

**UNIVERSIDADE REGIONAL DE BLUMENAU**  
**CENTRO DE CIÊNCIAS EXATAS E NATURAIS**  
**CURSO DE SISTEMAS DE INFORMAÇÃO – BACHARELADO**

**SISTEMA DE CÁLCULO DE CUSTO DE  
DESENVOLVIMENTO DE SOFTWARE UTILIZANDO FPA**

**ANA PAULA BERLANDA**

**BLUMENAU**  
**2005**

**2005/1-03**

**ANA PAULA BERLANDA**

**SISTEMA DE CÁLCULO DE CUSTO DE  
DESENVOLVIMENTO DE SOFTWARE UTILIZANDO FPA**

Trabalho de Conclusão de Curso submetido à  
Universidade Regional de Blumenau para a  
obtenção dos créditos na disciplina Trabalho  
de Conclusão de Curso II do Curso de  
Sistemas de Informação — Bacharelado.

Prof. Wilson Pedro Carli – Orientador

**BLUMENAU  
2005**

**2005/1-03**

**SISTEMA DE CÁLCULO DE CUSTO DE  
DESENVOLVIMENTO DE SOFTWARE UTILILIZANDO FPA**

Por

**ANA PAULA BERLANDA**

Trabalho aprovado para obtenção dos créditos  
na disciplina de Trabalho de Conclusão de  
Curso II, pela banca examinadora formada  
por:

Presidente:

---

Prof. Wilson Pedro Carli – Orientador, FURB

Membro:

---

Prof. Everaldo Artur Grahl, FURB

Membro:

---

Prof. Luiz Heinzen, FURB

Blumenau, 27 de julho de 2005.

Dedico este trabalho especialmente aos meus pais Adolar Berlanda e Elisete Vieira, por serem a razão do meu viver, por sempre estarem ao meu lado me incentivando e apoiando. Também a todas as pessoas que acreditaram em mim e sempre estiveram ao meu lado.

“Jamais considere seus estudos como uma obrigação, mas como uma oportunidade invejável para aprender a conhecer a influência libertadora da beleza do reino do espírito, para seu próprio prazer pessoal e para proveito da comunidade à qual seu futuro trabalho pertencer”.

Albert Einstein

## **AGRADECIMENTOS**

À Deus, pelo seu imenso amor e graça. Por me guiar e iluminar meus caminhos e minha vida.

À minha família, por acreditarem em mim e estarem sempre presentes.

Aos meus colegas e amigos pelo incentivo, em especial Fabio Cristiam Meisen, Alexandro Deschamps e Roseléia Martins pelo imenso coleguismo nestes anos que se passaram e pelos que estão por vir.

Aos mestres pelos ensinamentos.

E a todas as pessoas que direta ou indiretamente contribuíram para realização deste trabalho, meu muito obrigada.

## RESUMO

Este trabalho apresenta o desenvolvimento de um aplicativo para analisar e auxiliar na avaliação do custo para desenvolvimento de software utilizando métricas de Análise por Ponto de Função - versão 4.1 ( FPA -*Function Point Analysis*), que “estabelece o tamanho do software pela qualificação de sua funcionalidade externa baseada no projeto lógico ou a partir do modelo de dados”. Também será possível efetuar levantamento de custo para o desenvolvimento de software, através da utilização das variações do cálculo FPA: método de contagem indicativa e método das estimativas percentuais com o intuito de auxiliar nas estimativas iniciais de custos do desenvolvimento para as negociações com o cliente.

Palavras chaves: FPA; Método de Contagem Indicativa; Método de Estimativas Percentuais.

## **ABSTRACT**

This work describes the design of an application to analysis and help the evaluations of the software developing costs, utilizes metrics of analysis by Function Point Analysis - version 4.1 (FPA), who defines the size of a software by qualifications of its outside functionality based on the logic project or data model. Also is possible, evaluating a research of the software developing costs, utilising the FPA variables: indicant counts method and percent estimating method with intention to helps the initial costs estimates for the negotiations with the customers.

Key-Words: Function Point Analysis; Indicant Counts Method and Percent Estimating Method.

## LISTA DE ILUSTRAÇÕES

Figura 01 – Fluxograma do Procedimento de Contagem .....	23
Figura 02 – Exemplo de MEP .....	37
Figura 03 - Casos de Usos Efetuados pelo Analista.....	45
Figura 04 – Diagrama de Atividades .....	47
Figura 05 – Modelo Conceitual .....	49
Figura 06 – Modelo Físico .....	50
Figura 07 – Tela Principal do Sistema.....	60
Figura 08 – Tela de Cadastro de Linguagem.....	60
Figura 09 – Tela de Cadastro de Cliente .....	61
Figura 10 – Tela de Cadastro de Sistema .....	62
Figura 11 – Tela de Movimentação de Arquivos Lógicos Internos .....	63
Figura 12 – Tela de Movimentação de Arquivos de Interface Externa.....	63
Figura 13 – Tela de Movimentação de Consultas Externas .....	64
Figura 14 – Tela de movimentação de Entradas Externas .....	65
Figura 15 – Tela de Movimentação de Saídas Externas.....	65
Figura 16 – Tela de Movimentação de Fator de Ajuste .....	66
Figura 17 – Tela de Movimento de Contagem Indicativa .....	67
Figura 18 – Tela de Movimento de Estimativas Percentuais .....	67
Figura 19 – Tela do Filtro de Relatório de Resumo da Classificação das Funções .....	68
Figura 20 – Tela do Relatório de Resumo da Classificação das Funções .....	68
Figura 21 – Tela do Filtro de Relatório Fluxo de Orçamento .....	69
Figura 22 – Tela do Relatório Fluxo de Orçamento.....	69
Figura 23– Tela do Filtro Relatório Linguagens X Custo Sistema .....	70
Figura 24– Tela do Relatório Linguagens X Custo Sistema .....	71
Figura 25– Tela do Filtro do Relatório Linguagens X Custo Sistema .....	72
Figura 26– Tela do Filtro do Relatório Linguagens X Custo Sistema .....	72
Figura 27– Tela do Relatório Linguagens X Custo Sistema .....	73

## LISTA DE TABELAS

Tabela 01 – Contribuição / pontuação dos PF não ajustados das funções tipo dado. ....	24
Tabela 02 – Contribuição / pontuação dos PF não-ajustados das funções tipo transação.....	25
Tabela 03 – NI das características gerais da aplicação.....	26
Tabela 04 – Consulta externa .....	51
Tabela 05 – Arquivo de interface externa .....	51
Tabela 06 – Arquivo lógico interno.....	52
Tabela 07 – Entrada externa .....	52
Tabela 08 – Saída externa.....	52
Tabela 09 – Fator de ajuste.....	53
Tabela 10 – Contagem indicativa .....	53
Tabela 11 – Linguagem .....	53
Tabela 12 – Método das estimativas percentuais .....	54
Tabela 13 – Cliente.....	54
Tabela 14 – Sistemas .....	54
Tabela 15 –Exemplos de Linguagens x Níveis de PF. ....	38
Tabela 16 – Relacionamento entre Nível de Linguagem e Produtividade .....	39
Tabela 17 – Nível de Linguagem e Níveis de Produtividade em PF/Pessoa_hora .....	39

## LISTA DE QUADROS

Quadro 01 – Complexidade funcional dos ALI e AIE. ....	24
Quadro 02 – Complexidade Funcional para EE. ....	25
Quadro 03 – Complexidade Funcional para SE e CE. ....	25
Quadro 04 – Graus de Influência para Comunicação de Dados. ....	27
Quadro 05 – Graus de Influência para Processamento Distribuído. ....	27
Quadro 06 – Graus de Influência para Performance. ....	28
Quadro 07 – Graus de Influência para Utilização de Equipamento. ....	28
Quadro 08 – Volume de Transações ....	29
Quadro 09 – Entrada de Dados <i>On-Line</i> .....	29
Quadro 10 – Eficiência do Usuário Final ....	30
Quadro 11 – Atualização <i>On-Line</i> ....	30
Quadro 12 – Complexidade de Processamento ....	31
Quadro 13 – Reutilização de Código ....	31
Quadro 14 – Facilidade de Instalação / Implantação.....	32
Quadro 15 – Facilidade Operacional.....	33
Quadro 16 – Múltiplos Locais ....	33
Quadro 17 – Facilidade de Mudanças ....	34
Quadro 18 – Requisitos Funcionais ....	43
Quadro 19 – Requisitos Não Funcionais ....	44

## LISTA DE SIGLAS

AIE – Arquivos de Interfaces Externas  
ALI – Arquivos Lógicos Internos  
BFPUG – Brazilian Function Point Users Group  
CE – Consultas Externas  
CI – Contagem Indicativa  
EE – Entradas Externas  
ENUPF – Encontro Nacional de Usuários de Pontos de Função  
FA – Fator de Ajuste  
FPA – Análise por Ponto de Função  
IFPUG – *International Function Point Users Group*  
MEP – Método das Estimativas Percentuais  
MER – Modelo de Entidade e Relacionamento  
NI – Nível de Influência  
P.F. – Pontos por Função  
SE – Saídas Externas  
TD – Tipos de Dados  
TR – Tipos de Registros

## SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>14</b>
1.1 OBJETIVOS DO TRABALHO .....	15
1.2 ESTRUTURA DO TRABALHO .....	15
<b>2 FUNDAMENTAÇÃO TEÓRICA .....</b>	<b>17</b>
2.1 MÉTRICAS .....	17
2.2 ANÁLISE DE PONTOS DE FUNÇÃO .....	17
2.3 CONTAGEM DOS PONTOS DE FUNÇÃO .....	20
2.3.1 Arquivos Lógicos Internos (ALI).....	20
2.3.2 Arquivo de Interface Externa (AIE) .....	21
2.3.3 Entrada Externa (EE).....	21
2.3.4 Consulta Externa (CE).....	21
2.3.5 Saída Externa (SE).....	22
2.4 ETAPAS DO PROCESSO DE CONTAGEM .....	22
2.4.1 Pontos de Função Não Ajustados .....	23
2.4.2 Fator de Ajuste .....	25
2.4.3 Pontos de Função Ajustados.....	34
2.5 CONTAGEM INDICATIVA E MÉTODO DAS ESTIMATIVAS PERCENTUAIS .....	35
2.5.1 Método de Contagem Indicativa.....	35
2.5.2 Método das Estimativas Percentuais .....	36
2.6 LINGUAGENS DE PROGRAMAÇÃO.....	37
2.7 PREÇO DE UM PONTO DE FUNÇÃO .....	39
2.8 MODELO DE PROPOSTA CONTRATUAL .....	40
2.8.1 Tipos de Contrato .....	41
2.9 TRABALHOS CORRELATOS.....	41
<b>3 DESENVOLVIMENTO DO TRABALHO.....</b>	<b>43</b>
3.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO.....	43
3.2 ESPECIFICAÇÃO .....	44
3.2.1 Diagrama de casos de uso.....	44
3.2.2 Modelo de Dados.....	48
3.2.3 Dicionário de dados .....	51
3.3 IMPLEMENTAÇÃO .....	55
3.3.1 TÉCNICAS E FERRAMENTAS UTILIZADAS .....	55

<b>4 CONCLUSÕES</b> .....	<b>75</b>
4.1 EXTENSÕES .....	75
<b>REFERÊNCIAS BIBLIOGRÁFICAS</b> .....	<b>76</b>
APÊNDICE A – DESCRIÇÃO DOS CASO DE USO.....	78

# 1 INTRODUÇÃO

Atualmente, as organizações estão cada vez mais dependentes da tecnologia da informação, e a geração de produtos de software com qualidade e com custo compensador, se tornou um fator crítico de sucesso. Para alcançar níveis de qualidade adequados às exigências dos atuais clientes/usuários, os gerentes e engenheiros de software deveriam praticar a gerencia de projetos e produtos com base em fatos e dados (Fernandes Apud Valcanaia, 1998, p.01).

