios gerenciais, etc.

A Figura 5-1 mostra a arquitetura do sistema, bem como a sua integração com outros sistemas corporativos.

5.1.2 Descrição dos Casos de Uso

O Diagrama de Casos de Uso do sistema é apresentado na Figura 5-2. Os principais Atores envolvidos são os seguintes:

§ Internauta: usuário dos sites, podendo ser: Não Cadastrado, Cadastrado, Assinante ou Comprador em Banca[3];

§ Sistema de Assinaturas: sistema responsável pela captação e gerenciamento de assinantes;

§ Sistema de Pagamento Online: sistema responsável pela cobrança e gerenciamento de compradores online;

§ Sistema de Newsletter e Marketing: sistema responsável pelo envio de newsletter e mala direta;

§ Parceiro: sites externos a Empresa, em que há algum grau de integração entre as bases de usuários para fins de autenticação mútua.

[2] Ticket: arquivo de dados contendo identificação do usuário, data de validade e a lista de recursos autorizados e respectivos status. Armazenado tanto em banco de dados quanto em formato de cookie na estação do usuário, quando possível, para fins de otimização de performance.

[3] Consumidor que adquire o produto através de compra direta em bancas de jornal, e que obtém acesso às áreas restritas através de um código impresso em cada edição.

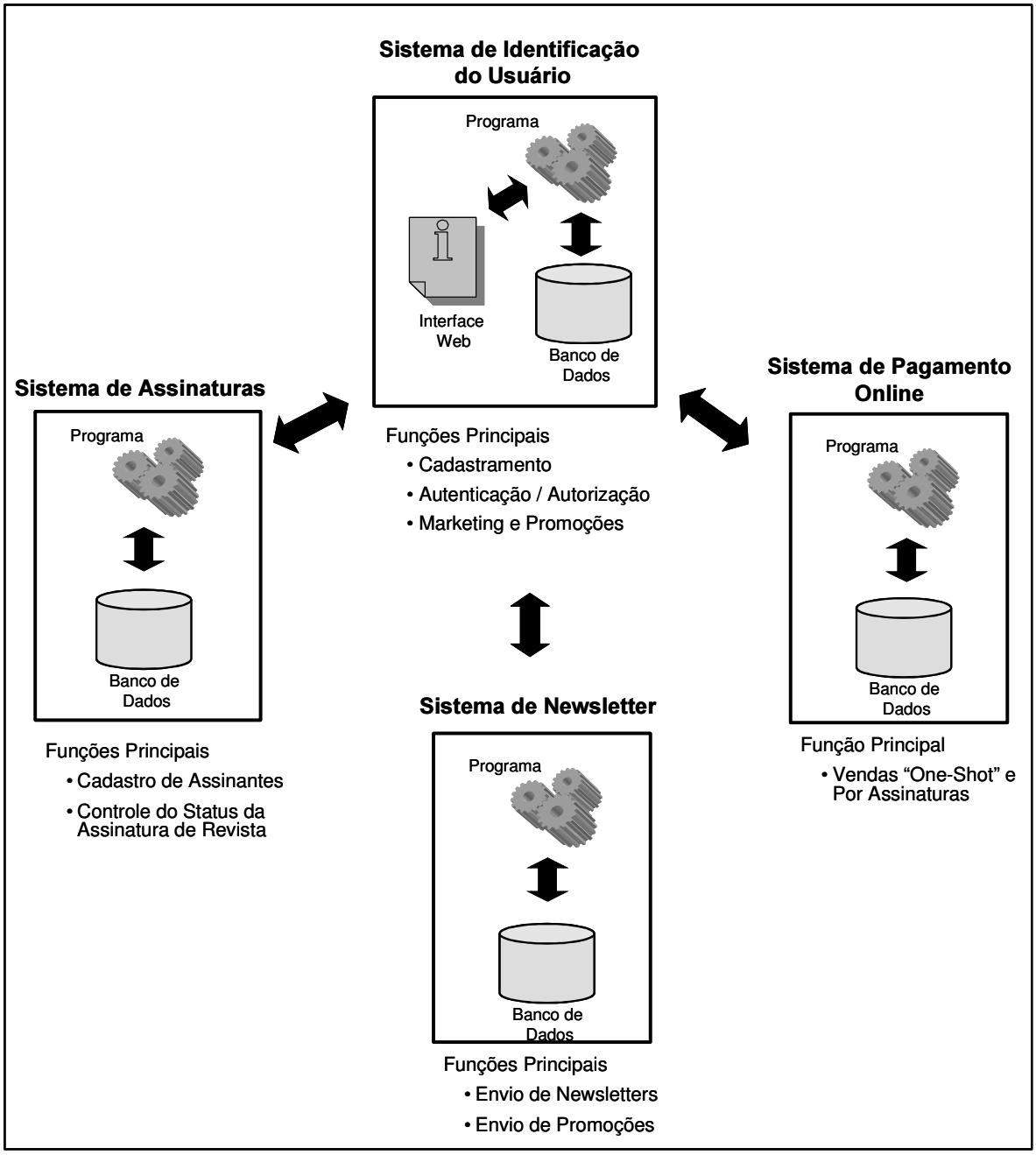

Figura 5-1 Estudo de Caso - Arquitetura do Sistema

O sistema ainda comportará um Ator com funções administrativas, que não fará parte do escopo deste estudo.

Os principais Casos de Uso e respectivas funções são as seguintes:

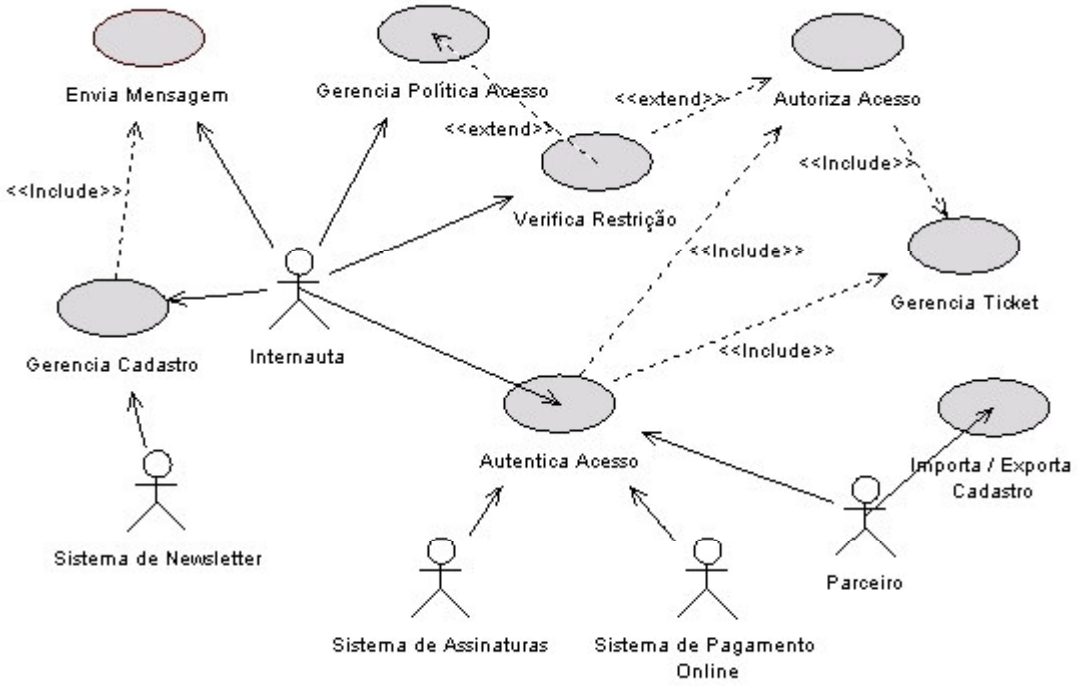

Figura 5-2 Estudo de Caso - Diagrama de Casos de Uso

- Gerencia Cadastro: inclusão, alteração e consulta de dados cadastrais do usuário, incluindo dados de *login*, indicações, assinatura de *newsletters* e ações de marketing;
- Verifica Restrição: consulta à Política de Acesso quanto ao tipo de restrição do Recurso acessado (descrito na Seção 5.1.3);
- Autentica Acesso: autenticação do *Login*, através do Ticket do usuário;
- Autoriza Acesso: geração e gravação do Ticket do usuário e redirecionamento da navegação para o Recurso autorizado;
- Gerencia Política de Acesso: inclusão, alteração, exclusão ou consulta das Políticas de Acesso de cada Recurso;
- Gerencia Ticket: inclusão, alteração, exclusão ou consulta dos dados armazenados no arquivo de *cookie* do sistema; inclui funções de criptografia e decriptografia;
- Importa / Exporta Cadastro: importação ou exportação da base de dados de usuário para sites Parceiros ou sistemas externos;
- Envia Mensagem: registro e envio de mensagens do usuário captadas na opção "Fale Conosco" da Central de Relacionamento (descrito na Seção 5.1.3).

5.1.3 Interfaces do Usuário

As interfaces do usuário são divididas em dois tipos: aquelas voltadas para o processo de autenticação/autorização e aquelas relacionadas com as funções de cadastramento e outros serviços.

O processo de autenticação dependerá do tipo de restrição exigida na área a ser acessada ("recurso") e do perfil do usuário. As possíveis restrições são:

1. recursos restritos a usuários registrados - requer apenas cadastro básico;
2. recursos restritos a compradores em banca ou assinantes - requer, além do cadastro, código de banca ou de assinatura;
3. recursos restritos a assinantes - requer, além do cadastro, código de assinatura.

O fluxo de autenticação é apresentado na Figura 5-3. As telas são apresentadas no Apêndice A

As demais funções, incluindo as de cadastramento[4], são acessadas através de uma tela inicial, chamada Central de Relacionamento, onde o usuário terá acesso ao menu com as seguintes opções:

- **Alterar Dados Cadastrais**: cadastramento de informações relativas a *login*, dados pessoais (Pessoa Física) ou da empresa (Pessoa Jurídica), código de assinante, endereço, telefone, assinatura de *newsletters*, indicação de pessoas e outras informações;
- **Alterar e-Mail**: cadastramento do e-Mail principal, utilizado no processo de autenticação;
- **Recebimento de Newsletters**: cadastramento (assinatura) de *newsletters*;
- **Ação de Marketing**: cadastramento (assinatura) de promoções ou publicidade;
- **Minhas áreas de acesso**: consulta das áreas com direitos de acesso;
- **Fale Conosco**: envio de mensagens à redação.

[4] Para as operações de alteração, requer autenticação (*Login*) prévia.

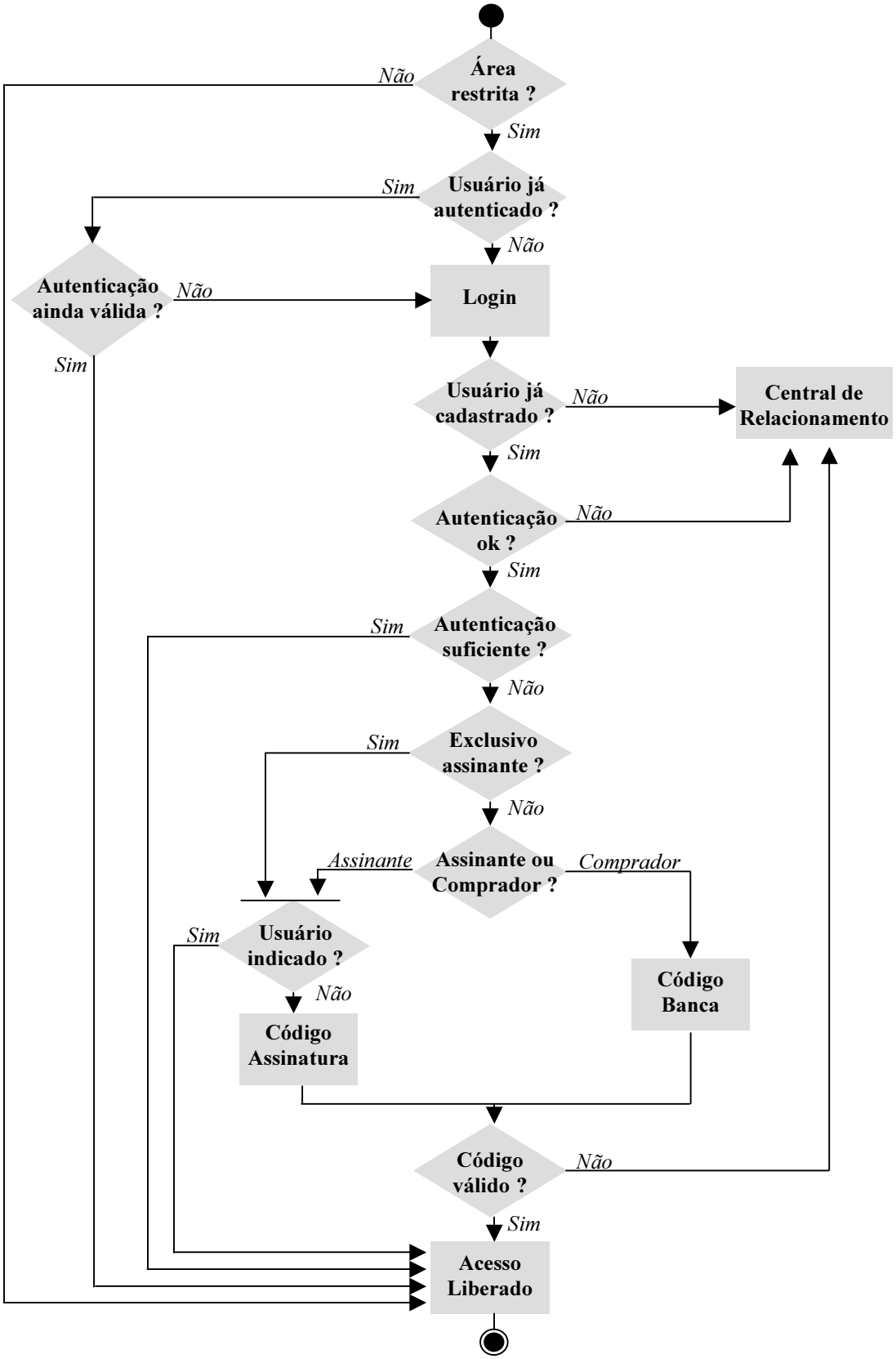

Figura 5-3 Estudo de Caso - Fluxo de Autenticação

5.1.4 Modelo de Dados

O modelo de dados, apresentado no Apêndice B, se divide em quatro grupos:

- **Cadastro**: composto de tabelas de dados necessárias para as funções de cadastramento;
- ***Newsletter* e Marketing**: composto de tabelas de dados necessárias para as funções de assinatura de *newsletter*, promoções e publicidade;
- **Autenticação**: composto de tabelas de dados necessários para as funções de autenticação;
- **Pesquisa**: composto de tabelas de dados necessárias para as funções de pesquisa de enriquecimento.

Em todos os grupos as tabelas PESSOA e SITE são comuns; tabelas externas ao sistema aparecem destacadas.

Nas funções de **cadastramento**, as tabelas utilizadas são:

Operações de Inclusão e Atualização

- PESSOA[5], PESSOA_TELEFONE, PESSOA_EMAIL, PESSOA_DOCUMENTO, PESSOA_ENDERECO: dados cadastrais comuns;
- PESSOA_FISICA,: dados cadastrais exclusivos à Pessoa Física;
- PESSOA_JURIDICA: dados cadastrais exclusivos à Pessoa Jurídica;
- PESSOA_INDICADA, PESSOA_RELACIONAMENTO: dados cadastrais de pessoas indicadas e tipo de relacionamento.

Operações de Consulta

- TIPO_ATIVIDADE, TIPO_ORGANIZACAO, TIPO_EMAIL, TIPO_DOCUMENTO, GRAU_INSTRUCAO, SITUACAO_CONJUGAL, SETOR_ATUACAO, PERGUNTA_SECRETA, SNP_PAIS_DNE, SNP_UF_DNE (externos): opções disponíveis nas listas das interfaces de cadastro (*listbox*), cadastradas previamente pelo Administrador do sistema;
- SITE (externo): sites cobertos pelo Sistema de Identificação, cadastrados previamente pelo Administrador do sistema;
- CAMPO, SITE_CAMPO (externos): dados necessários para a montagem e validação do formulário de cadastramento.

[5] Nomes próprios de colunas, arquivos ou tabelas do Modelo de Dados serão descritos em caixa alta.

Para as funções de ***newsletter*** **e marketing**, as tabelas utilizadas são:

Operações de Inclusão e Atualização

§ PESSOA_NEWSLETTER, PESSOA_OPCAO_MARKETING: assinaturas de newsletters, promoções e publicidades;

Operações de Consulta

§ TIPO_NEWSLETTER (externo): opções de newsletters disponíveis na interface de cadastro, cadastrados previamente pelo Administrador do sistema;
§ PESSOA: dados cadastrais comuns;
§ SITE (externo): sites cobertos pelo Sistema de Identificação, cadastrados previamente pelo Administrador do sistema.

Para as funções de **autenticação**, as tabelas utilizadas são:

Operações de Inclusão e Atualização

§ TICKET: dados de autorização de acesso;

Operações de Consulta

§ PARCEIRO (externo): sites externos com política de acesso integrada;
§ RECURSO (externo): unidades de acesso ao Site com uma política de acesso específica (*e.g.* página HTML, áudio, vídeo, ferramenta), cadastradas previamente pelo Administrador do sistema;
§ PESSOA: dados cadastrais comuns;
§ PESSOA_INDICADA: dados de indicação;
§ SITE (externo): sites cobertos pelo Sistema de Identificação, cadastrados previamente pelo Administrador do sistema.

Para as funções de **pesquisa**, as tabelas utilizadas são:

Operações de Inclusão e Atualização

§ PESSOA_RESPOSTA: respostas referentes a cada uma das perguntas adicionais que fazem parte da pesquisa de enriquecimento de cada site.

Operações de Consulta

§ PERGUNTA, GRUPO_PERGUNTA, SITE_GRUPO_PERGUNTA,

RESPOSTA_POSSIVEL, TIPO_CAMPO (externos): dados necessários para a montagem do formulário de pesquisa de ennriquecimento de cada site;

§ PESSOA: dados cadastrais comuns;

§ SITE (externo): sites cobertos pelo Sistema de Identificação, cadastrados previamente pelo Administrador do sistema.

5.2 Metodologia Empregada para Contagem

Para a contagem dos Pontos de Função do sistema serão utilizados, além do método de contagem IFPUG FPA, apresentado na Seção 3.1, os métodos Web Points e Pontos de Caso de Uso, apresentados na Seção 4.4. Ao final da aplicação de cada método é calculado um valor estimado de esforço baseado em valores de Taxa de Produtividade média, conforme apresentado na Seção 2.2.1. Tal metodologia para obtenção de estimativa foi escolhida por ser a mais usual utilizada no mercado e por sua simplicidade.

5.3 Contagem pelo método IFPUG FPA

Para a contagem pelo método IFPUG FPA serão utilizados os passos de contagem conforme previsto em seu Manual Prático de Contagem e apresentados na Seção 3.1.

5.3.1 Identificação do Escopo da Contagem e Fronteira da Aplicação

Apesar de se tratar de uma evolução e, portanto, de já existir atualmente, o novo sistema de cadastramento está sendo totalmente reescrito e novas funcionalidades estão sendo incorporadas, de forma que se trata de um projeto de desenvolvimento para fins de contagem. Por questões de simplicidade, não farão parte do escopo da contagem todas as funções relacionadas à administração do sistema - portanto todas as funções de dados relacionados à configuração do sistema serão contabilizadas como externas ao sistema.

O sistema possui uma forte integração com outros sistemas da Empresa, especificamente o Sistema de Assinaturas e o de Pagamentos Online, o que provocou a necessidade de adaptações nos mesmos por conta da inclusão/alteração de funcionalidades na nova versão. Apesar deste razoável grau de acoplamento, tanto a fronteira da aplicação quanto a medição do caso real serão limitados, para fins didáticos, ao Sistema de Identificação de Usuários, objeto do estudo de caso.

5.3.2 Contagem das Funções Tipo Dado

A partir dos modelos de dados apresentados no Apêndice B, foram identificadas as seguintes funções tipo dado:

- Cadastro[6] (ILF)

Descrição: grupo de dados relativo às funções de cadastramento;

DETs: COD_PESSOA, DSC_IDENT_PESSOA, DSC_SENHA_PESSOA, NOM_PESSOA, DSC_APELIDO_PESSOA, COD_PERGUNTA_SECRETA, DSC_RESPOSTA, COD_TIPO_PESSOA, COD_SITE, COD_SITUACAO_CONJUGAL, COD_TIPO_ATIVIDADE, QTD_FILHO, COD_GRAU_INSTRUCAO, IND_POSSUI_FILHO, COD_SEXO, DAT_NASCIMENTO, NUM_ANO_FUNDACAO, DSC_ENDERECO_WEB, COD_TIPO_ORGANIZACAO, COD_SETOR_ATUACAO, COD_TIPO_TELEFONE, NUM_SEQUENCIA_TELEF, NUM_TELEFONE[7], NUM_SEQUENCIA_EMAIL, COD_TIPO_EMAIL, END_EMAIL_PESSOA, IND_EMAIL_PRINCIPAL, NUM_SEQUENCIA_DOC, NUM_DOCUMENTO[8], NOM_LOGRADOURO[9], DSC_IDENT_PESSOA_INDICADA, COD_TIPO_RELACIONAMENTO, NOM_PESSOA_INDICADA, DAT_INDICACAO, COD_INFORMADO;

RETs: 2 (Pessoa Física e Pessoa Jurídica);

Grau de Complexidade: Médio (35 DETs e 2 RETs).

- PESSOA_NEWSLETTER (ILF)

Descrição: dados relativos à assinatura de *newsletters*;

DETs: COD_PESSOA, COD_NEWSLETTER;

RET: 1[10];

Grau de Complexidade: Baixo (2 DETs e 1 RET).

[6] Composto pelas tabelas PESSOA, PESSOA_TELEFONE, PESSOA_EMAIL, PESSOA_DOCUMENTO, PESSOA_ENDERECO, PESSOA_FISICA, PESSOA_JURIDICA, PESSOA_INDICADA e PESSOA_RELACIONAMENTO.

[7] Composto pelos campos NUM_DDD, NUM_TELEFONE e NUM_RAMAL.

[8] Composto pelos campos NUM_DOCUMENTO, DAT_EMISSAO e DSC_ORGAO_EXPEDIDOR.

[9] Composto pelos campos NOM_LOGRADOURO, COMPLEMENTO_LOGRADOURO, NUM_LOGRADOURO, NUM_CEP, NOM_BAIRRO, NOM_LOCALIDADE, SGL_UF, SGL_PAIS e IND_ENDERECO_DNE.

[10] Quando não informado explicitamente, refere-se ao próprio ILF ou EIF em questão.

- PESSOA_OPCAO_MARKETING (ILF)

Descrição: dados relativos à assinatura de promoções e publicidades;

DETs: COD_PESSOA, IND_PROMOCAO, IND_PROMOCAO_PARCEIROS, IND_CELULAR, IND_CELULAR_PARCEIROS;

RET: 1;

Grau de Complexidade: Baixo (5 DETs e 1 RET).

- TICKET (ILF)

Descrição: dados relativos à autorização de acesso;

DETs: COD_PESSOA, DAT_FINAL_VALIDADE, DSC_IDENT_PESSOA, DSC_SENHA_PESSOA, NOM_PESSOA, COD_SEXO, DAT_NASCIMENTO, SGL_UF, IND_CADASTRO_INCOMPLETO, DSC_LISTA_RECURSO, DSC_LISTA_STATUS_RECURSO;

RET: 1;

Grau de Complexidade: Baixo (11 DETs e 1 RET).

- PESSOA_RESPOSTA (ILF)

Descrição: dados referentes às respostas de pesquisas de enriquecimento;

DETs: COD_PESSOA, COD_PERGUNTA, DSC_RESPOSTA;

RET: 1;

Grau de Complexidade: Baixo (3 DETs e 1 RET).

- Opções do Cadastro[11] (EIF)

Descrição: opções disponíveis no formulário de cadastro, mantidas por aplicação externa (Administração) ou cadastro corporativo;

DET: DSC_SITUACAO_CONJUGAL, DSC_GRAU_INSTRUCAO, DSC_SETOR_ATUACAO, DSC_TIPO_ATIVIDADE, IND_REMUNERACAO, DSC_TIPO_TELEFONE, DSC_TIPO_EMAIL, DSC_TIPO_DOCUMENTO, DSC_TIPO_ORGANIZACAO, DSC_TIPO_RELACIONAMENTO, NOM_PAIS, NOM_UF, NOM_CAMPO, NUM_TAMANHO_CAMPO, IND_OBRIGATORIO;

[11] Composto pelas tabelas SITUACAO_CONJUGAL, GRAU_INSTRUCAO, SETOR_ATUACAO, TIPO_ATIVIDADE, TIPO_TELEFONE, TIPO_EMAIL, TIPO_DOCUMENTO, TIPO_ORGANIZACAO, TIPO_RELACIONAMENTO, SNP_PAIS_DNE, SNP_UF_DNE, CAMPO, SITE_CAMPO.