Conforme Silva (2000), todo desenvolvedor de software precisará estimar preço para quem solicitar o desenvolvimento de software. Esta pessoa ou empresa, certamente desejará saber o preço a ser cobrado pelo trabalho, antes mesmo que ele seja iniciado. Evidentemente que este preço poderá ser estimado com base em experiências anteriores. Também poderá avaliar com base nas informações que receber se o software será “pequeno” ou “grande” e cobrar um preço justo pelo serviço. Porém, independente do tamanho do software não nos diz muita coisa, e o usuário embora pagando o que foi cobrado, talvez não fique convencido de que fez um bom negócio. Utilizando pontos de função como sua unidade de medida padrão, é muito provável que o usuário consiga entender melhor o preço cobrado pelo seu produto.

Segundo Vazquez (2003), nas fases iniciais do projeto ainda não há o conhecimento completo das características do produto que permita a apuração de sua futura dimensão, neste caso é necessário estimar. Vários modelos de estimativas foram criados para fornecer métricas que permitam atender com menor margem de erro às necessidades de comunicação e informação do projeto. A análise por pontos de função permite não só medir o tamanho do sistema em termos da funcionalidade fornecida ao usuário, mas também estimar seu tamanho em qualquer fase do seu ciclo de vida (mesmo que os requisitos ainda não tenham sido detalhados).

Outra possibilidade é a de cadastrar os sistemas a serem desenvolvidos (com a finalidade de auxiliar o acompanhamento do que deve ser realizado), cadastrar a linguagem a ser utilizada para o desenvolvimento do software - com a finalidade de medir o custo por ponto de função (PF) - estimando o custo com a utilização da linguagem escolhida. No final, ou mesmo no decorrer do desenvolvimento do software, poderá ser emitido relatório de acompanhamento para saber o custo real por PF, podendo haver comparações do valor

estimado (através dos métodos de contagem indicativa e métodos das estimativas percentuais) e o valor de fato realizado.

Diante destes problemas, este trabalho tem por finalidade apresentar a importância de utilização da norma Análise por Ponto de Função (FPA *Function Point Analysis*), versão 4.1, para regular e orientar a atividade de produção de software. Segundo Fuck, (1995), com a utilização de métricas no processo de desenvolvimento de um software há um considerável aumento da qualidade nos produtos e na prestação de serviços e isto é uma das grandes prioridades das empresas hoje em dia.

## 1.1 OBJETIVOS DO TRABALHO

O objetivo geral deste trabalho foi desenvolver um sistema que auxilie pequenas e médias empresas desenvolvedoras de softwares a estimar custos para o desenvolvimento de software, utilizando a técnica FPA e algumas variações.

Os objetivos específicos do trabalho são:

- a) analisar e verificar o tempo que cada linguagem irá utilizar para o desenvolvimento de software;
- b) especificar o custo utilizando método de contagem indicativa e o método das estimativas percentuais;
- c) gerar relatório de fechamento da proposta comercial.

## 1.2 ESTRUTURA DO TRABALHO

No primeiro capítulo, o trabalho aborda a introdução do trabalho, os objetivos a serem alcançados no seu desenvolvimento e a estrutura do trabalho.

No segundo capítulo aborda-se a fundamentação teórica do presente trabalho. Descreve-se um breve histórico sobre a evolução das métricas, a importância da medição de software também para cálculo de custo. Apresenta-se uma visão geral sobre custo, contrato para o desenvolvimento de software e por fim trabalhos correlatos e diferenciais do trabalho.

O terceiro capítulo contempla o desenvolvimento do trabalho, tais como, a especificação ilustrando o Diagrama de Casos de Uso e o Modelo de Entidade Relacionamento – MER e Dicionário de Dados. Contém a implementação do sistema desenvolvido, descrevendo técnicas e ferramentas utilizadas.

Finalizando, o quarto capítulo descreve as considerações finais sobre o trabalho, incluindo também as sugestões para trabalhos futuros.

## 2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo faz-se uma explanação sobre os principais tópicos abordados neste trabalho, tais como métricas, análise por pontos de função, método de contagem indicativa, método das estimativas percentuais e modelo de proposta contratual.

### 2.1 MÉTRICAS

As métricas surgiram a partir da década de 70 e originaram-se da aplicação de cálculos para quantificar indicadores sobre o processo de desenvolvimento de software. O conceito de métrica, segundo Fernandez (1995), é o método de determinar, quantitativamente, a extensão em que o projeto, o processo e o produto de software têm certos atributos. Isto inclui a fórmula para determinar o valor da métrica como também sua forma de apresentação e as diretrizes de utilização e interpretação dos resultados obtidos no contexto do ambiente de desenvolvimento de software.

Uma métrica de software é qualquer tipo de medição que se refira a um sistema de software, processo ou documentação relacionada. As métricas podem ser de controle (geralmente associadas a processos de software) ou preditivas (geralmente associadas a produtos de software) (SOMMERVILLE, 2003).

Com a utilização de métricas no processo de desenvolvimento de um software há um considerável aumento da qualidade nos produtos e na prestação de serviços, sendo uma das grandes prioridades das empresas atualmente (Putman Apud Deimling, 1999, p.05).

### 2.2 ANÁLISE DE PONTOS DE FUNÇÃO

Segundo Vazquez (2003), a técnica da análise de pontos de função surgiu na IBM, no início da década de 70, como uma alternativa às métricas baseadas em linhas de código. Na época, Allan Albrecht foi encarregado de medir a produtividade de vários projetos de software desenvolvidos pela IBM. Esses projetos tinham sido desenvolvidos em uma grande variedade de linguagens, tornando inviável uma análise conjunta de produtividade. Foi isso que motivou a busca por uma medida que fosse independente da linguagem programação utilizada.

Depois que a técnica foi apresentada à comunidade no final da década de 70 e dos contínuos trabalhos realizados por Capers Jones destacando sua importância, houve um crescimento acelerado do número de usuários de pontos de função, culminando, em 1986, na fundação do *International Function Point Users Group* – IFPUG. Com esta aceitação crescente do mercado Allan Albrecht apresentou novas variações do ponto de função.

Em 1990, foi efetivamente lançada a primeira versão do Manual de Práticas de Contagem ou *Counting Practices Manual* (COM), que tinha como objetivo principal a padronização das técnicas e controle de sua evolução; assim, conforme Valcanaia (1998), em 1990 foi aprovada a versão 3.0, seguindo-se as versões 3.1, 3.2, 3.3, 3.4 em 1994 a versão 4.0, em 2002 a versão 4.1 (conforme Vazquez (2003), esta foi submetida a um *Public Available Specification* para a ISO, sendo posteriormente aprovada sob denominação ISO/IEC 20926:2002) a qual é abordada para desenvolvimento do trabalho e atualmente a versão 4.2 (que aborda a gerência do escopo de projetos de software, validação dos requisitos funcionais, análise de produtividade e qualidade, aplicação de estimativas, contratos de desenvolvimento de sistemas) . E desde então este é o padrão reconhecido pela indústria para pontos de função.

Conforme Valcanaia (1998), com a disseminação da técnica surgiram outras aplicações e verificou-se que a técnica de FPA poderia ser utilizada para medição da produtividade de desenvolvimento e a qualidade de sistemas através do acompanhamento de índices de erros por pontos de função, possibilitando a utilização do FPA como benefício gerenciais, aumentando desta forma o número de usuários.

Segundo BFPUG (1999), a medida funcional de tamanho de software é um conceito definido pelo padrão ISO/IEC 14143-1:1998 e refere-se à medição do tamanho do software considerando-se apenas a funcionalidade solicitada e recebida pelos respectivos usuários. Nesse sentido, uma medida funcional de tamanho é uma medida externa, pois mede somente aquilo que é percebido pelos usuários do produto de software, independentemente da forma de implementação escolhida.

Segundo Fernandez (1995), o objetivo da análise por ponto de função (FPA) é medir o tamanho de software pela quantificação de sua funcionalidade externa, baseada no projeto lógico ou a partir do modelo de dados. A contagem dos Pontos de Função deve:

- a) medir o que o cliente/usuário requisitou e recebeu;
- b) medir independentemente da tecnologia utilizada para a implantação;

- c) propiciar uma métrica de tamanho para apoiar análises de qualidade e produtividade;
- d) propiciar um veículo de estimativa de softwares;
- e) proporcionar um fator de normalização para comparação entre softwares.

Segundo Fernandez Apud Deimling, 1999, p.18, neste método o tamanho de um software é calculado por componentes os quais fornecem informações como o tamanho de processamento e a complexidade técnica de fatores de ajuste.

Conforme Hazan (2000), a análise de PF mede o software através da quantificação de sua funcionalidade fornecida ao usuário baseando-se no Projeto Lógico. Os objetivos da análise por ponto de função são:

- a) medir funcionalidade requisitada e recebida pelo usuário;
- b) medir projetos de desenvolvimento e manutenção de software, independentemente da tecnologia utilizada na implementação.

Ressalta ainda, que os principais benefícios da análise de PF nas organizações podem ser considerados como:

- a) uma ferramenta para determinar o tamanho do pacote de software adquirido através da contagem de todos os PF incluídos no pacote;
- b) uma ferramenta para apoiar a análise da qualidade e da produtividade;
- c) um mecanismo para estimar-se custos e recursos envolvidos em projetos de desenvolvimento e manutenção de software;
- d) um fator de normalização para comparação de software.

No artigo publicado por *Brazilian Function Point Users Group* (BFPUG, 1999), através da utilização dos pontos de função muitos orçamentos de manutenção são estabelecidos com base na performance de anos anteriores. Muitas organizações tentam reduzir os custos de manutenção e não planejam aumentar seus custos de manutenção de ano a ano, para nenhum aplicativo em particular.

Se as quantidades de novas funcionalidades forem determinadas e adicionadas ao produto básico, o custo unitário de manutenção poderá diminuir, enquanto o gasto real permanece constante ou aumenta. Se os custos de manutenção estiverem crescendo a uma

taxa inferior à taxa de crescimento dos pontos de função, então os custos de manutenção estarão caindo. Por exemplo, se uma organização gasta em média \$ 100.000 para manter 10.000 pontos de função (\$ 10 por ponto de função) e o número de pontos de função cresce 10 por cento e passa a ser 11.000, permanecendo constantes os gastos em manutenção, terá o custo unitário de manutenção caído para \$ 9 por ponto de função.