RET: 2 (Pessoa Física e Pessoa Jurídica);
Grau de Complexidade: Baixo (15 DETs e 2 RETs).

- TIPO_NEWSLETTER (EIF)

Descrição: opções disponíveis no formulário de assinatura de *newsletter*, mantidas por aplicação externa (Administração);
DET: DSC_TIPO_NEWSLETTER;
RET: 1;
Grau de Complexidade: Baixo (1 DET e 1 RET).

- SITE (EIF)

Descrição: dados relativos aos sites cobertos pelo Sistema de Identificação, mantidas por aplicação externa (Administração);
DETs: COD_SITE, DSC_SITE, DSC_URL_VENDA, IND_CPF_OBRIGATORIO, COD_SITE_PAI, IND_DOMINIO, NUM_IDADE_MINIMA;
RET: 1;
Grau de Complexidade: Baixo (7 DETs e 1 RET).

- RECURSO (EIF)

Descrição: unidades de acesso ao Site com uma política de acesso específica, mantidas por aplicação externa (Administração);
DETs: COD_RECURSO, NOM_RECURSO, DSC_RECURSO, COD_TIPO_AUTENTICACAO, COD_TIPO_VALIDACAO;
RET: 1;
Grau de Complexidade: Baixo (5 DETs e 1 RET).

- PARCEIRO (EIF)

Descrição: sites externos com política de acesso integrada, mantidos por aplicação externa (Administração);
DETs: COD_PARCEIRO, NOM_PARCEIRO, DSC_PARCERIA, NOM_CLASSE_AUTENTICACAO;
RET: 1;
Grau de Complexidade: Baixo (4 DETs e 1 RET).

- Pesquisa[12] (EIF)

Descrição: dados relacionados a montagem da pesquisa de enriquecimento;

DETs: COD_PERGUNTA, COD_GRUPO_PERGUNTA, DSC_PERGUNTA, COD_TIPO_CAMPO, DSC_GRUPO_PERGUNTA, DSC_TIPO_CAMPO, COD_RESPOSTA, DSC_RESPOSTA;

RET: 1;

Grau de Complexidade: Baixo (8 DETs e 1 RET).

5.3.3 Contagem das Funções Tipo Transação

A partir das telas apresentadas no Apêndice A e dos modelos de dados apresentados no Apêndice B, foram identificadas as seguintes funções tipo transação:

- Inclusão ou Alteração de Cadastro - P. Física (EI)

Descrição: inclusão de dados cadastrais de Pessoa Física;

DET: Endereço de e-Mail, Tipo de e-Mail, Senha, Confirmação de Senha, Nome, Apelido, Sexo, Data de Nascimento, CPF, RG, Código de Assinante, CEP, Logradouro, Número, Complemento, Bairro, Cidade, País, UF, Telefone Residencial - DDD, Telefone Residencial - Número, Telefone Comercial - DDD, Telefone Comercial - Número, Telefone Comercial - Ramal, Telefone Celular - DDD, Telefone Celular - Número, Situação Conjugal, Opções (*Listbox*) de Situação Conjugal, Tipo de Atividade, Opções (*Listbox*) de Tipo de Atividade, Grau de Instrução, Opções (*Listbox*) de Grau de Instrução, Possui Filhos, Quantidade de Filhos, e-Mail Adicional, Tipo do e-Mail Adicional, Pergunta Secreta, Opções (*Listbox*) de Pergunta Secreta, Resposta, Assinaturas de *Newsletter*, Nome - Indicados, e-Mail - Indicados, Relacionamento - Indicados, Opção de Marketing, Comando[13] e Mensagem para o Usuário[14];

FTR: Cadastro, Opções de Cadastro e SITE;

Grau de Complexidade: Alto (46 DETs e 3 FTRs).

- Inclusão ou Alteração de Cadastro - P. Jurídica (EI)

Descrição: inclusão de dados cadastrais de Pessoa Jurídica;

[12] Composto pelas tabelas PERGUNTA, GRUPO_PERGUNTA, SITE_GRUPO_PERGUNTA, TIPO_CAMPO, RESPOSTA_POSSIVEL.
[13] Tecla *enter* ou botão OK.
[14] De confirmação, erro, ajuda, etc.

DET: Endereço de e-Mail, Tipo de e-Mail, Senha, Confirmação de Senha, Razão Social, Nome Fantasia, Ano Fundação, CNPJ, Tipo de Organização, Opções (*Listbox*) de Tipo de Organização, Setor de Atuação, Opções (*Listbox*) de Setor de Atuação, Website, Código de Assinante, CEP, Logradouro, Número, Complemento, Bairro, Cidade, País, UF, Telefone Residencial - DDD, Telefone Residencial - Número, Telefone Comercial - DDD, Telefone Comercial - Número, Telefone Comercial - Ramal, Telefone Celular - DDD, Telefone Celular - Número, Situação Conjugal, Opções (*Listbox*) de Situação Conjugal, Tipo de Atividade, Opções (*Listbox*) de Tipo de Atividade, Grau de Instrução, Opções (*Listbox*) de Grau de Instrução, Possui Filhos, Quantidade de Filhos, e-Mail Adicional, Tipo do e-Mail Adicional, Pergunta Secreta, Opções (*Listbox*) de Pergunta Secreta, Resposta, Assinaturas de *Newsletter*, Nome - Indicados, e-Mail - Indicados, Relacionamento - Indicados, Opção de Marketing, Comando e Mensagem para o Usuário;

FTR: Cadastro, Opções de Cadastro e SITE;

Grau de Complexidade: Alto (49 DETs e 3 FTRs).

- Alteração de Email (EI)

Descrição: alteração da chave de Login; requer *Login* prévio;

DET: e-Mail novo, Tipo de e-Mail novo, Comando e Mensagem para o Usuário;

FTR: TICKET e Cadastro;

Grau de Complexidade: Baixo (4 DETs e 2 FTRs).

- Cadastramento/Descadastramento de *Newsletters* (EI)

Descrição: assinatura de *newsletters*; requer *Login* prévio;

DET: Seleção de *Newsletters*, Comando e Mensagem para o Usuário;

FTR: SITE, TIPO_NEWSLETTER e PESSOA_NEWSLETTER;

Grau de Complexidade: Médio (3 DETs e 3 FTRs).

- Cadastramento/Descadastramento de Ações de Marketing (EI)

Descrição: assinatura de promoções e publicidade; requer *Login* prévio;

DET: Seleção de Ações de Marketing, Comando e Mensagem para o Usuário;

FTR: SITE e PESSOA_OPCAO_MARKETING;

Grau de Complexidade: Baixo (3 DETs e 2 FTRs).

- Alteração de Senha (EI)

Descrição: alteração da senha de autenticação; envolve lógica de processamento para operações de criptografia; requer *Login* prévio;
DET: Senha Atual, Nova Senha, Confirmação de Nova Senha, Comando e Mensagem para o Usuário;
FTR: TICKET e Cadastro;
Grau de Complexidade: Baixo (5 DETs e 2 FTRs).

- Envio de Mensagens (EI)

Descrição: envio de mensagens através da opção "Fale Conosco"; requer *Login* prévio;
DET: Nome, e-Mail, Telefone-DDD, Telefone-Número, Assunto, Mensagem, Comando, Mensagem para o Usuário;
FTR: *não há*;
Grau de Complexidade: Baixo (8 DETs e 0 FTR).

- Consulta a Áreas de Acesso (EQ)

Descrição: consulta aos sites autorizados; requer *Login* prévio;
DET: Comando e Áreas de Acesso;
FTR: TICKET, SITE, PARCEIRO e RECURSO;
Grau de Complexidade: Médio (2 DETs e 4 FTRs).

- Acesso a Recursos do Site (EO)

Descrição: navegação nos recursos do Site; envolve lógica de processamento para verificação de acesso em área restrita;
DET: Seleção de Recurso, Recurso ou Tela de Login;
FTR: TICKET, PARCEIRO, SITE e RECURSO.
Grau de Complexidade: Médio (3 DETs e 4 FTRs).

- *Login* (EO)

Descrição: processo de autenticação e autorização às áreas restritas para usuários registrados; envolve lógica de processamento para operações de criptografia;

DET: e-Mail, Senha, Comando e Mensagem para o Usuário;
FTR: TICKET, PARCEIRO, SITE, Cadastro e RECURSO;
Grau de Complexidade: Médio (4 DETs e 5 FTRs).

- Salvar Senha (EO)

Descrição: gravação de senha em arquivo de *cookie* (Ticket), a fim de agilizar o processo de autenticação; requer *Login* prévio;
DET: Opção de Salvar Senha;
FTR: TICKET;
Grau de Complexidade: Baixo (1 DET e 1 FTR).

- Seleção do Perfil do Usuário (EO)

Descrição: seleção do perfil (assinante ou comprador em banca) para fins de acesso às áreas restritas; requer *Login* prévio;
DET: Seleção do Perfil do Usuário, Comando, Tela de Acesso Restrito a Assinantes ou a Compradores em Banca;
FTR: *não há*;
Grau de Complexidade: Baixo (3 DETs e 0 FTR).

- Autenticação via Código de Assinante (EO)

Descrição: processo de autenticação e autorização às áreas restritas para usuários assinantes; envolve lógica de processamento para operações de criptografia; requer *Login* prévio;
DET: Código de Assinante, Comando, Mensagem para o Usuário;
FTR: TICKET, SITE, RECURSO;
Grau de Complexidade: Baixo (3 DETs e 3 FTRs).

- Autenticação via Código de Banca (EO)

Descrição: processo de autenticação e autorização às áreas restritas para usuários compradores em banca; envolve lógica de processamento para operações de criptografia; requer *Login* prévio;
DET: Código de Banca, Comando, Mensagem para o Usuário;
FTR: TICKET, SITE, RECURSO;
Grau de Complexidade: Baixo (3 DETs e 3 FTRs).

- Preenchimento de pesquisa de enriquecimento (EI)

Descrição: preenchimento de pesquisa após o *Login*, com perguntas variáveis por site; requer *Login* prévio;

DET: Resposta, Comando e Mensagem para o Usuário;

FTR: Pesquisa;

Grau de Complexidade: Baixo (3 DETs e 1 FTR).

5.3.4 Cálculo do Valor Não Ajustado

Usando a Tabela 3-4 e a fórmula para cálculo do valor de UFP, conforme descrito na Seção 3.1.5, chegamos ao seguinte resultado:

$$
\begin{aligned}
UFP_{ILF} &= 7*4 + 10*1 + 15*0 = 38 \\
UFP_{EIF} &= 5*6 + 7*0 + 10*0 = 30 \\
UFP_{EI} &= 3*4 + 4*1 + 6*2 = 28 \\
UFP_{EO} &= 4*5 + 5*2 + 7*0 = 30 \\
UFP_{EQ} &= 3*0 + 4*1 + 6*0 = 4
\end{aligned}
$$

$$UFP = 38 + 30 + 28 + 30 + 4$$

UFP = 130

5.3.5 Cálculo do Fator de Ajuste

Para o cálculo do Fator de Ajuste, definiremos os pesos para cada uma das 14 Características Gerais do Sistema (CGS) apresentadas na Tabela 3-6 conforme as diretrizes definidas pelo Manual Prático de Contagem (MPC) do método. Por exemplo, a característica "Comunicação de Dados", que trata do grau com que a aplicação se comunica com o processador, deverá ser pontuada com o fator máximo (5), uma vez que a aplicação possui um comportamento mais complexo do que um simples front-end e envolver vários protocolos de comunicação. Em outro exemplo, a característica "Processamento de Dados Distribuídos", que considera o grau com que a aplicação transfere dados para seus componentes, também é pontuado com o valor máximo, devido à arquitetura distribuída da aplicação. Estes e os demais valores se encontrados apresentados na Tabela 5-1.

De acordo com as fórmulas apresentadas na Seção 3.1.6, temos portanto:

$$TDI = \sum_{i=1}^{14} peso_i \text{ (CGS)} = 49$$

$$VAF = (TDI * 0{,}01) + 0{,}65 = (49 * 0{,}01) + 0{,}65 = 1{,}14$$

Tabela 5-1 Estudo de Caso - FPA - Características Gerais do Sistema

CGS	Peso	CGS	Peso
Comunicação de Dados	5	Atualização *Online*	3
Processamento de Dados Distribuídos	5	Processamento Complexo	3
Performance	5	Reusabilidade	3
Configuração Altamente Utilizada	2	Facilidade de Instalação	3
Taxa de Transações	5	Facilidade de Operação	3
Entrada de Dados *Online*	5	Múltiplos Sites	3
Eficiência do Usuário Final	4	Facilidade de Alteração	0

5.3.6 Cálculo do Valor Ajustado

Conforme já mencionado, o tipo de contagem adotado foi o de um projeto de desenvolvimento e não um de melhoria, apesar de já existir uma versão anterior do sistema, uma vez que o sistema foi totalmente reescrito. Não obstante, alguma função de conversão do modelo de dados se fez necessário, com um esforço estimado em cerca de 10% do valor total. Portanto, o valor de Pontos de Função Não Ajustado das funções de conversão - CFP pode ser calculado como:

$$CFP = 10\% * UFP = 0,1 * 130 = 13$$

Portanto, de acordo com a fórmula apresentada na Seção 3.1.7, temos que o valor de Pontos de Função Ajustado do projeto de desenvolvimento - DFP é igual a:

$$DFP = (UFP + CFP) * VAF = (130 + 13) * 1,14$$

DFP = 163

Considerando a curva de Taxa de Produtividade x Tamanho Funcional, produzida pelo International Software Benchmarking Standards Group - ISBSG [ISB 98] para projetos de desenvolvimento, chega-se por inferência a uma Taxa de Produtividade de 0,093 AFP/HH para o volume de FPs em questão, o que resulta em um esforço estimado de

$$Esforço = 163 (AFP) / 0,093 (AFP/HH)$$

Esforço = 1.753 Homens-Hora

5.4 Contagem pelo método de Pontos de Caso de Uso

Conforme apresentado na Seção 4.4.1, o método Pontos de Caso de Uso requer, inicialmente, a contagem dos Atores e dos Casos de Uso. Considerando o diagrama da Figura 5-2 e as premissas indicadas nas Tabelas 4-2, chegamos a seguinte classificação dos Atores envolvidos no caso em questão:

- 2 Atores Simples (Sistemas de Assinatura e Pagamento Online);
- 1 Ator Médio (Sistema de Newsletter);
- 2 Atores Complexos (Internauta e Parceiro);

Novamente considerando o diagrama da Figura 5-2 e as regras para definição da complexidade dos Casos de Uso, listadas na Seção 4.4.1, chegamos a seguinte classificação dos Casos de Uso envolvidos no caso em questão:

- 2 Casos de Uso Simples (Importa/Exporta Cadastro e Envia Mensagem)
- 1 Caso de Uso Médio (Verifica Restrição)
- 5 Casos de Uso Complexos (Autoriza e Autentica Acesso, Gerencia Ticket, Cadastro e Política de Acesso)

De acordo com a fórmula apresentada na Seção 4.4.1, temos que o Peso total dos Atores Não Ajustado - UAW é igual:

$$UAW = AWs*\sum As + AWm*\sum Am + AWc*\sum Ac = 2*1 + 1*2 + 2*3 = 10$$

Da mesma forma, o Peso total dos Casos de Uso Não Ajustado - UUCW é igual:

$$UUCW = UCWs*\sum UCs + UCWm*\sum UCm + UCWc*\sum UCc = 2*5+1*10+5*15 = 95$$

E portanto o Peso total Não Ajustado - UUCP é igual a:

$$UUCP = UAW + UUCW = 10 + 95$$

UUCP = 105

Em seguida, calcula-se o Fator de Ajuste atribuindo-se um grau de influência a cada um dos Fatores de Complexidade Técnica e de Ambiente apresentadas nas Tabelas 4-4 e 4-5. Por exemplo, o fator T1 ("Sistema Distribuído") deverá ser pontuado com o valor máximo (5), devido à arquitetura distribuída da aplicação. Em outro exemplo, o fator E1 ("Familiaridade com RUP ou outro processo formal") também é pontuado com o valor máximo, uma vez que a equipe do projeto possui de fato tal atributo (outras considerações a respeito podem ser encontradas na Seção 5-6). Estes e os demais valores encontram-se apresentados na Tabelas 5-2 e 5-3.

Temos portanto:

$$\text{TFactor} = \sum_{i=1}^{13} \text{peso}_i \ (\text{TCF}) = 57$$

$$\text{TCF} = 0{,}6 + (0{,}01 * \text{TFactor}) = 0{,}6 + (0{,}01 * 57) = 1{,}17$$

$$\text{EFactor} = \sum_{i=1}^{8} \text{peso}_i \ (\text{EF}) = 28{,}5$$

$$\text{EF} = 1.4 + (-0.03 * \text{EFactor}) = 1.4 + (-0.03 * 28{,}5) = 0{,}55$$

Finalmente, os valores do Peso total Não Ajustado e dos Fatores de Complexidade Técnica e de Ambiente são utilizados para se calcular o valor do Ponto de Caso de Uso Ajustado - AUCP, como segue:

$$\text{AUCP} = \text{UUCP} * \text{TCF} * \text{EF} = 105 * 1{,}17 * 0{,}55$$

AUCP = 68

Tabela 5-2 Estudo de Caso - UCP - Fatores Técnicos

Grau de Influência												
T1	**T2**	**T3**	**T4**	**T5**	**T6**	**T7**	**T8**	**T9**	**T10**	**T11**	**T12**	**T13**
5	5	5	3	2	3	3	5	0	5	4	5	0

Tabela 5-3 Estudo de Caso - UCP - Fatores de Ambiente

Grau de Influência							
E1	**E2**	**E3**	**E4**	**E5**	**E6**	**E7**	**E8**
5	5	5	5	3	1	0	3

Considerando um valor médio de 20 Homens-Hora / AUCP [KAR 93], chega-se a uma estimativa de esforço de:

Esforço = 20 HH/AUCP * 68 AUCP

Esforço = 1.360 Homens-Hora

5.5 Contagem pelo método Web Points

O método Web Points, conforme apresentado na Seção 4.4.2, tem como base de contagem as páginas HTML do sistema. Considerando as interfaces do sistema, descritas no Apêndice A, e as regras para definição do grau de complexidade apresentados na Tabela 4-6, pode-se construir a Tabela 5-4.

Tabela 5-4 Estudo de Caso - Web Points - Páginas HTML

Interface	# Palavras	# Links e Componentes	Complexidade
Login	198	7	Baixa
Acesso a Áreas Restritas	153	7	Baixa
Assinante	157	7	Baixa
Comprador em Banca	174	7	Baixa
Central de Relacionamento	171	12	Baixa
Alterar Dados Cadastrais - P. Física	741	17	Alta
Alterar Dados Cadastrais - P. Jurídica	680	16	Alta
Alterar e-Mail	129	4	Baixa
Alterar Senha	152	5	Baixa
Recebimento de Newsletter	326	5	Baixa
Ações de Marketing	175	5	Baixa
Minhas Áreas de Acesso	106	3	Baixa
Fale Conosco	145	4	Baixa

Portanto, de acordo os valores apresentados na tabela 5-4 e a fórmula para definição do número de Web Points - WP, descrita na Seção 4.4.2, temos que:

$$WP = W_B * \sum WP_B + W_M * \sum WP_M + W_A * \sum WP_A = 11*13 + 0*5 + 2*11$$

WP = 165

Utilizando como referência para a Taxa de Produtividade média o mesmo valor de 0,05 WP/HH obtida por Cleary [CLE 00], chega-se a uma estimativa de esforço de:

Esforço = 165 (WP) / 0,05 (WP/HH)

Esforço = 3.300 Homens-Hora

5.6 Dados Reais

O projeto de desenvolvimento e implantação do Sistema de Identificação de Usuário iniciou-se em primeiro de julho de 2004 e foi concluído em 30 de novembro do mesmo ano.

A equipe consistiu de um coordenador, que também atuou como analista/projetista nas fases iniciais do projeto, três outros analistas/programadores sêniores em tempo parcial e uma consultoria para as atividades de *Web designing*.

O esforço total perfez um total aproximado de **2.030 Homens-Hora**. Para efeito de análise comparativa foram descontadas também as perdas no cronograma relativas à indefinição de escopo. Estão contabilizadas na carga horária total as fases de análise, projeto, codificação, testes e implantação.

Vale mencionar o fato de que todos os programadores estiveram de alguma forma envolvidos com a versão anterior do sistema, cuja experiência contribuiu para uma Taxa de Produtividade acima da média.

. . .

No próximo capítulo serão discutidos os resultados obtidos a partir do estudo de caso e apresentadas as conclusões do trabalho.

Capítulo 6

Conclusão

Os resultados obtidos no estudo de caso estão resumidos na Tabela 6-1.

Tabela 6-1 Estudo de Caso - Resultados

Métodos de Contagem	Esforço (HH)	Diferença entre Estimado e Real (%)
IFPUG FPA	1.753	-14%
Pontos de Caso de Uso	1.360	-33%
Web Points	3.300	+63%
Caso Real	**2.030**	**-**

Inicialmente observa-se uma grande discrepância (63%) entre o valor obtido pelo método Web Point e o caso real. Isto demonstra que este método não é adequado para aplicações Web mais complexas, com funcionalidades que vão além de uma simples disponibilização de conteúdo estático (páginas HTML). Para aplicações que possuem um alto grau de interação com usuário e comportamento tipicamente dinâmico, como aquelas encontradas no estudo de caso apresentado, parece claro que uma métrica mais elaborada se faz necessária.

Neste sentido, o método IFPUG FPA continua naturalmente a ser uma opção que deve ser considerada. Outros métodos decorrentes ainda das primeiras gerações de métricas funcionais (descritos na Seção 2.2) poderiam se mostrar mais adequados em determinados aspectos. O método Feature Point, por exemplo, se aplica melhor à contagem de algoritmos e lógica de processamento, sendo indicado portanto em situações onde há esforço adicional envolvido em implementação de determinadas rotinas, como as de criptografia, ou de lógicas elaboradas como o processo de autenticação, ambos presentes no caso em estudo. Já o método COSMIC FFP, por utilizar um conceito de visão em camadas, poderia melhor refletir as questões de infraestrutura específicas do ambiente Web, como demonstrado em [ROL 00].

Contudo, apesar da diferença entre os resultados deste método e o caso real (14%) estar dentro da faixa considerável aceitável, sobretudo se levarmos em consideração a fase inicial do projeto em que a contagem foi realizada, conforme exposto na Seção 2.3.1, devemos considerar na análise dos resultados o esforço, não contabilizado, relacionado aos

processos de *authoring* e de *designing*, conforme apresentado na Seção 4.3. Como exemplo, uma aplicação pode demandar uma quantidade considerável de homens-hora relacionada a atividades de tratamento de imagens. Apesar de esta situação não ter ocorrido no estudo de caso em questão, uma vez que a demanda por elementos de multimídia neste sistema foi mínima (menos do que 10 % do esforço total), tais esforços mesmo que pequenos não foram considerados na contagem da FPA. Para este fim, um método que estenda a FPA (ou suas variantes) para contemplar tal esforço, como Web Objects (apresentado na Seção 4.4.3), parece ser a solução mais adequada. Vale mencionar que a mesma conclusão obteve Ruhe *et al.* [RUH 03] em seus estudos, que visaram obter uma validação empírica do método. Alternativamente, alguns autores, como Mendes [MEN 01], têm trabalhado no sentido de estabelecer métricas que possam contabilizar, de forma mais precisa, este tipo de atividade.

A métrica de Pontos de Caso de Uso, apesar de apresentar um resultado inferior ao método IFPUG FPA (diferença de 33%), não pode ser totalmente descartada, sobretudo pela sua facilidade de utilização. Contudo, seu emprego deve ser realizado de forma criteriosa, uma vez que, além do problema do nível de decomposição a ser adotado, já descrito na Seção 4.4.1, outras questões como as variações no estilo de especificação dos Casos de Uso e na sua formalização podem dificultar um processo de medição [SMI 99].

O resultado obtido demonstrou que a contagem de Pontos de Função através do método FPA se mostrou suficiente para contabilizar a maioria das funções presentes em aplicações em Web e consequentemente útil para efeito de estimativa de esforço neste ambiente. Contudo, aspectos inerentes desta plataforma podem, conforme o caso, contribuir para a produção de resultados aquém do real, em virtude de eventuais esforços não contabilizados, e que portanto métodos alternativos devem ser investigados a fim de se obter métodos mais genéricos que atendam a todos os tipos de requisitos existentes neste tipo de sistema.

Por último, é importante mencionar as limitações envolvidas no estudo de caso em questão. Além dos eventuais desvios inerentes ao próprio processo de contagem, deve ser considerado o reduzido escopo do projeto analisado, com efeitos tanto na restrição das funcionalidades expostas, quanto na precisão das estimativas, conforme exposto na Seção 2.2.1.

6.1 Trabalhos Futuros

Recomenda-se na seqüência um estudo mais abrangente sobre a eficácia de um método que estenda a FPA a fim de contabilizar aspectos específicos da Web, como por exemplo o Web Objects, utilizando-se um universo mais amplo e diversificado de aplicações Web, onde tais aspectos possam estar presentes em maior intensidade e complexidade. Recomenda-se também a análise de um método-base mais atualizado do que o próprio IFPUG FPA, como por exemplo o COSMIC FFP. Desta forma, poderá ser obtido um novo método que contemple todas as particularidades aqui discutidas, e ao mesmo tempo aproveite os conceitos já validados no modelo original.