No Brasil, o uso da técnica da análise de P.F. começou significativamente no início da década de 90, com um forte apoio de uma grande empresa – Unisys. No período de 1991 a 1994, foram realizados seis Encontro Nacional de Usuários de Pontos de Função (ENUPF), inclusive com a participação de palestrantes internacionais. Porém o interesse do mercado consolidou-se apenas quando grandes contratos públicos começaram a ser baseados em pontos de função. Em paralelo, o interesse crescente das organizações envolvidas no desenvolvimento de software por modelos de qualidade e maturidade (ISO e CMM) contribuiu para o aumento de interesse no assunto. A criação do BFPUG foi também um importante marco para essa consolidação (VAZQUEZ, 2003).

## 2.3 CONTAGEM DOS PONTOS DE FUNÇÃO

Segundo BFPUG (1999), a contagem de Pontos de Função é efetuada com base na especificação do sistema, complementada por informações dos usuários e analistas sendo que os componentes considerados são: arquivos lógicos internos, arquivos de interface externa, entrada externa, consulta externa e saída externa.

### 2.3.1 Arquivos Lógicos Internos (ALI)

Correspondem aos arquivos mantidos e utilizados pelo sistema sendo contado. Uma tabela que insere, exclui e altera os dados é considerada um arquivo lógico interno. Segundo Vazques (2003), a tabela que guarda o cadastro de clientes é um exemplo de arquivo lógico interno. Para que este tipo de função seja contado como um arquivo lógico interno todas as regras relacionadas a seguir devem ser válidas:

- a) o grupo de dados ou informações de controle é logicamente relacionado e identificável pelo usuário;
- b) o grupo de dados deve ser mantido dentro da fronteira da aplicação que é contada.

### 2.3.2 Arquivo de Interface Externa (AIE)

Correspondem aos arquivos utilizados pelo sistema sendo contado, porém mantidos por outros sistemas. Exemplo: no cadastro de cliente é necessário fazer a checagem do financeiro do cliente.

Possui como principal finalidade armazenar dados referenciados por meio de um ou mais processos elementares dentro da fronteira da aplicação sendo contada, ou seja, obrigatoriamente deve ser um ALI de outra aplicação (VAZQUEZ, 2003).

Para que este tipo de função seja contado como AIE todas as regras relacionadas a seguir devem ser válidas:

- a) grupo de dados ou informações de controle é logicamente relacionado e identificável pelo usuário;
- b) grupo de dados é referenciado pela aplicação sendo contada, porém é externo a ela;
- c) grupo de dados não é mantido pela aplicação sendo contada;
- d) grupo de dados é mantido por outra aplicação.

### 2.3.3 Entrada Externa (EE)

Corresponde a transações cujo objetivo é a manutenção de arquivos ou a alteração do comportamento do sistema. Possui como principal função, manter um ou mais ALI e/ou modificar o comportamento do sistema. Exemplo: janela que permite adicionar, excluir e alterar registros em arquivos. Desta forma contribuem com três EE.

### 2.3.4 Consulta Externa (CE)

Corresponde a transações cujo objetivo é a apresentação de informações aos usuários, provenientes dos arquivos, sem a geração de dados derivados, atualização de arquivos ou a utilização de cálculos/fórmulas. Este é um processo elementar que envia dados ou informações de controle para fora da fronteira da aplicação, apresentando desta forma, informações ao usuário por meio de uma simples recuperação de dados ou informações de controle de um ALI ou AIE. Sendo que a lógica de processamento não deve conter fórmula matemática, cálculo ou dados derivados. Nenhum ALI é mantido durante seu processamento,

nem o comportamento do sistema é alterado. Exemplo: Tela de help, informações em formato gráfico.

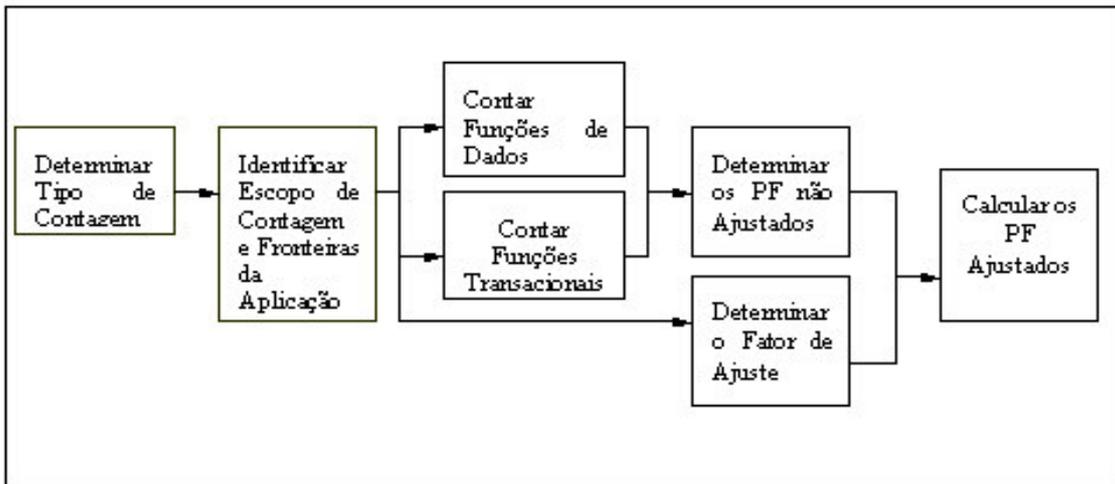
### 2.3.5 Saída Externa (SE)

Corresponde a transações cujo objetivo é a apresentação de informações aos usuários, não necessariamente provenientes de arquivos, podendo ocorrer a geração de dados derivados, atualização de arquivos e a utilização de cálculos/fórmulas. Tem como função apresentar informação ao usuário por meio de lógica de processamento que não seja apenas a recuperação de dados ou informações de controle. A lógica de processamento deve obrigatoriamente conter ao menos uma fórmula matemática ou cálculo, ou criar dados derivados. Também pode manter um ou mais ALI e/ou alterar o comportamento do sistema. Exemplo: relatórios com totalizações de dados.

Os ALI e AIE são funções do tipo dado que representam a funcionalidade fornecida ao usuário para atender suas respectivas necessidades de dados internos e externos a aplicação. A diferença entre eles é que o AIE não é mantido pela aplicação sendo contada. O termo arquivo refere-se ao grupo de dados logicamente relacionados e reconhecidos pelo usuário. E eventualmente um arquivo no sentido da APF (Análise por Ponto de Função) pode estar mapeado em um ou mais arquivos físicos do sistema operacional ou em tabelas do banco de dados. A forma como a aplicação implementa essas funcionalidades não é relevante para a determinação das funções do tipo dado. E são consideradas funções de transações as funções do tipo transação que representam a funcionalidade fornecida ao usuário para atender necessidades de processamento de dados pela aplicação sendo classificados em EE, SE ou CE (VAZQUEZ, 2003).

## 2.4 ETAPAS DO PROCESSO DE CONTAGEM

Primeiramente os componentes são identificados, conforme as regras de aplicações específicas para cada caso. Esta etapa é a que exige maior prática do analista de P.F. O fluxograma da figura 01 ilustra o procedimento de contagem de pontos de função.



Fonte: Vazquez, 2003.

Figura 01 – Fluxograma do Procedimento de Contagem

#### 2.4.1 Pontos de Função Não Ajustados

Segundo as normas do IFPUG, após a identificação, a cada um dos componentes é atribuída a complexidade simples, média ou complexa, conforme o número de campos no caso dos arquivos, ou o número de campos e referências a arquivos no caso das transações. Com base na complexidade e em tabelas específicas, a cada componente é atribuída uma quantidade de pontos de função, denominada contribuição do componente à contagem.

Cada ALI e AIE deve ser classificado com relação à sua complexidade funcional baseado em:

- a) número de tipos de dados (TD);
- b) número de tipos de registros (TR).

Um tipo de dado deve ser obrigatoriamente um campo único, e reconhecido pelo usuário não podendo ser repetido. Já um TR é um subgrupo de TD, também reconhecido pelo usuário, componente de um ALI ou AIE, sendo que o TR pode se dividido em dois subgrupos, os opcionais (que são aqueles em que o usuário tem a opção de não informar no processo elementar que cria ou adiciona dados ao arquivo), ou obrigatórios (onde o usuário requer que sejam sempre utilizados pelo processo primário que cria ou adiciona dados ao arquivo).

Uma vez determinadas as quantidades de tipos de dados e de tipos de registros, a classificação com relação à complexidade é fornecida da seguinte forma demonstrada no quadro 01:

TIPOS DE REGISTROS	TIPOS DE DADOS REFERENCIADOS			
		1 a 19	20 a 50	51 ou mais
	Apenas 1	Simple	Simple	Média
	2 a 5	Simple	Média	Complexa
	5 ou mais	Média	Complexa	Complexa

Fonte: Vazquez, 2003.

Quadro 01 – Complexidade funcional dos ALI e AIE.

Após a determinação da complexidade dos arquivos faz-se necessário o cálculo da pontuação ou contribuição conforme representada na tabela 01, seguinte:

Tipo de Função	Simple	Média	Complexa
ALI	7 PF	10 PF	15 PF
AIE	5 PF	7 PF	10 PF

Fonte: Vazquez, 2003.

Tabela 01 – Contribuição / pontuação dos PF não ajustados das funções tipo dado.

Lembrando que o tipo de dado deve ser um campo único e reconhecido pelo usuário, não podendo ser repetido. E para determinar o número de tipos de registros deve contar um tipo de registro para cada subgrupo, obrigatório ou opcional, caso não houver nenhum subgrupo, contar o próprio ALI ou AIE como um tipo de registro.

Segundo Vazquez (2003), cada EE, SE e CE deve ser classificada com relação à sua complexidade funcional baseada em número de arquivos referenciados e número de tipos de dados, lembrando que são funções do tipo transação.

No quadro 02 é ilustrada a forma de classificação da complexidade das EE:

ARQUIVOS REFERENCIADOS	TIPOS DE DADOS REFERENCIADOS			
		1 a 4	5 a 15	16 ou mais
	0 ou 1	Simple	Simple	Média
	2	Simple	Média	Complexa
	3 ou mais	Média	Complexa	Complexa

Fonte: Vazquez, 2003.

Quadro 02 – Complexidade Funcional para EE.

ARQUIVOS REFERENCIADOS	TIPOS DE DADOS REFERENCIADOS			
		1 a 5	6 a 19	20 ou mais
	0 ou 1	Simple	Simple	Média
	2 ou 3	Simple	Média	Complexa
	4 ou mais	Média	Complexa	Complexa

Fonte: Vazquez, 2003.

Quadro 03 – Complexidade Funcional para SE e CE.

Tipo de Função	Simple	Média	Complexa
EE	3 PF	4 PF	6 PF
SE	4 PF	5 PF	7 PF
CE	3 PF	4 PF	6 PF

Fonte: Vazquez, 2003.

Tabela 02 – Contribuição / pontuação dos PF não-ajustados das funções tipo transação.

#### 2.4.2 Fator de Ajuste

Os pontos de função ajustados, resultado final da contagem, são obtidos a partir da aplicação de 14 fatores, denominados Características Gerais dos Sistemas (CGS), os quais

alteram a contagem anterior em  $-35\%$  a  $+35\%$  do valor inicial não ajustado, sendo que cada ponto atribuído ao nível de influência afeta o resultado em  $1\%$ .