Lista de Abreviações

AFP	Adjusted Function Point
APF	Análise por Ponto de Função
API	Application Programming Interface
AUCP	Adjusted Use Case Point
AW	Actor Weight
B2B	Business-to-Business
B2C	Business-to-Commerce
BFPUG	Brazilian Function Point Users Group
CASE	Computer-Aided Software Engineering
CGI	Common Gateway Interface
CMM	Capability Maturity Model
COBRA	Cost Estimation, Benchmarking, and Risk Assessment
COCOMO	Constructive Cost Model
COSMIC	Common Software Measurement International Consortium
DET	Data Element Type
DFP	Development Function Point
EF	Environment Factor
EI	External Input
EIF	External Interface File
EJB	Enterprise Java Bean
ENUPF	Encontro Nacional de Usuários de Usuários de Ponto de Função
EO	External Output
EQ	External Query
FFP	Full Function Point
FPA	Function Point Analysis
FSM	Finite State Machine
FTP	File Transfer Protocol
FTR	File Type Referenced
GSC	Geral System Characteristic
GUI	Graphical User Interface
HH	Homem-Hora
HTML	Hypher Text Mark-up Language
IEC	International Electrotechnical Commission

ISBSG	International Software Benchmarking Standards Group
ILF	Internal Logical File
ISO	International Organization for Standardization
JPEG	Joint Photographic Experts Group
JSP	Java Server Page
JTC	Joint Technical Committee
MCT	Ministério da Ciência e Tecnologia
MIS	Management Information System
MPC	Manual Prático de Contagem
NESMA	Nederlandse Software Metrieken Associatie
OMT	Object Modeling Technique
OO	Object Oriented
OOFP	Object Oriented Function Point
RET	Register Element Type
RUP	Rational Unified Process
SC	SubCommittee
SEPIN	Secretaria de Política de Informática
SLOC	Source Line Of Code
SRS	Software Requirement Specification
TCF	Technical Complexity Factor
TCP/IP	Transmission Control Protocol / Internet Protocol
TDI	Total Degree Influence
UAW	Unadjusted Actor Weight
UC	Use Case
UCW	Use Case Weight
UFR	User Functional Requirement
UFP	Unadjusted Function Point
UKSMA	United Kingdom Software Metrics Association
UML	Unified Modeling Language
UUCP	Undajusted Use Case Point
UUCW	Unadjusted Use Case Weight
VAF	Value Adjust Factor
WG	Work Group
WP	Web Point
XML	Extensible Markup Language
XP	Extreme Programming

Referências Bibliográficas

[ABR 03] ABRAHÃO S., PASTOR, O. "Measuring the Functional Size of Web Applications" **International Journal of Web Engineering and Technology**, 1(1): 5-16, 2003.

[ABR 98] ABRAN, A. *et al.* "Design of a Functional Size Measurement for Real-Time Software" Research report no. 13, Software Engineering Management Research Laboratory, **Universite du Quebec a Montreal, Canada**, November 1998.

[ABR 99] ABRAN, A. *et al.* "COSMIC FFP Measurement Manual" COSMIC-FFP ISO/IEC 19761 Version 2003, Software Engineering Management Research Laboratory, **Universite du Quebec a Montreal, Canada**. Disponível em <http://www.lrgl.uqam.ca/cosmic-ffp/manual.jsp>. Obtido em agosto de 2004.

[ALB 79] ALBRECHT, A. J. "Measuring Application Development Productivity" **IBM Applications Development Symposium**, Monterey, USA, 1979.

[AND 01] ANDA, B. *et al.* "Estimating Software Development Effort Based on Use Cases - Experiences from Industry", UML 2001 - The Unified Modeling Language. Modeling Languages, Concepts, and Tools, **4th International Conference**, Toronto, Canada, October 1-5, 2001, LNCS 2185 Springer.

[BAN 01] BANERJEE, G. "Use Case Points - An Estimation Approach" August, 2001. Disponível em <http://www.bfpug.com.br/Artigos/UCP/Banerjee-UCP_An_Estimation_Approach.pdf>. Obtido em agosto de 2004.

[BEC 99] BECK, K. "Extreme Programming Explained: Embrace Change" 1st edition, **Addison-Wesley Pub Co**; October , 1999

[BER 03] BERTOLAMI, M. A. "Una Propuesta de Análisis de Puntos Función aplicado a LEL y Escenarios" Facultad de Informática, **Universidad Nacional de La Plata**, Argentina, 2003.

[BFP 98] BFPUG Web Site. Disponível em <http://www.bfpug.com.br>. Obtido em agosto de 2004.

[BOE 81] BOEHM, B. W. "Software Engineering Economics" Englewood Cliffs, NJ, **Prentice-Hall**: 1981.

[BOE 95] BOEHM, B. W. *et al.* "The COCOMO 2.O Software Cost Estimation Model", **Proceedings of the 1995 ISPA Conference**, San Diego, USA, May, 1995.

[BRI 98] BRIAND, L. C. *et. al* "COBRA: A Hybrid Method for Software Cost Estimation, Benchmarking and Risk Assessment" **Proceeding of the 20th International Conference on Software Engineering**, pp. 390-399, April, 1998.

[BUG 04] BUGLIONE, L. "Functional Size Measurement". Disponível em <http://www.geocities.com/lbu_measure/fpa/fpa.htm>. Obtido em agosto/2004.

[CAL 98] CALDIERA, G. *et al.* "Definition and Experimental Evaluation of Function Points for Object-Oriented Systems" **5th International Software Metrics Symposium**, IEEE Proceeding, November 1998.

[CLE 00] CLEARY, D. "Web-based development and functional size measurement" Proc. IFPUG, **Annual Conference,** San Diego, USA, 2000.

[COS 04] COSMIC Web Site. Disponível em <http://www.cosmicon.com/historycs.asp>. Obtido em agosto de 2004.

[DEK 99] DEKKERS, C. A. "Use Cases and Function Points -- Where's the Fit?" **IT Metrics Strategies**, January 1999.

[DEK 01] DEKKERS, C. A., AGUIAR, M. "Applying Function Point Analysis to Requirements Completeness" **Crosstalk**, February 2001

[FEN 96] FENTON, N., PFLEEGER, S. "Software Metrics, A rigorous and Practical Approach" Second Edition, **PWS Publishing Company**, 1996.

[FET 99] FETCKE, T. "A Generalized Structure for Function Point Analysis" **International Workshop on Software Measurement**, Canada, September 1999.

[FRE 03] FREIRE, H. "Calculando Estimativas: o Método de Pontos de Caso de Uso" **Developer's Magazine,** número 78, Fevereiro de 2003.

[FUR 97] FUREY, S. "Why We Should Use Function Points" **IEEE Computer**, pp. 28-30, May/April 1997.

[GAR 01] GARMUS, D., HERRON, D. "Function Point Analysis - Measurement Practices for Successful Software Projects", **Addison-Wesley**, 2001.

[IFP 86] **IFPUG** Web Site. Disponível em <http://www.ifpug.org>. Obtido em agosto de 2004.

[ISB 98] "Worlwide Software Development: The Benchmark", Release 5, **ISBSG**, March 1998.

[ISO 98] "Software Measurement - Functional Size Measurement - Part 1 Definition of Concepts" **AS/NZS ISO/IEC** 14143-1:1999.

[JEF 96] JEFFERY, R. *et al.* "Sizing and estimating the coding and unit testing effort for GUI systems" **3rd International Software Metrics Symposium**, IEEE Proceedings, March 1996

[JON 87] JONES, C. "A Short History of Function Points and Feature Points" **Software Productivity Research Inc.**, USA, 1987.

[KAR 93] KARNER, G. "Use Case Points - Resource Estimation for Objectory Projects" **Objective Systems SF AB** (Rational Software), 1993.

[KUL 00] KULIK, P. "A Practical Approach to Software Metrics" **IEEE IT Professional**, Volume 2, Number 1, January/February 2000.

[LON 04a] LONGSTREET, D. "Use Cases and Function Points". Disponível em <http://www.ifpug.com/Articles/usecases.htm>. Obtido em julho de 2004.

[LON 04b] LONGSTREET, D. "Estimating Software Development". Disponível em <http://www.ifpug.com/Articles/estimating.htm>. Obtido em julho de 2004.

[MCT 01] "Qualidade e Produtividade no Setor de Software Brasileiro" Secretaria de Política de Informática - SEPIN, **Ministério da Ciência e Tecnologia**, 2001.

[MEN 01] MENDES, E. *et al.* "Web Metrics - Estimating, Design and Authoring Effort", **IEEE Multimedia**, 1070-986X/01, January-March 2001.

[MEN 03] MENDES, E. *et al.* "Early Web Size Measures and Effort Prediction for Web Costimation", **Proceedings of the Ninth International Software Metrics Symposium (METRICS'03)**, IEEE 1530-1435/03, 2003.

[NES 98] Nederlandse Software Metrieken Associatie. Disponível em < http://www.nesma.nl/english/earlyfpa.htm>. Obtido em outubro de 2004.

[OFF 02] OFFUTT, J. "Quality Attributes of Web Software Applications" **IEEE Software**, March / April 2002.

[PAS 01] PASTOR, O. *et al.* "The OO-Method Approach for Information Systems Modelling: From Object-Oriented Conceptual Modeling to Automated Programming", **Information Systems**, Vol. 26, N. 7 (2001) 507–534.

[PCE 99] PCEI Working Group "Parametric Estimating Handbook" 2nd. Edition, **International Society of Parametric Analysts - ISPA**, 1999. Disponível em <http://www.ispa-cost.org>. Obtido em agosto de 2004.

[PRI 97] PRICE Systems, LLC, "PRICE S - Reference Manual" Version 3.0, Mt Laurel, NJ, **Lockheed-Martin**: 1997.

[REI 02a] REIFER, D. "Estimating Web Development Costs: There Are Differences" **Crosstalk**, pp. 13-17, June 2002.

[REI 02b] REIFER, D. "Web Objects Counting Conventions" Disponível em <http://www.reifer.com>. Obtido em outubro de 2004.

[ROD 02] RODRIGUEZ, D. *et al.* "A Generic Model and Tool Support for Assessing and Improving Web Processes" **8th International Software Metrics Symposium**, IEEE Proceeding, 4-7, pp. 141-152, June 2002.

[ROE 00] ROETZHEIM, W. H. "Estimating Internet Development" **Software Development Magazine**, August 2000.

[ROL 00] ROLLO, T. "Sizing E-commerce" **ACOSM 2000**.

[RUH 03a] RUHE, M. *et al.* "Cost Estimation for Web Applications" **Proceedings of the 25th International Conference on Software Engineering - ICSE**, Portland, USA, pp. 285-294, 2003.

[RUH 03b] RUHE, M. *et al.* "Using Web Objects for Estimating Software Development Effort for Web Applications" **9th International Software Metrics Symposium,** IEEE Proceeding, 3-5, pp. 30- 37, September 2003.

[SMI 99] SMITH, J. "The Estimation of Effort and Size Based on Use Cases" **Rational Software White Paper**, 1999.

[STE 98] STEPHEN, G. *et al.* "Metric Selection for Effort Assessment In Multimedia Systems Development" **5th International Software Metrics Symposium**, IEEE Proceeding, pp. 97-104, 20-21 November 1998.

[STU 96] STUKE, R. D. "Software Estimating Technology: A Survey" **CrossTalk**, May 1996.

[SYM 98] SYMONS, C. "Function Point Analysis: Difficulties and Improvements" **IEEE Transactions on Software Engineering**, 14(1), pp. 2-11, 1998.

[SYM 01] SYMONS, C. "Come back Function Point Analysis (modernised) - All is forgiven!" **Software Measurement Services Ltd**, 2001.

[UKS 98] UKSMA Metrics Practices Committee "MkII Function Point Analysis - Counting Practices Manual" version 1.3.1, **United Kingdom Software Metrics Association - UKSMA**, September 1998. Disponível em <http://www.uksma.co.uk/public/mkIIr131.pdf>. Obtido em agosto de 2004.

[VAZ 04] VAZQUEZ, C. E. *et al.* "Análise de Pontos de Função - Medição, Estimativas e Gerenciamento de Projetos de Software" 2a. Edição, **Editora Érica**, 2004.

[WHI 92] WHITMIRE, S. A. "3D Function Points: Scientific and Real-time Extensions to Function Points" **Pacific Northwest Software Quality Conference**, 1992.

Apêndice A

Estudo de Caso - Telas do Sistema

As interfaces do usuário, conforme detalhado na Seção 5.1.4, dividem-se em dois grupos de funções: aquelas utilizadas no processo de autenticação, a partir da tela de *Login,* e aquelas relacionadas às funções de cadastramento, acessadas a partir de uma tela inicial chamada Central de Relacionamento, conforme mostra o mapa da Figura A-1.

Figura 6-1 Estudo de Caso - Mapa da interface

As telas relativas ao processo de *Login* estão apresentadas nas Figuras A-2 a A-5.

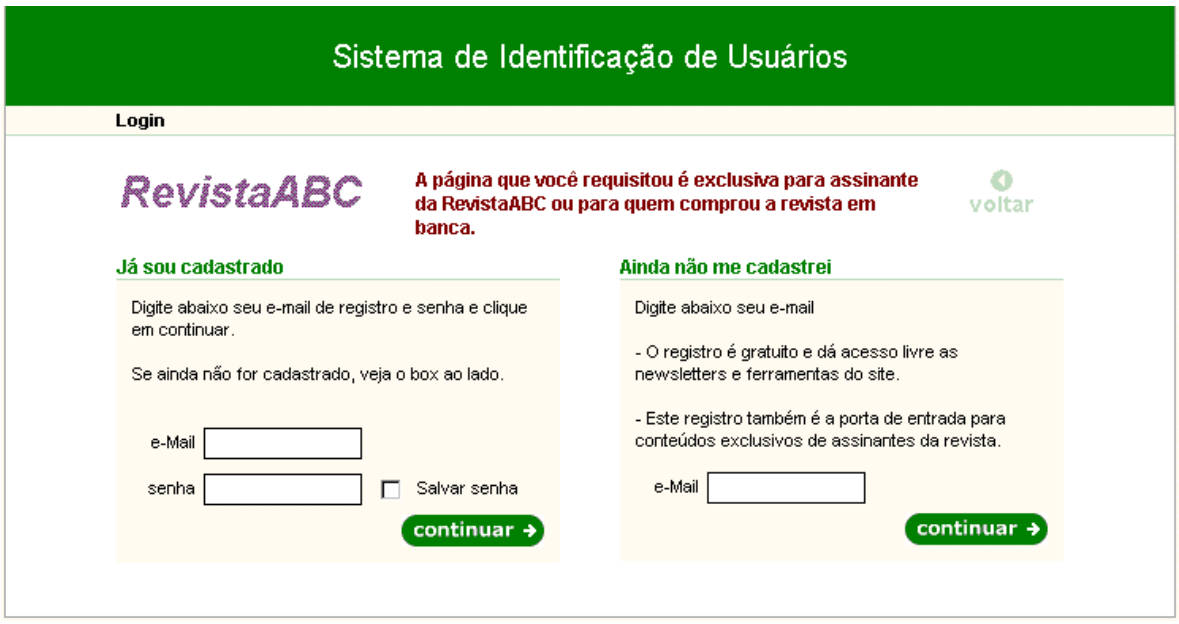

Figura 6-2 Estudo de Caso -Tela de Login

Figura 6-3 Estudo de Caso - Tela de Acesso a Áreas Restritas

Figura 6-4 Estudo de Caso - Tela de Acesso a Áreas Restritas - Assinante

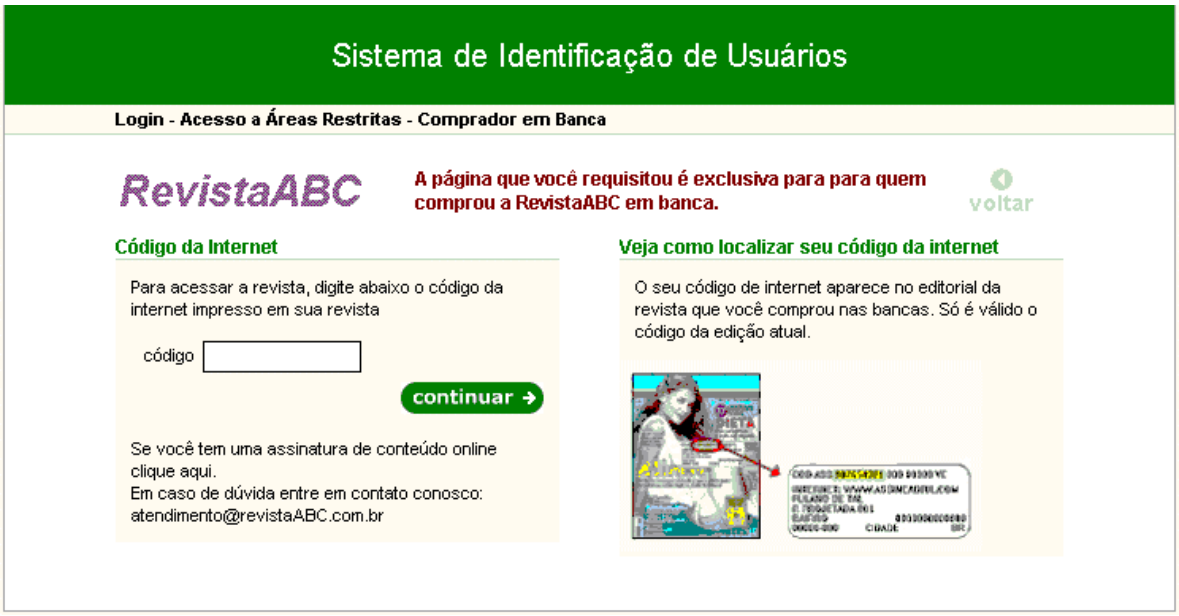

Figura 6-5 Estudo de Caso - Tela de Acesso a Áreas de Acesso - Comprador em Banca

Todas as telas que compõem a Central de Relacionamento podem ser visualizadas nas Figuras A-6 a Figura A-13.