Segundo Vazquez (2003), enquanto as funções do tipo dado refletem um requisito específico de armazenamento e as funções do tipo transação um requisito específico de processamento, as características gerais refletem funções que podem afetar a aplicação de maneira geral. Cada uma dessas características possui um nível de influência sobre a aplicação que pode variar em um intervalo discreto de zero a cinco, conforme a tabela 03 explicativa:

0.	Nenhuma Influência
1.	Influência Mínima
2.	Influência Moderada
3.	Influência Média
4.	Influência Significativa
5.	Grande Influência

Fonte: Vazquez, 2003.

Tabela 03 – NI das características gerais da aplicação.

A determinação do nível de influência de uma característica geral é fornecida pela IFPUG. A seguir são apresentadas as diretrizes conforme IFPUG para a determinação do NI para cada característica geral de sistema. Caso nenhuma dessas orientações aplica-se ao sistema, uma avaliação deve ser realizada para determinar o NI mais próximo possível.

**Comunicação de Dados:** os dados e informações de controle usados na aplicação são envolvidos e recebidos através de recursos de comunicação de dados, ou seja, determinam em que nível a aplicação está se comunicando com o processador de forma direta. Sendo que o protocolo é um conjunto de convenções que permite a transferência ou intercâmbio de informações entre dois sistemas ou dispositivos. Terminais, estações de trabalho e micros são alguns exemplos. Determina o nível em que aplicação se comunica diretamente com o processador. A pontuação deve ser de acordo com as seguintes orientações ilustradas no quadro 04.

<b>Grau</b>	<b>Descrição – Pontuação</b>
0	Quando a aplicação é puramente batch ou uma estação de trabalho isolada
1	Aplicação batch, mas utiliza entrada de dados ou impressão remota.
2	Aplicação batch, mas possui entrada de dados e impressão batch.
3	Aplicação inclui entrada de dados on-line em um processamento batch ou de um sistema de consulta
4	Aplicação é mais que uma entrada on-line, mas suporta apenas um tipo de protocolo de comunicação.
5	Aplicação é mais que uma entrada on-line, mas suporta mais de um tipo de protocolo de comunicação.

Quadro 04 – Graus de Influência para Comunicação de Dados.

**Processamento Distribuído:** define o nível em que a aplicação irá transferir os dados entre seus componentes. Função ou dados distribuídos dentro da fronteira é um exemplo de características da aplicação. A pontuação deve ser realizada de acordo com as orientações ilustradas no quadro 05.

<b>Grau</b>	<b>Descrição – Pontuação</b>
0	Aplicação não participa da transferência de dados ou processamento de função entre os componentes do sistema
1	Aplicação prepara dados para processamento pelo usuário final em outro componente do sistema, como planilhas eletrônicas ou banco de dados.
2	Os dados são preparados para transferência, e são processados em outro componente do sistema (não para processamento pelo usuário final)
3	Processamento distribuído e transferência de dados são feitos on-line e em apenas uma direção
4	Processamento distribuído e transferência de dados são realizados on-line em ambas direções
5	O processamento de funções é executado dinamicamente no componente mais apropriado do sistema

Quadro 05 – Graus de Influência para Processamento Distribuído.

**Performance:** define o nível em considerações sobre tempo de resposta e taxa de transações que influenciam o desenvolvimento da aplicação. Sendo que os objetivos estabelecidos ou aprovados pelo usuário, em termos de tempo de resposta ou taxa de transações, influenciam (ou influenciaram) o projeto, desenvolvimento, instalação e suporte da aplicação. A pontuação deve ser realizada de acordo com as orientações ilustradas no quadro 06.

<b>Grau</b>	<b>Descrição – Pontuação</b>
0	Nenhum requerimento especial de performance foi requerido pelo usuário
1	Requisitos de performance e projeto foram estabelecidos e revistos, mas nenhuma ação em especial foi requerida
2	Tempo de resposta ou taxa de transações são críticos durante as horas de pico. Não é necessário nenhum projeto especial para utilização de CPU, sendo o limite para processamento o dia seguinte.
3	Tempo de resposta ou taxa de transações são críticos durante todas as horas de trabalho. Nenhuma determinação especial para a utilização do processador foi estabelecida. O limite de processamento é crítico
4	Adicionalmente, os requisitos estabelecidos pelo usuário, requerem tarefas de análise de desempenho na fase de análise e projeção da aplicação.
5	Adicionalmente, ferramentas de análise de desempenho foram usadas nas fases de projeto, desenvolvimento e/ou implementação para atingir os requerimentos de desempenho estabelecido pelo usuário.

Quadro 06 – Graus de Influência para Performance.

**Utilização de Equipamento:** define e tenta identificar o grau de utilização dos computadores em que será utilizado o sistema, que solicite considerações especiais de análise e desempenho do aplicativo. A avaliação deve ser realizada visando identificar a carga de trabalho onde a aplicação está sendo analisada. A pontuação deve ser realizada de acordo com as orientações ilustradas no quadro 07.

<b>Grau</b>	<b>Descrição – Pontuação</b>
0	Nenhuma restrição operacional explícita ou mesmo implícita foi solicitada.
1	Existem restrições operacionais leves. Não há esforço especial necessário ao atendimento dessas restrições.
2	Algumas considerações de sincronismo ou segurança são especificadas
3	Existem requisitos específicos de processador para uma parte específica da aplicação.
4	Restrições operacionais requerem cuidados especiais no processo dedicado ou utilização pesada do processador central
5	Adicionalmente, há limitações nos componentes distribuídos da aplicação.

Quadro 07 – Graus de Influência para Utilização de Equipamento.

**Volume de Transações:** define em que nível o alto volume de transações influencia o projeto, desenvolvimento, instalação e suporte da aplicação. A pontuação deve ser realizada de acordo com as orientações ilustradas no quadro 08.

<b>Grau</b>	<b>Descrição – Pontuação</b>
0	Não estão previstos períodos de picos de taxa de transações.
1	São previstos períodos de pico de processamento com frequência.
2	São previstos picos semanais
3	São previstos picos diários
4	Altas taxas de transação definidas pelo usuário nos requisitos ou os níveis de serviço acordados são altos o bastante para requererem tarefas de análise de performance na fase do projeto
5	Alem do descrito anteriormente, é necessário utilizar ferramentas de análise de desempenho nas fases de projeto, desenvolvimento e/ou implantação

Quadro 08 – Volume de Transações

**Entrada de Dados On-line:** define em que níveis são efetuadas entrada de dados na aplicação por meio de transações interativas. Esta característica quantifica a entrada de dados on-line provida pela aplicação. A pontuação deve ser realizada de acordo com as orientações ilustradas no quadro 09.

<b>Grau</b>	<b>Descrição – Pontuação</b>
0	Todas as transações são processadas em lote
1	De 1% a 7% das transações são entradas de dados on-line
2	De 8% a 15% das transações são entradas de dados on-line
3	De 16% a 23% das transações são entradas de dados on-line
4	De 24% a 30% das transações são entradas de dados on-line
5	Mais de 30% das transações são entradas de dados on-line

Quadro 09 – Entrada de Dados *On-Line*

**Eficiência do Usuário Final:** define em que nível os fatores humanos e facilidades de uso pelo usuário final interferem no desenvolvimento da aplicação. As funções interativas fornecidas pela aplicação enfatizam um projeto para o aumento da eficiência do usuário final. Segundo Vazquez (2003), o projeto inclui:

- a) auxílio para navegação;
- b) menus;
- c) ajuda *on-line* e documentação;
- d) movimentação automática de cursor;
- e) paginação;
- f) impressão remota por meio de transações on-line;
- g) teclas de função predefinidas;
- h) tarefas em lote submetidas a transações on-line;
- i) seleção feita por posicionamento de cursor em tela de dados;
- j) uso intenso de vídeo reverso, brilho, cores e outros indicadores;

- k) documentação impressa das transações;
- l) interface de mouse;
- m) janelas *pop-up*;
- n) utilização de número mínimo de telas para executar uma função do negócio;
- o) suporte a dois idiomas (conte como quatro itens);
- p) suporte a mais de dois idiomas (conte como seis itens).

Realiza-se a pontuação de acordo com as orientações ilustradas no quadro 10.

Grau	Descrição – Pontuação
0	Nenhum dos itens mencionados anteriores
1	De 1 à 3 dos itens anteriores
2	De 4 à 5 dos itens anteriores
3	Seis ou mais dos itens anteriores, não existem requisitos específicos do usuário associado à eficiência.
4	Seis ou mais dos itens anteriores e requisitos explícitos sobre a eficiência para o usuário final são fortes e necessitam tarefas de projeto que incluam fatores humanos.
5	Seis ou mais dos itens anteriores e requisitos explícitos sobre a eficiência para o usuário final são fortes o bastante para necessitarem do uso de ferramentas e processos especiais para demonstrar que os objetivos foram alcançados.

Quadro 10 – Eficiência do Usuário Final

**Atualização On-line:** define o nível que a aplicação provê atualização on-line dos ALI. A pontuação deve ser realizada de acordo com as orientações ilustradas no quadro 11.

Grau	Descrição – Pontuação
0	Não há atualização on-line
1	Atualização on-line de 1 a 3 arquivos. O volume de atualização baixo e recuperação dos dados é fácil.
2	Atualização on-line de 4 ou mais arquivos. O volume de atualização é baixo e recuperação dos dados é fácil.
3	Atualização da grande maioria dos arquivos é <i>on-line</i>
4	Adicionalmente, a proteção contra perda de dados é essencial e foi especialmente projetada e programada no sistema.
5	Adicionalmente, o alto volume de processamento torna necessária a análise do custo do processo de recuperação. São incluídos procedimentos altamente automatizados com um mínimo de intervenção do operador.

Quadro 11 – Atualização *On-Line*

**Complexidade de Processamento:** define-se em que nível o processamento lógico ou matemático influencia o desenvolvimento da aplicação, sendo que a complexidade de processamento é uma das características da aplicação, os seguintes componentes estão presentes:

- a) controle sensível e/ou processamento de segurança da aplicação;
- b) processamento lógico extensivo;
- c) processamento matemático extensivo;
- d) processamento gerando muitas exceções, resultando em transações incompletas que devem ser processadas novamente;
- e) processamento complexo para manipular múltiplas possibilidades de entrada e saída.

A pontuação deve ser realizada de acordo com as orientações ilustradas no quadro 12.

<b>Grau</b>	<b>Descrição – Pontuação</b>
0	Nenhum dos itens descritos anteriormente
1	Qualquer um dos itens descritos anteriores
2	Quaisquer dois itens anteriores
3	Quaisquer três itens anteriores
4	Quaisquer quatro itens anteriores
5	Todos os itens anteriores

Quadro 12 – Complexidade de Processamento

**Reutilização de Código:** a aplicação e seu código serão projetados, desenvolvidos e suportados para serem utilizados em outras aplicações. A pontuação deve ser realizada de acordo com as orientações ilustradas no quadro 13.

<b>Grau</b>	<b>Descrição – Pontuação</b>
0	Nenhuma preocupação com reutilização de código
1	Código reutilizável é utilizado na aplicação
2	Menos de 10% da aplicação levou em consideração as necessidades de mais de um usuário
3	10% ou mais da aplicação levou em consideração as necessidades de mais de um usuário
4	A aplicação foi especificamente empacotada e/ou documentada para fácil reutilização. Ela é customizada pelo usuário em nível de código
5	Aplicação foi especificada, projetada e documentada para fácil reutilização. Ela é customizada pelo usuário por meio de manutenção e parâmetros

Quadro 13 – Reutilização de Código

**Facilidade de instalação / Implantação:** define-se em que nível a conversão de ambientes preexistentes influencia o desenvolvimento da aplicação. Um plano e/ou ferramentas de conversão e instalação foram fornecidos e testados durante a fase de teste do sistema. A pontuação deve ser realizada conforme orientações ilustradas no quadro 14.