Figura 6-6 Estudo de Caso - Tela Central de Relacionamento

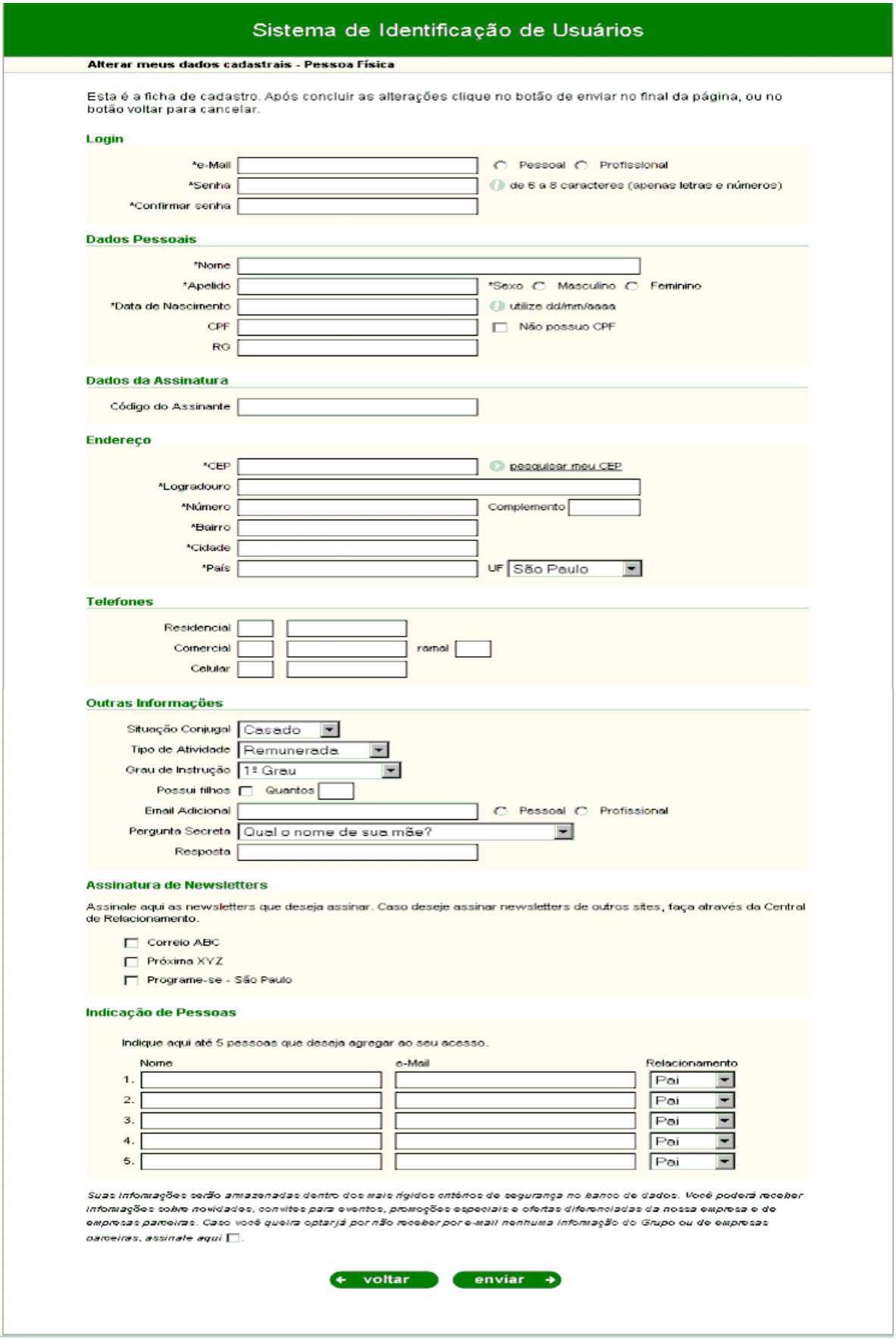

Figura 6-7 Estudo de Caso - Alterar Dados Cadastrais - P. Física

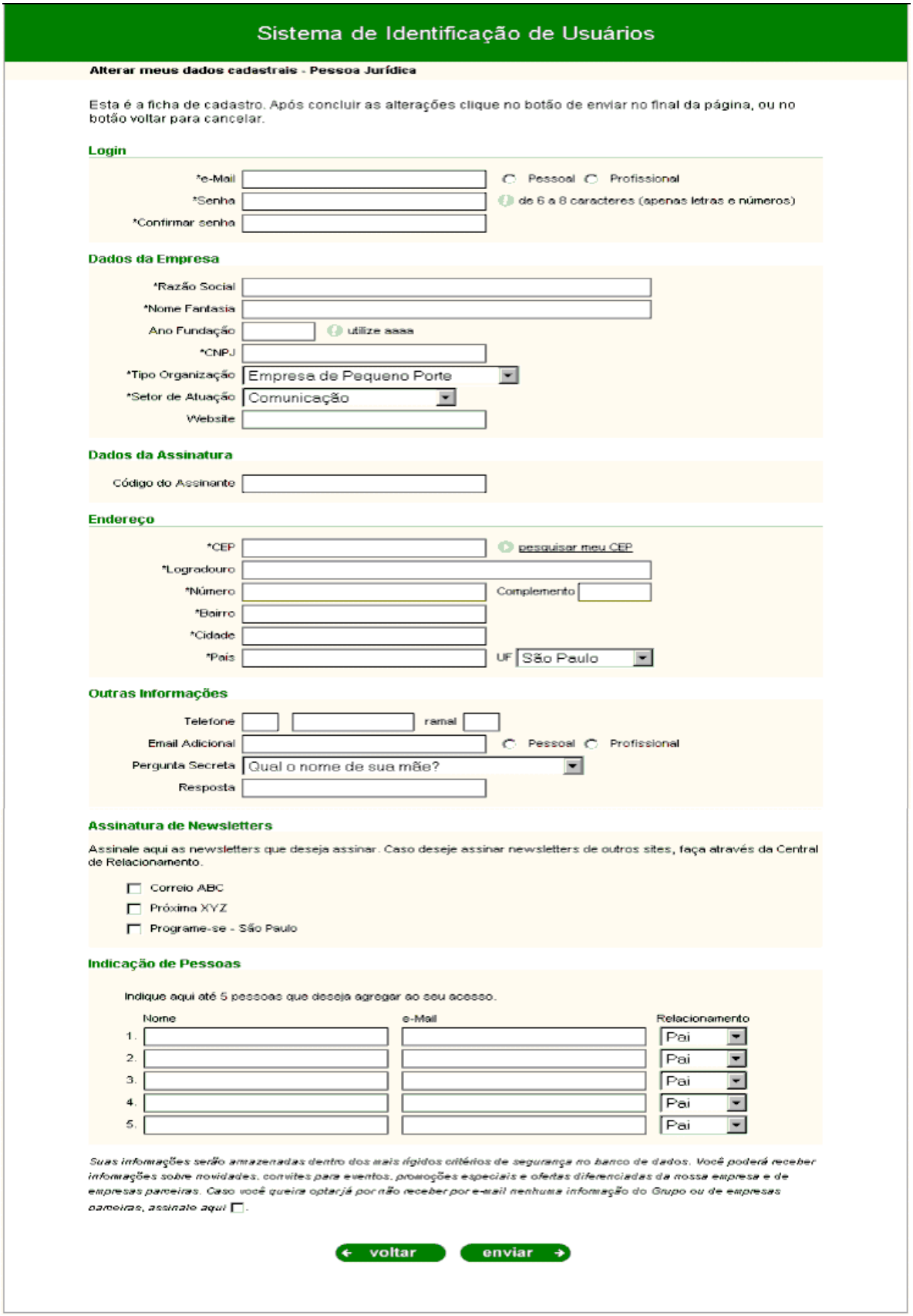

Figura 6-8 Estudo de Caso - Tela Alterar Dados Cadastrais - P. Jurídica

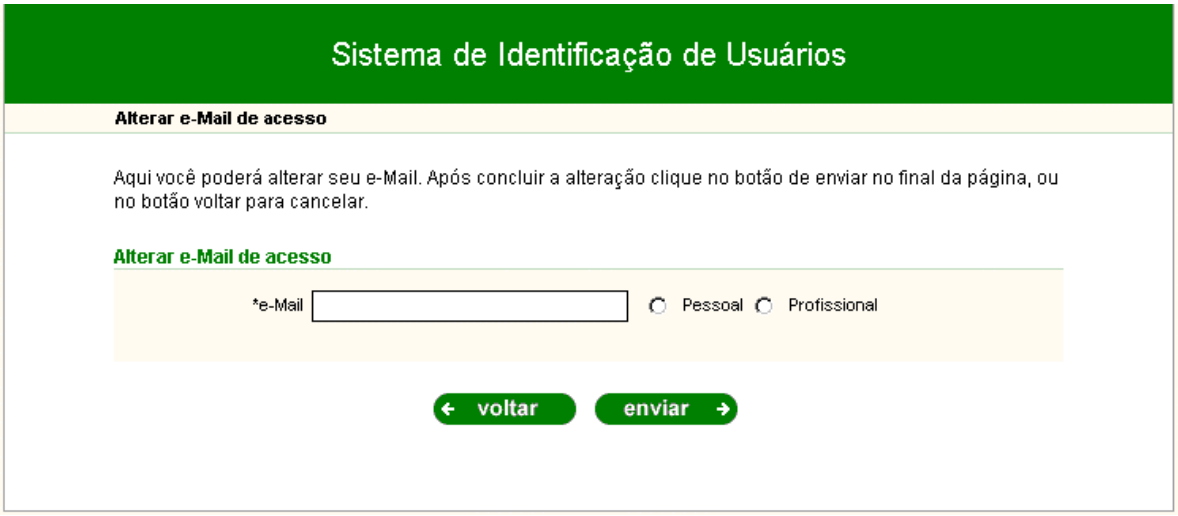

Figura 6-9 Estudo de Caso - Tela Alterar e-Mail de acesso

Figura 6-10 Estudo de Caso - Tela Alterar Senha de acesso

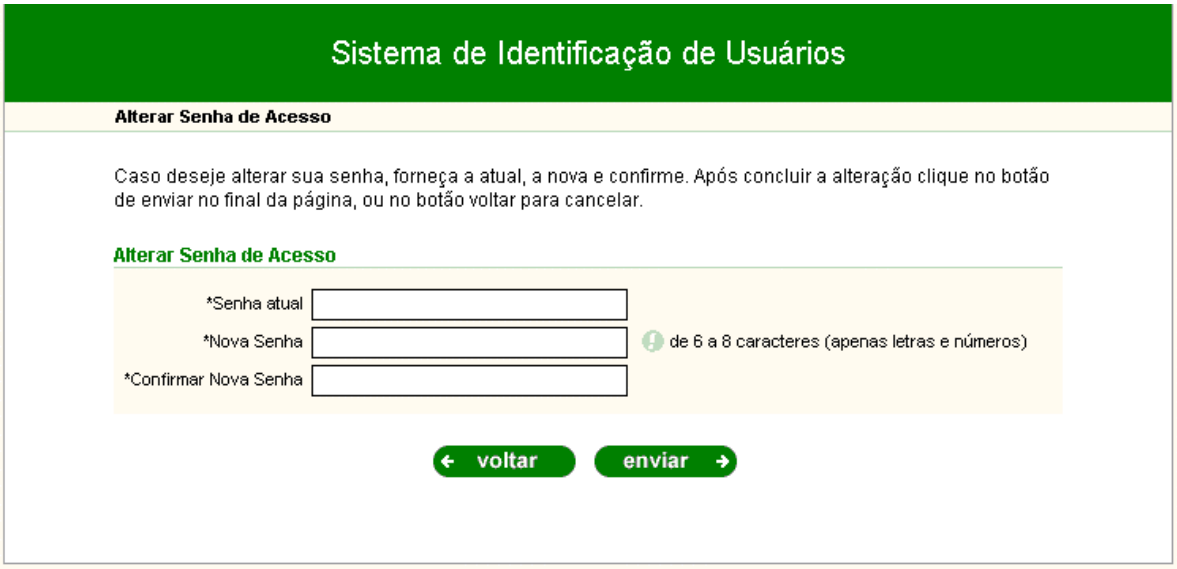

Figura 6-11 Estudo de Caso - Tela Recebimento de Newsletters

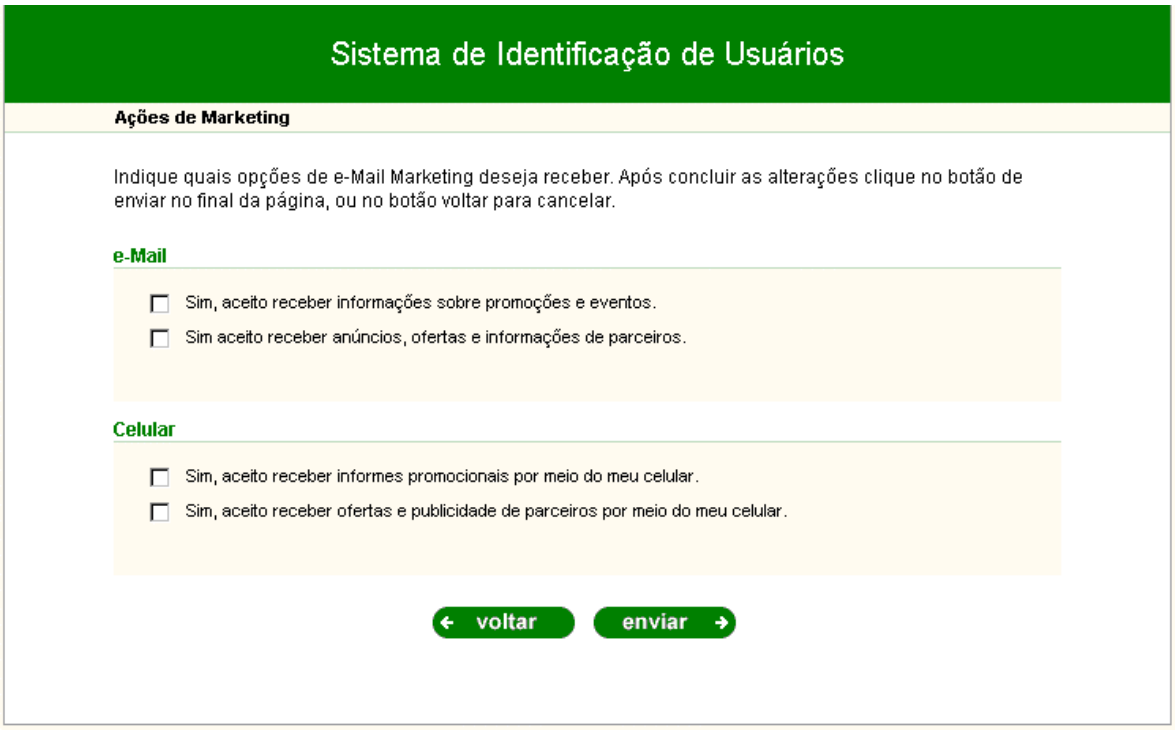

Figura 6-12 Estudo de Caso - Tela Ações de Marketing

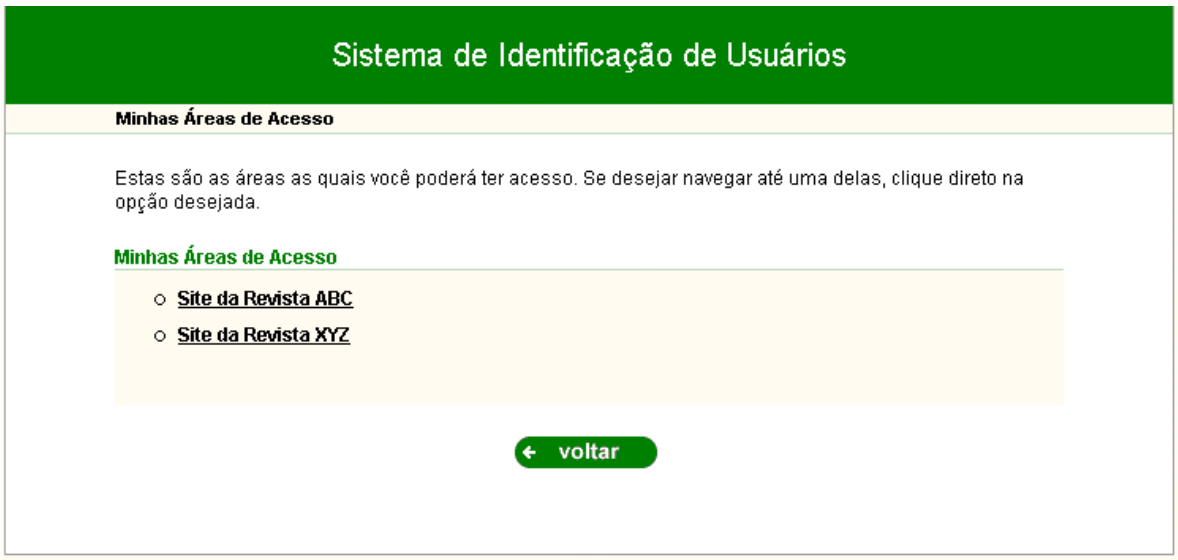

Figura 6-13 Estudo de Caso - Tela Minhas Áreas de Acesso

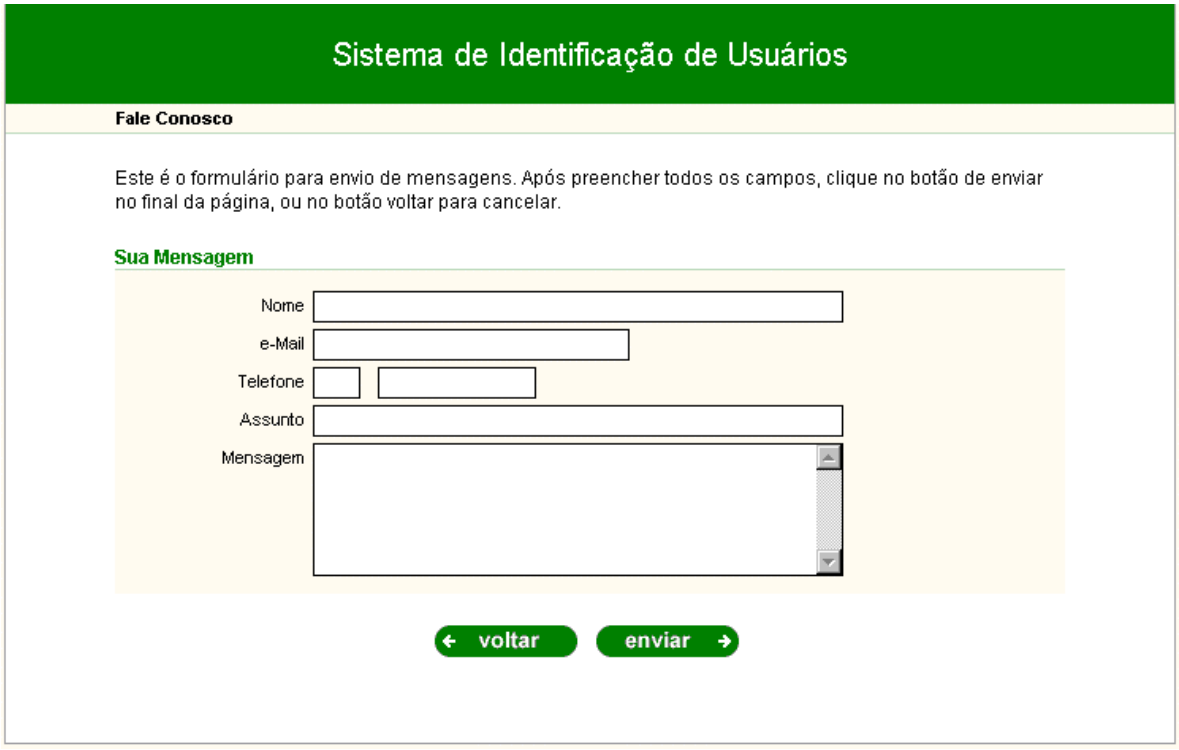

Figura 6-14 Estudo de Caso - Tela Fale Conosco

Apêndice B

Estudo de Caso - Modelo de Dados

O modelo de dados do sistema, representado através de Modelos de Entidade e Relacionamento apresentados nas Figuras B-1 à B-4, compõe-se dos grupos Cadastro, Newsletter e Marketing, Autenticação e Pesquisa, conforme detalhado na Seção 5.1.4. As tabelas em destaque pertencem a sistemas fora da fronteira da aplicação ou do escopo de contagem considerado.