Grau	Descrição – Pontuação
0	O usuário não definiu considerações especiais, não precisando nenhum <i>setup</i> para a instalação.
1	O usuário não definiu considerações especiais, porém necessita <i>setup</i> para instalação.
2	Requisitos de instalação e conversão foram definidos pelo usuário, e guias de conversão e instalação foram fornecidas e testadas. O impacto de conversão não é considerado importante
3	Requisitos de instalação e conversão foram definidos pelo usuário, guias de conversão e instalação foram fornecidas e devidamente testadas
4	Além do item 2, ferramentas de instalação e conversão automáticas foram fornecidas e testadas
5	Além do item 3. ferramentas de instalação e conversão automáticas foram fornecidas e testadas

Quadro 14 – Facilidade de Instalação / Implantação

**Facilidade Operacional:** facilidade operacional é uma das características da aplicação, atende-se alguns aspectos operacionais como procedimentos de inicialização, segurança e recuperação. A aplicação minimiza a necessidade de atividades manuais, como manipulação de papel e intervenção manual pelo operador.

A pontuação deve ser realizada de acordo com as orientações ilustradas no quadro 15.

Grau	Descrição – Pontuação
0	Usuário estabeleceu apenas os procedimentos de segurança normais
1-4	Um, alguns ou todos os seguintes itens são válidos para a aplicação. Selecione aqueles que sejam válidos. Cada item tem um valor de um ponto, a exceção de onde seja citado o contrário. <ul style="list-style-type: none"> <li>• Procedimentos de inicialização, salvamento e recuperação foram fornecidos. É necessária intervenção do operador (conte como dois itens);</li> <li>• Procedimentos de inicialização, salvamento e recuperação foram fornecidos. Não é necessária intervenção do operador (contar como dois itens);</li> <li>• Aplicação minimiza a necessidade de montagem de fitas;</li> <li>• Aplicação minimiza a necessidade de manipulação de papel.</li> </ul>
5	Aplicação foi projetada para trabalhar sem operador, nenhuma intervenção do operador é necessária para operar o sistema além de executar e encerrar a aplicação; possui rotinas automáticas para recuperação em caso de erros.

Quadro 15 – Facilidade Operacional

**Múltiplos Locais:** a aplicação foi projetada, desenvolvida e suportada especialmente para ser instalada em múltiplos locais ou para múltiplas organizações. A pontuação deve ser realizada de acordo com as orientações ilustradas no quadro 16.

Grau	Descrição – Pontuação
0	Os requerimentos do usuário não consideraram a necessidade de instalação em mais de um local.
1	Necessidade de múltiplos locais foi considerada no projeto, e a aplicação foi projetada para trabalhar apenas sobre o mesmo ambiente de software e hardware.
2	Necessidade de múltiplos locais foi considerada no projeto, e a aplicação foi projetada para operar em apenas ambientes de hardware e de software similares.
3	Necessidade de múltiplos locais foi considerada no projeto, e a aplicação foi projetada para operar em ambientes diferentes de hardware e de software.
4	Adicionalmente aos itens 1 ou 2, plano de suporte e documentação são fornecidos e testados para suportar a aplicação em múltiplos locais.
5	Adicionalmente ao item 3, plano de suporte e documentação são fornecidos e testados para suportar a aplicação em múltiplos locais.

Quadro 16 – Múltiplos Locais

**Facilidade de Mudanças:** descreve-se em que nível a aplicação foi projetada e desenvolvida visando facilitar sua manutenção. Sendo que as seguintes características podem ser válidas para a aplicação:

- a) estão disponíveis mecanismos de consulta flexível, que permitem a manipulação de pedidos simples;
- b) estão disponíveis mecanismos de consulta flexível, que permitem a manipulação de pedidos de media complexidade (contar como dois itens);
- c) estão disponíveis mecanismos de consulta flexível, que permitem a manipulação de pedidos complexos (contar como três itens);
- d) dados de controle do negócio são mantidos pelo usuário por meio de processos interativos, mas as alterações só tem efeito no próximo dia útil;
- e) dados de controle de negocio são mantidos pelo usuário por meio de processos interativos, e as alterações tem efeito imediato (contar como dois itens).

A pontuação deve ser realizada de acordo com as orientações ilustradas no quadro 17.

<b>Grau</b>	<b>Descrição – Pontuação</b>
0	Nenhum dos itens descritos anteriormente
1	Qualquer um dos itens anteriores
2	Quaisquer 2 itens anteriores
3	Quaisquer 3 itens anteriores
4	Quaisquer 4 itens anteriores
5	Todos os itens anteriores

Quadro 17 – Facilidade de Mudanças

### 2.4.3 Pontos de Função Ajustados

Assim como o último passo da contagem de PF, envolve-se o cálculo final para os três tipos de contagem: projeto de desenvolvimento, projeto de melhoria e projeto de aplicação. Neste caso, como enfoque do trabalho, será apresentado o tipo de contagem para projeto de desenvolvimento.

**Projeto de Desenvolvimento:** os componentes considerados para o cálculo do número de PF de um projeto de desenvolvimento são:

- **Funcionalidade de aplicação requisitada pelo usuário para o projeto:** funções utilizadas após a instalação do software para satisfazer as necessidades correntes do negócio do usuário.
- **Funcionalidade de conversão requisitada pelo usuário para o projeto:** funções disponíveis na instalação da aplicação para converter dados ou fornecer requisitos de

conversão especificados pelo usuário como relatório de verificação de conversão. Depois da instalação estas funções são desconsideradas.

- **Valor do FA da aplicação:** reflete a complexidade da aplicação é determinado pela ponderação das 14 características gerais de sistemas.

**Fórmula:**

$$\boxed{\mathbf{DFP = (UFP + CFP) * VAF}}$$

Onde:

- ⇒ **DFP:** Corresponde ao número de PF do projeto de desenvolvimento;
- ⇒ **UFP:** Número de PF não ajustados das funções disponíveis após instalação;
- ⇒ **CFP:** Número de PF não-ajustados das funções de conversão;
- ⇒ **VAF:** Valor do Fator de Ajuste.

## 2.5 CONTAGEM INDICATIVA E MÉTODO DAS ESTIMATIVAS PERCENTUAIS

Conforme Hazan (2000), umas das primeiras fases do processo de desenvolvimento de software é o levantamento e análise dos requisitos do sistema. Nesta fase, o desenvolvedor ainda não possui conhecimento suficiente de todo escopo e funcionalidades da aplicação a ser desenvolvida, fazendo com que os gerentes de software realizem uma estimativa inicial do custo de desenvolvimento para as negociações com o cliente. Dados quantitativos tais como tamanho da aplicação são fundamentais neste momento para auxiliar na precisão das estimativas.

Sendo assim, dados iniciais são utilizados para chegar a um cálculo de estimativa de todo projeto. Para chegar nestas estimativas, é abordado a seguir duas variações de FPA para calculo do custo final para desenvolvimento de software, que é o método de contagem indicativa e método das estimativas de métodos percentuais.

### 2.5.1 Método de Contagem Indicativa

Segundo Hazan (2000), este método foi criado pela associação de métricas da Holanda (*Netherlands Software Metrics Association* - NESMA), com a finalidade de obter o tamanho da aplicação, baseando-se no número de Arquivos Lógicos Internos (ALIs) e Arquivos de Interface Externa (AIEs) da aplicação. Para isto, é necessário um modelo de dados, mesmo

que inicial, para facilitar a visualização das funções de dados. A fórmula utilizada para o cálculo dos Pontos por Função é a seguinte:

$$PF = (35 * \text{Quantidade de ALIs}) + (15 * \text{Quantidade de AIEs})$$

Por exemplo, suponha uma aplicação com 7 arquivos lógicos internos. Aplicando-se a fórmula  $(35 * 7 + 0 * 15)$ , obtém-se o resultado de 245 Pontos por Função.

Vazquez (2003), menciona que na contagem indicativa, é necessária apenas a identificação dos Arquivos de Interface Externa e Arquivos Lógicos Internos. Considerando-se 35 PF para cada ALI e 15 PF para cada AIE identificado. Já os números 35 e 15 representam as médias de pontos de função identificadas para cada um dos tipos de arquivos, considerando projetos que, em média, possuam três entradas externas, duas saídas e uma consulta externa para cada arquivos lógicos internos e uma saída externa e uma consulta externa para cada arquivo de interface externa.

### 2.5.2 Método das Estimativas Percentuais

Conforme Hazan (2000), a empresa de SPR (*Software Productivity Research*), referência mundial na área de PF, a partir de dados históricos de projetos verificou a existência de uma correlação entre vários tipos de função de uma aplicação que expressa da seguinte forma:

- a) a quantidade de ALI representa 25% do total de funções de uma aplicação;
- b) a quantidade de AIE representa 3% do total de funções de uma aplicação;
- c) a quantidade de EE representa 30% do total de funções de uma aplicação;
- d) a quantidade de SE representa 28% do total de funções de uma aplicação;
- e) a quantidade de CE representa 14% do total de funções de uma aplicação.

Com base nesta análise estatística, identifica-se a quantidade de um dos tipos de função da aplicação para derivar a quantidade dos outros. A quantidade de um tipo de função deve ser representada sempre em valores inteiros. Para calcular os Pontos por Função, deve-se sempre trabalhar com uma complexidade funcional **média** para todos os tipos de função.

Considerando o valor do Fator de Ajuste igual a 1, abaixo temos um exemplo de cálculo e respectivo resultado ilustrado na figura 02.

Suponha a aplicação do exemplo anterior com 7 ALI. Aplicando o método descrito acima, pode-se concluir que:

- se 7 ARQUIVOS = 25% do total de funções, então **total de funções = 28.**
- se INTERFACES = 3% do total de funções, então INTERFACES = 1
- se ENTRADAS = 30% do total de funções, então ENTRADAS = 8
- se SAÍDAS = 28% do total de funções, então SAÍDAS = 8
- se CONSULTAS = 14% do total de funções, então CONSULTAS = 4

A partir destes valores pode-se calcular os PF Não-Ajustados da aplicação.

TIPO DE FUNÇÃO	COMPLEXIDADE FUNCIONAL	TOTAL TIPO FUNÇÃO
ARQUIVO	7 MÉDIA * 10	70
INTERFACE	1 MÉDIA * 7	7
ENTRADA	8 MÉDIA * 4	32
SAÍDA	8 MÉDIA * 5	40
CONSULTA	4 MÉDIA * 4	16
TOTAL DE PF's NÃO-AJUSTADOS =		165

Figura 02 – Exemplo de MEP

A precisão de estimativa do tamanho de uma aplicação varia de acordo com o grau de conhecimento adquirido sobre a mesma. É relevante destacar que a contagem dos PF's nas fases iniciais do ciclo de vida do software são estimativas da Contagem de PF final, realizada quando o escopo da aplicação torna-se mais claro e as funções são desenvolvidas. É freqüente identificar funcionalidades adicionais que não foram especificadas com os requisitos originais. É fundamental atualizar-se a contagem da aplicação em cada fase do ciclo de vida e quando o projeto estiver completo. Como a funcionalidade é modificada durante o desenvolvimento, a contagem de PF ao final do ciclo de vida reflete precisamente todas as funcionalidades entregues ao usuário.