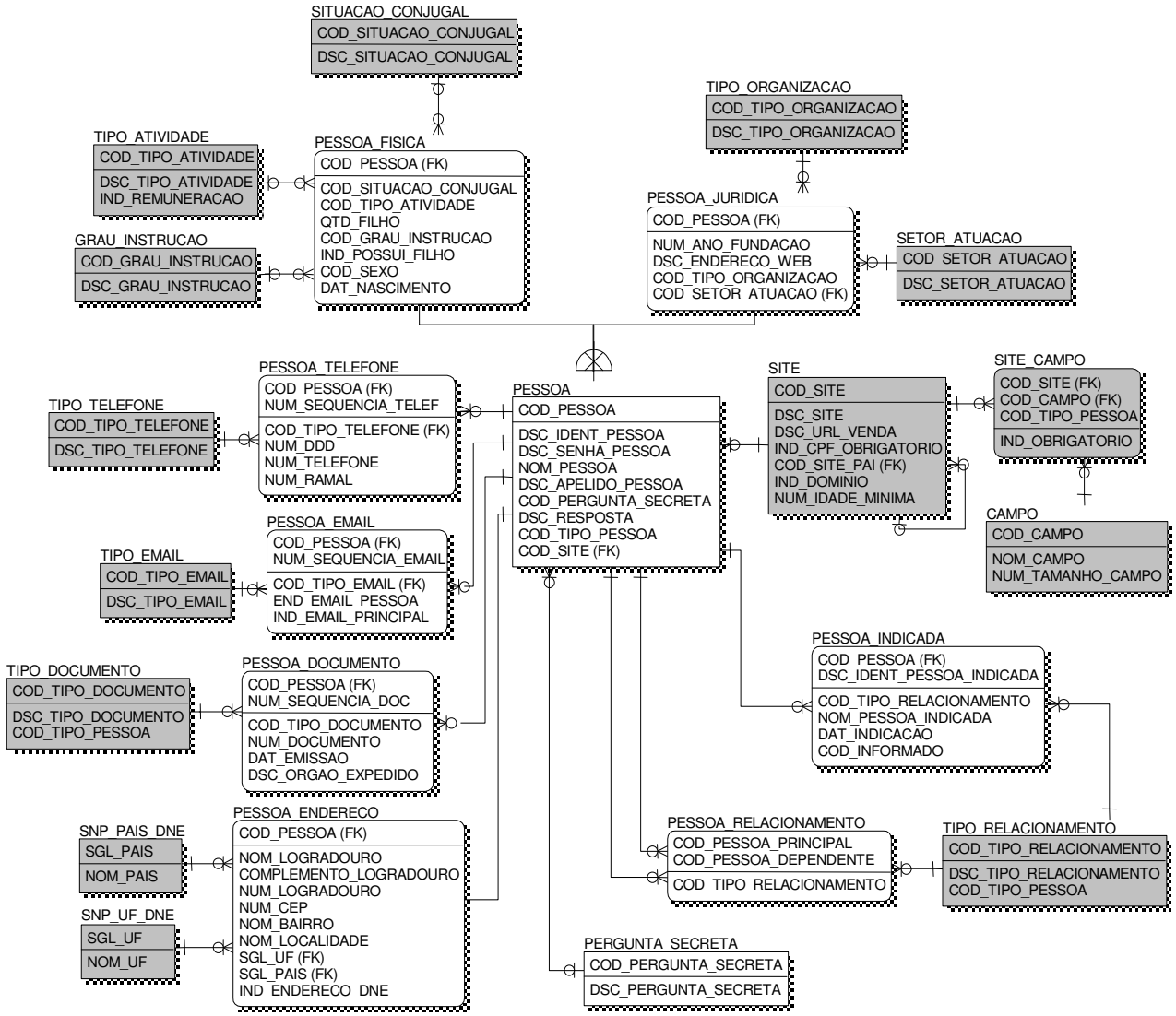

Figura 6-15 Estudo de Caso - Modelo de Dados - Cadastro

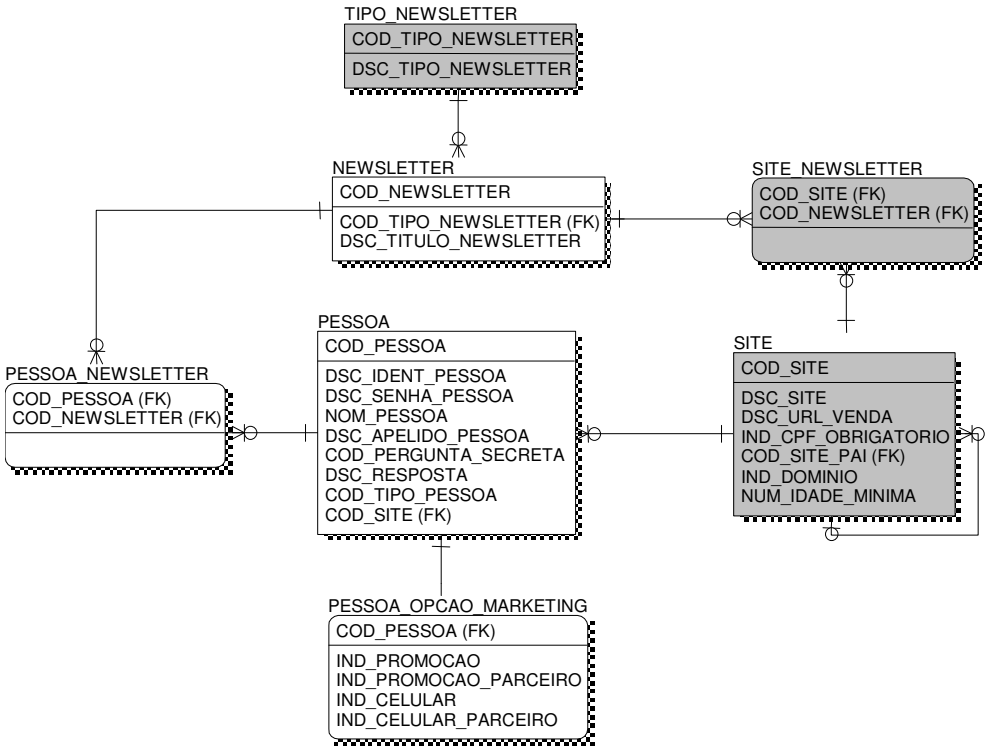

Figura 6-16 Estudo de Caso - Modelo de Dados - Newsletter e Marketing

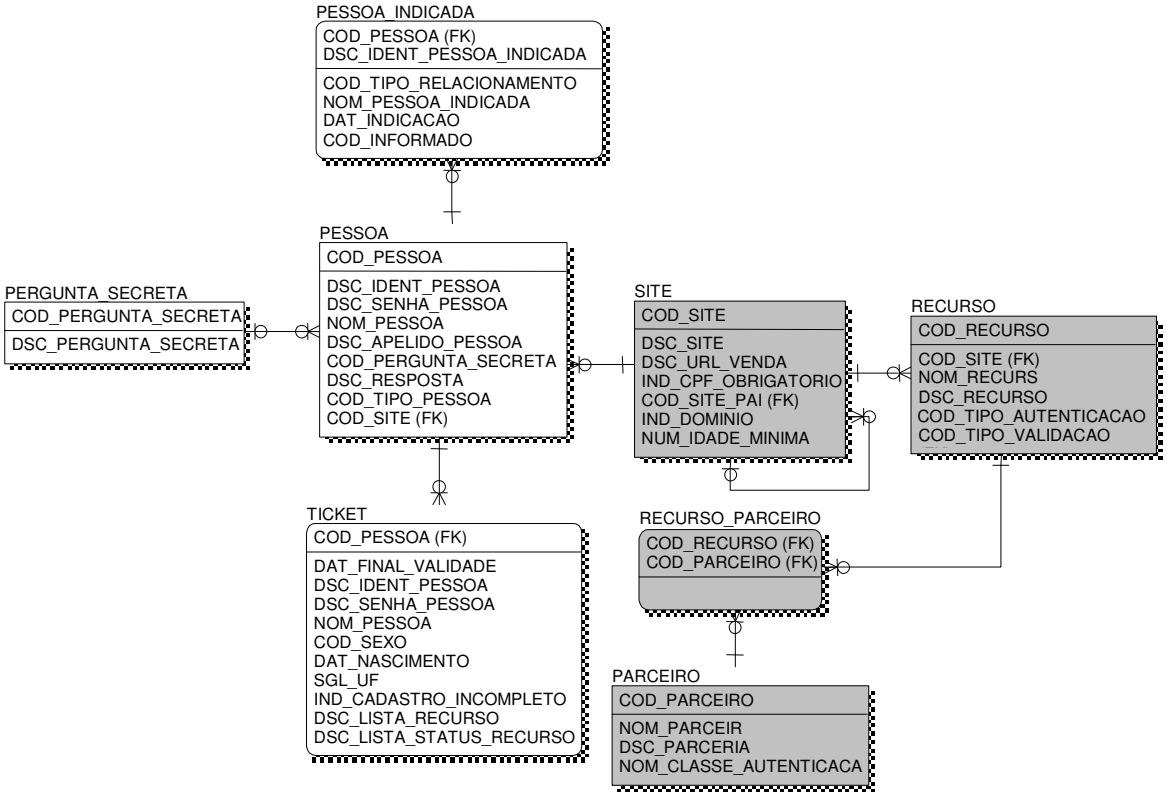

Figura 6-17 Estudo de Caso - Modelo de Dados - Autenticação

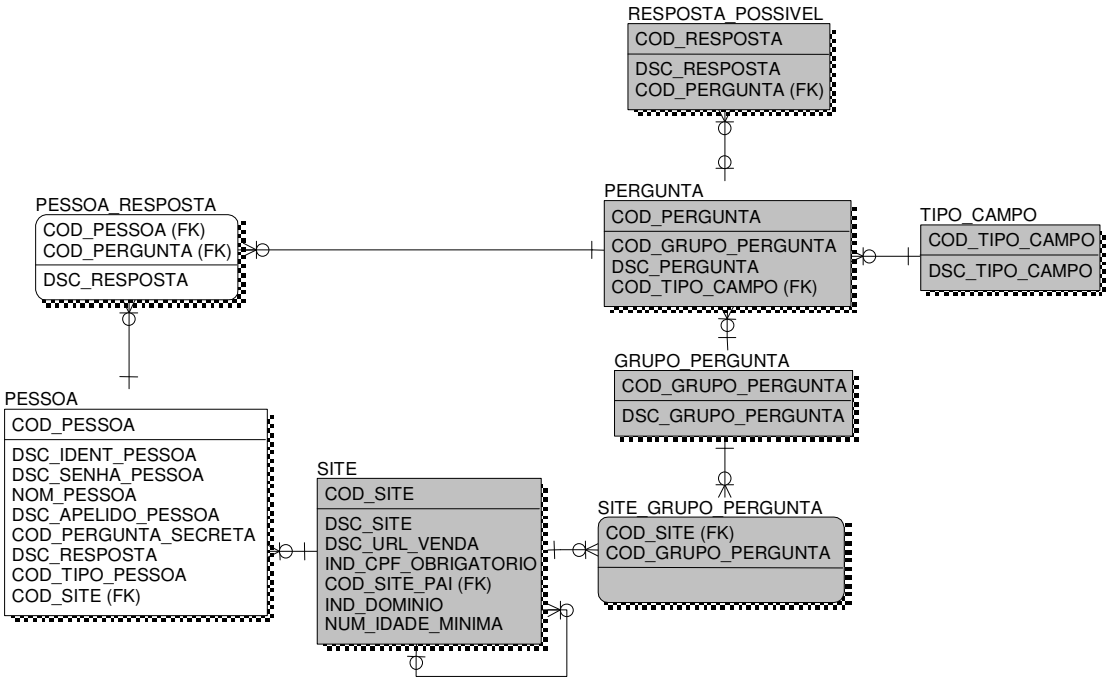

Figura 6-18 Estudo de Caso - Modelo de Dados - Pesquisa