## 2.6 LINGUAGENS DE PROGRAMAÇÃO

Hazan (2000) propõe que, quanto mais as linguagens de programação evoluem, menor é o esforço para produção de um PF por hora. Classificando os tipos de linguagens de programação em níveis numéricos, possibilita estimar a produtividade e também o esforço. A tabela 15 apresenta exemplos de linguagens e seus respectivos Níveis.

Linguagem	Nível
Assembly	1.00
C++	6.00
Clipper	17.00
Cobol	3.00
Delphi	11.00
HTML 3.0	22.00
Java	6.00
Oracle	8.00
Pascal	3.50
Visual Basic 5.0	11.00

Fonte: Hazan, 2000.

Tabela 15 –Exemplos de Linguagens x Níveis de PF.

Hazan (2000), afirma que o nível de uma linguagem e sua produtividade não é linear. Taxas precisas de produtividade de programação são examinadas com base em média mensal das taxas de produção de PF associadas com níveis das linguagens. A tabela 16 demonstra como os níveis de linguagem afetam a produtividade. Já a tabela 17 apresenta a produtividade em PF por hora, sendo que a primeira coluna desta tabela é associada ao nível de linguagem apresentado na tabela 15. Por exemplo, imagine um sistema de complexidade média, desenvolvido em Visual Basic 5.0, cujo nível é 11.00 (tabela 15), pela tabela 17 obtém-se a produtividade 0,1477 PF/pessoa\_hora, com uma equipe de três pessoas e considerando 1 dia com 6 horas, tem-se uma produtividade diária de 2,7 PF/equipe\_dia.

<b>Nível da Linguagem</b>	<b>Produtividade Média por Pessoa-mês</b>
1 – 3	5 a 10 Pontos por Função
4 – 8	10 a 20 Pontos por Função
9 – 15	16 a 23 Pontos por Função
16 – 23	15 a 30 Pontos por Função
24 – 55	30 a 50 Pontos por Função
Acima de 55	40 a 100 Pontos por Função

Tabela 16 – Relacionamento entre Nível de Linguagem e Produtividade

<b>Nível de Linguagem (Limite)</b>	<b>Produtividade Mínimo (PF/h)</b>	<b>Produtividade Média (PF/h)</b>	<b>Produtividade Máxima (PF/h)</b>
3.5	0.0379	0.0568	0.0758
8.5	0.0758	0.1136	0.1515
15.5	0.1212	0.1477	0.1742
23.5	0.1136	0.1704	0.2273
54.5	0.2273	0.3030	0.3788
99.99	0.3030	0.5303	0.7576

Tabela 17 – Nível de Linguagem e Níveis de Produtividade em PF/Pessoa\_hora

## 2.7 PREÇO DE UM PONTO DE FUNÇÃO

Para Vazquez (2003), o preço de um ponto de função pode variar de acordo com o trabalho requerido para a construção de um ponto de função e dos subprodutos a serem também entregues. Desta forma, não existe um preço único para ponto de função. Há necessidade de avaliar o conjunto de atividades referentes as funcionalidades que serão disponibilizadas em cada ponto de função, sendo que o modelo de contrato ditará a remuneração de um ponto de função bem como os aspectos não-funcionais que são desconsiderados na medição dos pontos de função.

De um modo geral, o preço de um PF poderá variar de acordo com as especificações solicitadas por cada trabalho, levando em consideração o tempo requerido para construção de cada PF e possíveis subprodutos também a serem entregues.

Segundo Sommerville (2003), para estimar o preço do software é necessário levar em consideração mais amplas sobre questões organizacionais, econômicas, políticas e de negócio, portanto, pode não haver uma relação simples entre o preço para o cliente do software e os custos de desenvolvimento. Esta é a razão das considerações organizacionais envolvidas. A estimativa de preço geralmente envolve a gerência sênior da organização, além dos gerentes de projeto de software.

## 2.8 MODELO DE PROPOSTA CONTRATUAL

O modelo de proposta contratual (fechamento de negócio) é disponibilizado como “instrumento” mediador aceito de comum acordo tanto pelo cliente quanto pelo fornecedor. Também serve como forma de reduzir os riscos em contratos cuja medição é realizada com base em pontos de função. Tem como objetivo, avaliar e auditar as contagens realizadas pela empresa contratada e resolver situações de possíveis conflitos.

Segundo Vazquez (2003), essa visão de negócio pode assumir as mais diversas formas. Recomenda-se que a organização contratante estabeleça sua perspectiva sobre os casos em que possa haver múltiplas interpretações do manual de práticas de contagem. O autor também relata outra sugestão no modelo de contrato como, por exemplo, o valor do fator de ajuste, caso ele seja utilizado. Em geral, quando a organização decide contratar um projeto, já existem informações suficientes para a determinação do Valor do Fator de Ajuste. Portanto, inserir essa informação de forma explícita no documento de solicitação de proposta ajudará o fornecedor a definir melhor seu preço e eliminará uma possível fonte de divergências.

Segundo o artigo A UTILIDADE DOS PONTOS DE FUNÇÃO (BFPUG, 2004), os pontos de função podem ser utilizados para ajudar a especificar os principais produtos a serem recebidos de um fornecedor, para assegurar a entrega dos níveis adequados de funcionalidade e para desenvolver medidas objetivas de eficácia em relação ao custo e qualidade. São mais eficazmente utilizados em contratos de preço fixo como forma de especificar exatamente o que deve ser entregue. Provê o melhor método objetivo para a avaliação do tamanho de um projeto de software e para o gerenciamento desse tamanho durante o desenvolvimento. Desta forma, o cliente também pode aceitar mais facilmente o risco para um dado tamanho de projeto de software (em pontos de função), e o desenvolvedor pode aceitar mais facilmente o risco para o custo de produção (custo por ponto de função).

A proposta comercial deverá conter a descrição dos elementos básicos que irão compor o sistema solicitado, como por exemplo, o preço, condições de pagamento, data de entrega e pagamento bem como o que foi definido brevemente na reunião.

### 2.8.1 Tipos de Contrato

Segundo Vazquez (2003), há dois tipos de contratação: Preço Fixo e Homem-Hora. O primeiro modelo consiste em um preço total fixo por um produto bem definido. Nas fases iniciais da concepção ou mesmo após a conclusão da especificação de requisitos, ainda existem características ocultas somente percebidas posteriormente. Dessa forma, a probabilidade de haver um aumento no escopo originalmente previsto é alta. Como consequência o contratante pode não receber o produto desejado ou o que foi contratado, tendo que arcar com custos adicionais que não são de sua direta responsabilidade. Muitas vezes sacrifica outros aspectos de maior dificuldade de medição.

No modelo Homem-Hora o risco é outro. Hora não é um bem ou serviço prestado, e sim uma unidade de tempo ou mesmo de custo, não dimensionando o quanto está sendo produzido. Quanto menos a empresa contratada produz em uma hora, mais ela será remunerada, ou seja, é a síntese do culto à produtividade. A administração da produtividade do contratado fica por conta de quem contrata, incorrendo em custos adicionais de controle. A aplicação da análise de PF é uma alternativa para solucionar este tipo de caso exposto e também um grande diferencial já exposto em trabalhos anteriormente apresentados.

## 2.9 TRABALHOS CORRELATOS

Atualmente métricas e estimativas estão focados como um dos principais tópicos de Engenharia de Software. O desenvolvimento da ferramenta através da métrica FPA tem o intuito de auxiliar empresas desenvolvedoras de software a estimar ou medir o processo de custo para o desenvolvimento de software.

Deimling (1999), utilizou as métricas de software como auxílio no desenvolvimento de sistema, com a finalidade de controlar e executar os trabalhos com maior precisão e qualidade. O trabalho por ele apresentado propôs o estudo de modelos de métricas e desenvolvimento de um protótipo para o gerenciamento de desenvolvimento de sistema em pequenas e médias empresas de software. Para a avaliação do processo de desenvolvimento de sistema, Deimling utilizou as métricas de FPA e COCOMO.

Seibt (2001) elaborou a construção de um protótipo de uma ferramenta capaz de analisar o código fonte de um projeto orientado a objetos em Delphi, extraíndo as classes seus métodos e atributos para posterior cálculo de métricas para softwares orientados a objetos. A

ferramenta, por ela desenvolvida, permite calcular dezenove métricas de projeto e de construção, como por exemplo, profundidade da árvore de herança e métodos ponderados por classe.

Valcanaia (1998) propôs o trabalho que complementa a especificação e implementação de um protótipo para contabilização e cálculo de pontos de função em sistemas desenvolvidos no banco de dados Microsoft Access 97. Neste, foram efetuados estudos de FPA, por ser a ferramenta de desenvolvimento do protótipo e pelo fato de que o protótipo analisa aplicações desenvolvidas no banco de dados.

O presente trabalho também traz como diferenciais dos demais trabalhos correlatos, a proposta de modelo para fechamento de negócio como forma de contrato do que será desenvolvido, evitando que ao final do desenvolvimento do produto não hajam divergências no acordado x realizado possibilitando o envio deste relatório por e-mail. Também possibilita efetuar estimativa inicial de custo através das variações de FPA método de contagem indicativa e método das estimativas percentuais, possibilitando efetuar análise comparativa entre elas através de gráfico. A verificação do custo por ponto de função para cada linguagem de programação utilizada para o desenvolvimento do software também é um dos diferenciais apresentados.

### 3 DESENVOLVIMENTO DO TRABALHO

Este capítulo descreve os principais requisitos, a especificação e a implementação do sistema. Por fim, são apresentados os resultados e discussão.

Com base nos objetivos propostos por este trabalho, desenvolveu-se um Sistema de Cálculo de Custo de Desenvolvimento de Software Utilizando FPA, versão 4.1 para atender a empresas desenvolvedoras de software que necessitem estimar e calcular o preço para desenvolvimento de determinado produto a ser desenvolvido.

#### 3.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

Os requisitos do sistema compreendem o levantamento das funcionalidades e/ou necessidades dos usuários para automatização pelo sistema de software.

A seguir, o quadro 18 apresenta os requisitos funcionais do sistema desenvolvido, definindo suas funcionalidades.

<b>Requisitos Funcionais</b>
RF01: O analista deverá ser capaz de cadastrar cliente.
RF02: O analista deverá ser capaz de cadastrar a linguagem de programação a ser utilizada no orçamento do desenvolvimento do sistema.
RF03: O analista deverá ser capaz de cadastrar o sistema a ser desenvolvido.
RF04: O analista deverá ser capaz de efetuar o cadastro /movimento com base na métrica FPA para o devido sistema (informar os dados para ALI, AIE, CE, SE, EE e FA)
RF05: O analista deverá ser capaz de cadastrar as informações / movimento das variantes FPA que são as informações necessárias para CI e MEP.
RF06: O sistema deverá ser capaz de emitir relatório “Resumo da Classificação das Funções” demonstrando a totalização da métrica, bem como valor total e linguagem utilizada.
RF07: O sistema deverá ser capaz de emitir relatório “Fluxo de Orçamento” demonstrando o total de PF estimado para CI, MEP e FPA
RF08: O sistema deverá ser capaz de emitir relatório “Linguagem x Custo do Sistema”, efetuando comparação entre valor total para desenvolvimento utilizando determinada linguagem de programação (comparação efetuada para MEP e CI)
RF09: O sistema deverá ser capaz de emitir relatório “Proposta Contratual”, com o objetivo de efetuar fechamento de contrato para desenvolvimento do software. Este poderá ser enviado por e-mail.

Quadro 18 – Requisitos Funcionais

O quadro 19 apresentado a seguir, lista os requisitos não funcionais do sistema desenvolvido, declarando as características do sistema que são relacionadas as suas funcionalidades.

<b>Requisitos Não Funcionais</b>
RNF01: Interface gráfica no padrão Windows
RNF02: Desenvolvido com base na versão 4.1
RNF03: Utiliza banco de dados Interbase
RNF04: Desenvolvido em linguagem Delphi 7.0

Quadro 19 – Requisitos Não Funcionais

## 3.2 ESPECIFICAÇÃO

A seguir são apresentadas as atividades desempenhadas na fase de especificação do sistema, como o Diagrama de Casos de Uso, o Modelo Entidade-Relacionamento e o Dicionário de Dados. Nos itens a seguir estas atividades são apresentadas detalhadamente.

Para os diagramas de casos de uso foi utilizada a Linguagem Unificada de Modelagem, com a ferramenta Enterprise Architect. , versão trial 5.0 da *Sparx System*.

Utilizou-se para elaboração do modelo de entidade relacionamento e dicionário de dados a ferramenta CASE Power Designer, versão trial 9.0 da *Sybase*.

### 3.2.1 Diagrama de casos de uso

Para Willemann (2004), “o diagrama de casos de uso é a especificação de interações entre um sistema e os agentes externos (atores) que utilizam o sistema”. Na modelagem foram observados nove casos de uso que são ilustrados na figura 03, onde é apresentado um ator, o analista que é a pessoa responsável pela área de engenharia software da empresa e será capaz de inserir os dados necessários para estimar o custo do software baseado na métrica FPA versão 4.1.

A descrição dos cenários encontra-se disponível no Apêndice A.

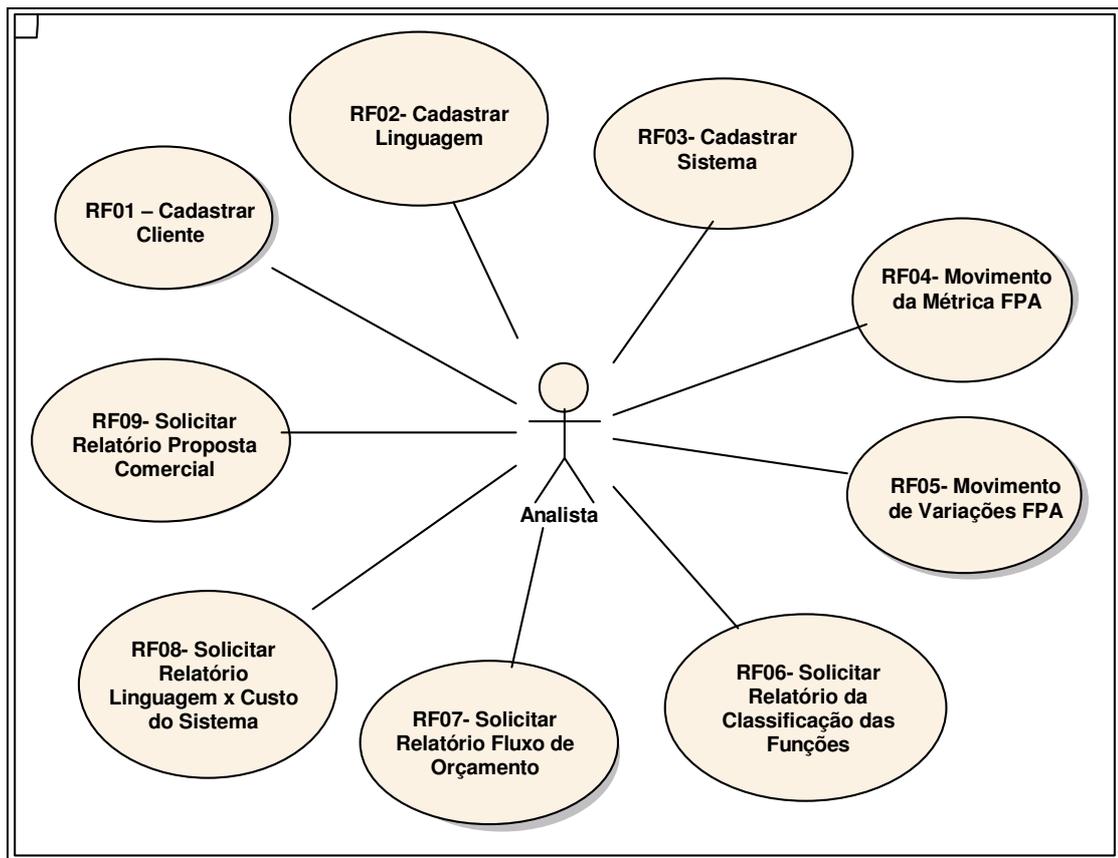


Figura 03 - Casos de Usos Efetuados pelo Analista

A seguir é apresentada uma breve descrição dos casos de uso.

- a) cadastrar cliente: permite ao analista cadastrar os clientes (podendo ser empresas), para que possa associar ao cadastro de sistema e suas devidas movimentações;
- b) cadastrar linguagem: permite ao analista (usuário do sistema), cadastrar a linguagem de programação a ser utilizada para efetuar cálculo de desenvolvimento do sistema utilizando a mesma;
- c) cadastrar sistema: permite o cadastro das informações do sistema a ser efetuado o levantamento do custo de desenvolvimento;
- d) movimento da métrica FPA: permite ao analista inserir informações baseadas na regra da métrica FPA para efetuar o cálculo da estimativa de custo para desenvolvimento do software (os dados são inseridos na tela de movimentação de AIE, ALI, CE, SE, EE e FA);

- e) movimento de variantes FPA – método de contagem indicativa (CI) e método das estimativas percentuais (MEP): permite inserir informações baseadas nas métricas de CI e MEP;
- f) solicitar relatório resumo da classificação das funções: possibilita a emissão de relatório apresentando o total das funções da métrica FPA, linguagem utilizada e valor total para desenvolvimento do software em questão;
- g) solicitar relatório fluxo de orçamento: possibilita a emissão de relatório apresentando gráfico comparativo entre o total de PF estimados na regra FPA, MEP e CI;
- h) solicitar relatório linguagem x custo do sistema: possibilita a emissão de relatório com finalidade de ilustrar o valor para desenvolvimento do sistema a ser desenvolvido utilizando as (determinadas) linguagens de programação cadastradas;
- i) solicitar relatório proposta contratual: possibilita a emissão de relatório e envio do mesmo por e-mail das informações necessárias para “fechamento de negócio” com o cliente, para posteriormente poder autorizar o desenvolvimento do sistema.

A figura 04 apresentada a seguir o diagrama de atividades do sistema.

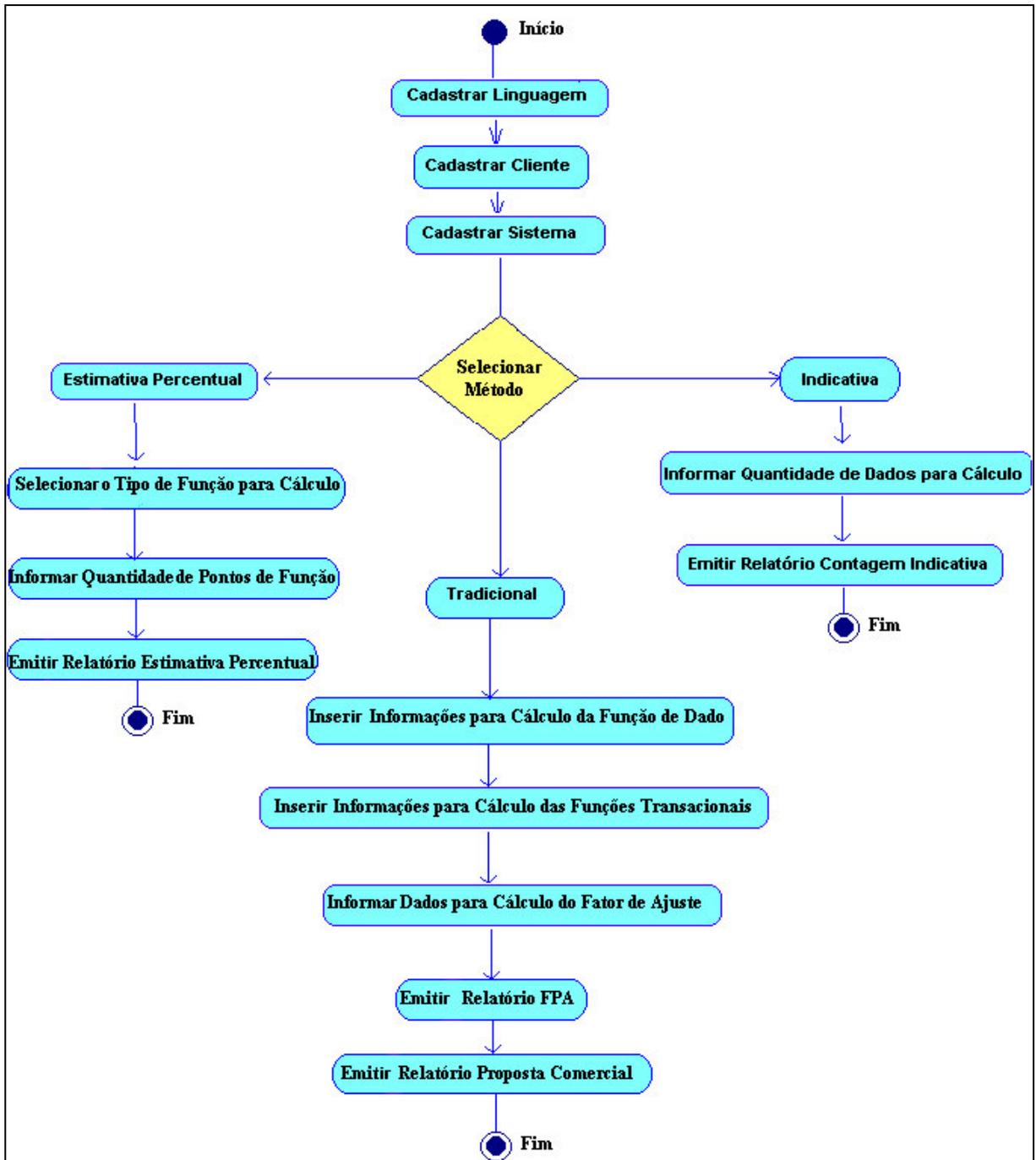


Figura 04 – Diagrama de Atividades

### 3.2.2 Modelo de Dados

Os modelos de dados geralmente são utilizados em conjunto com o modelo de fluxo de dados e servem para apresentar a estrutura da informação que está sendo processada.

A seguir, nas figuras 05 e 06 é demonstrado o modelo de Entidade-Relacionamento do sistema gerado pelo *Power Designer*, versão trial 9.0 da *Sybase*. O diagrama de entidade-relacionamento (também pode ser chamado de modelo de entidade-relacionamento (MER)), especifica os relacionamentos entre as entidades que fazem parte da administração da organização, podendo ser decomposto em modelos de dados detalhados. Tem como principal objetivo da sua construção, a criação de uma descrição da semântica dos dados da realidade e suas necessidades de informação.

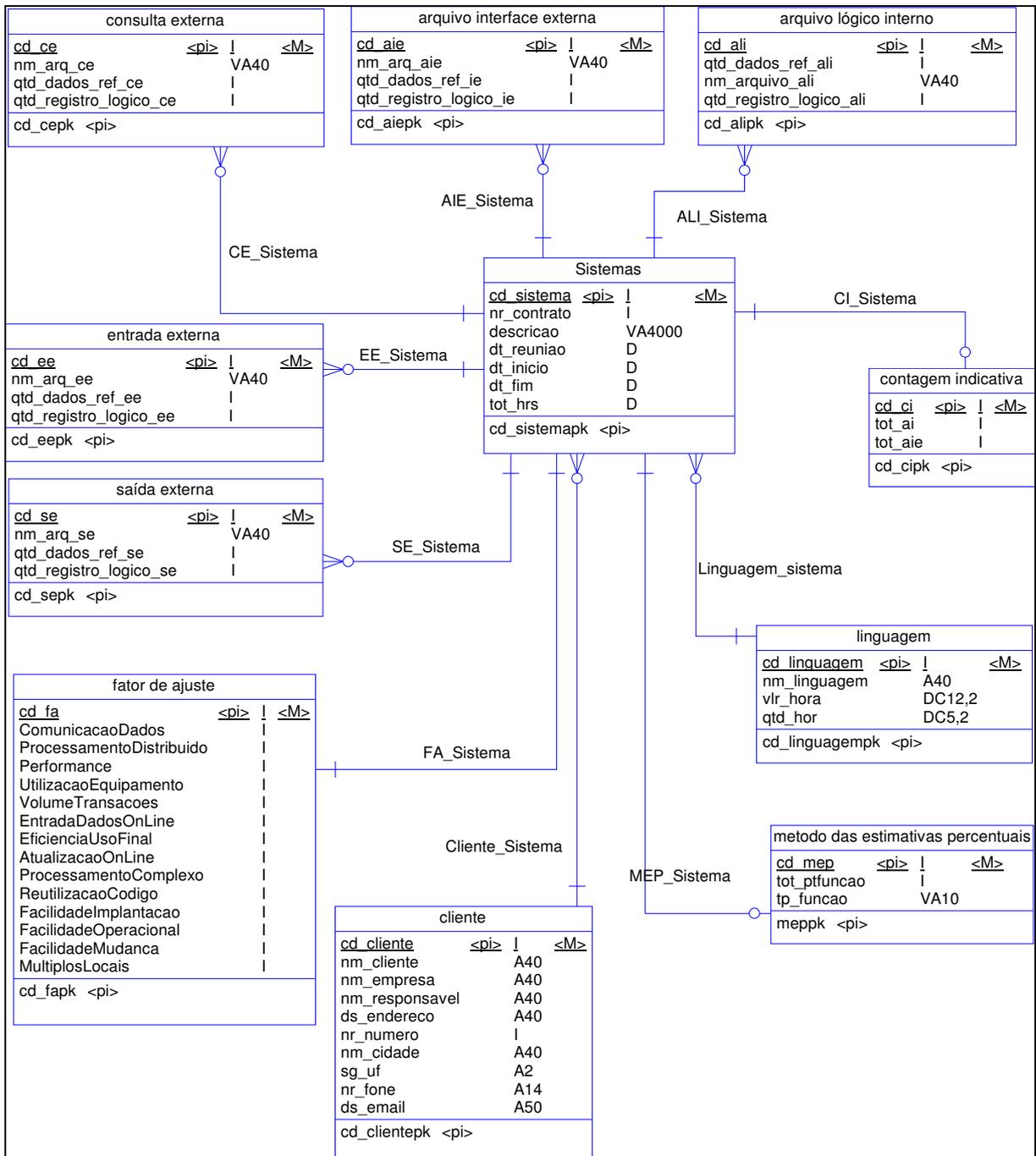


Figura 05 – Modelo Conceitual

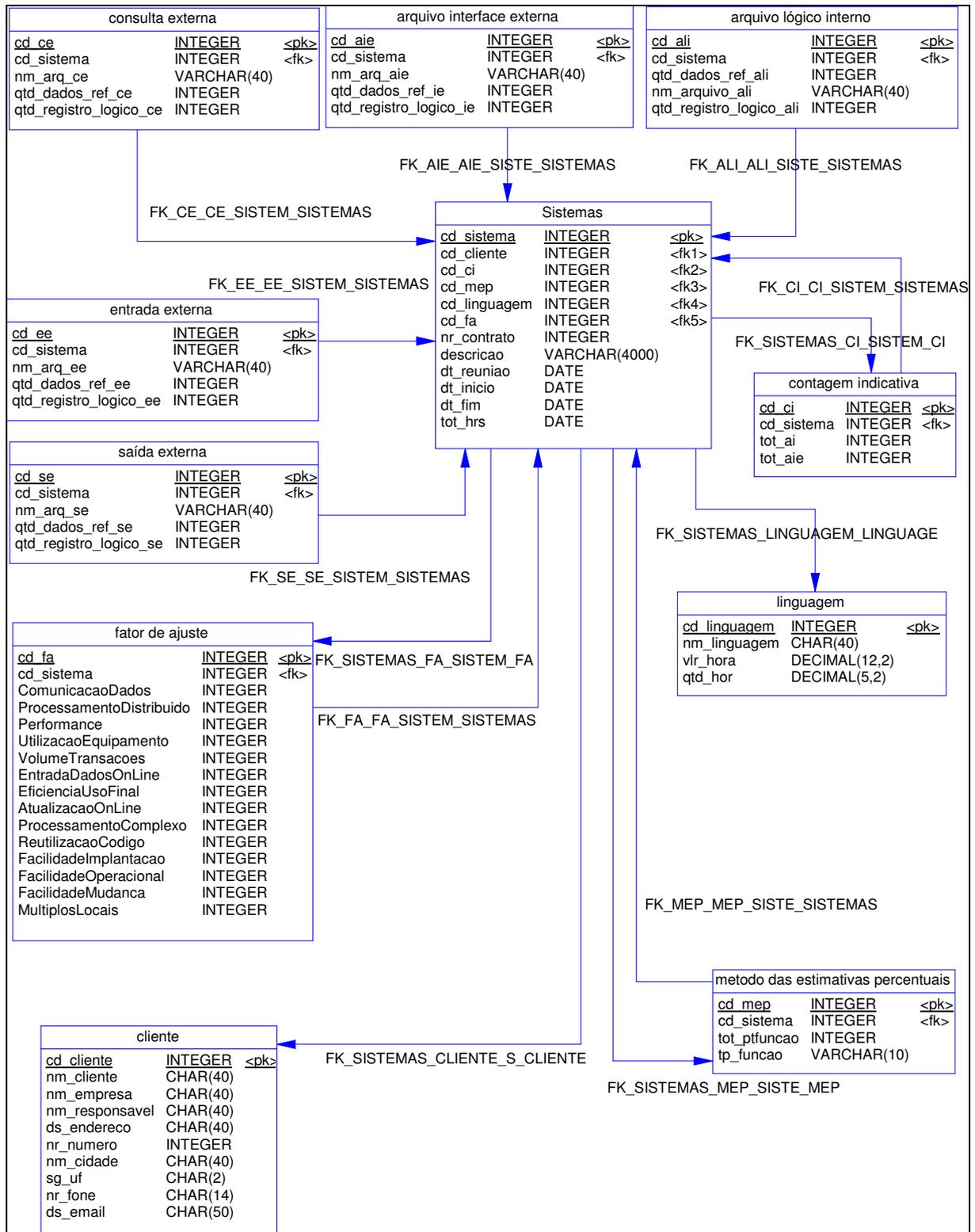


Figura 06 – Modelo Físico

### 3.2.3 Dicionário de dados

Sommerville (2003), afirma que o dicionário de dados é “uma lista alfabética dos nomes inclusos nos diferentes modelos do sistema”. Além do nome é necessário incluir uma descrição associada da entidade citada. O dicionário de dados do sistema é apresentado nas tabelas a seguir de 04 a 14 onde é apresentada na parte superior a descrição do que se refere cada tabela, na primeira coluna é ilustrado a descrição completa de cada informação para entendimento de cada item adicionado, na segunda coluna é apresentado o código do atributo na tabela, na terceira coluna é apresentado o tipo (variando entre Integer, Varchar(40), Char e date), e finalizando a quarta coluna ilustra a informação chave (podendo ser primária ou estrangeira).

<b>TABELA CONSULTA EXTERNA</b>			
<b>Descrição Completa</b>	<b>Cód. Atributo</b>	<b>Tipo</b>	<b>Chave</b>
Código da consulta externa	cd_ce	Integer	Primária
Código do sistema	cd_sistema	Integer	Estrangeira
Nome do arquivo de consulta externa	nm_arq_ce	Varchar(40)	
Quantidade de dados referenciados da consulta externa	qtd_dados_ref_ce	Integer	
Quantidade registro lógico da consulta externa	qtd_registro_lógico_ce	Integer	

Tabela 04 – Consulta externa

<b>TABELA DE ARQUIVO INTERFACE EXTERNA</b>			
<b>Descrição Completa</b>	<b>Cód. Atributo</b>	<b>Tipo</b>	<b>Chave</b>
Código arquivos de interface externa	cd_aie	Integer	Primária
Código do sistema	cd_sistema	Integer	Estrangeira
Nome do arquivo de interface externa	nm_arq_aie	Varchar(40)	
Quantidade de dados referenciados do arquivo de interface externa	qtd_dados_ref_ie	Integer	
Quantidade de registros lógicos de interface externa	qtd_registro_lógico_ie	Integer	

Tabela 05 – Arquivo de interface externa

<b>TABELA DE ARQUIVO LÓGICO INTERNO</b>			
<b>Descrição Completa</b>	<b>Cód. Atributo</b>	<b>Tipo</b>	<b>Chave</b>
Código de arquivo lógico interno	cd_ali	Integer	Primária
Código do sistema	cd_sistema	Integer	Estrangeira
Quantidade de dados referenciados dos arquivos lógicos internos	qtd_dados_ref_ali	Integer	
Nome do arquivo de arquivos lógicos internos	nm_arquivo_ali	Varchar(40)	
Quantidade de registros lógicos de arquivos lógicos internos	qtd_registro_lógico_ali	Integer	

Tabela 06 – Arquivo lógico interno

<b>TABELA DE ENTRADA EXTERNA</b>			
<b>Descrição Completa</b>	<b>Cód. Atributo</b>	<b>Tipo</b>	<b>Chave</b>
Código de entrada externa	cd_ee	Integer	Primária
Código do sistema	cd_sistema	Integer	Estrangeira
Quantidade de dados referenciados da entrada externa	qtd_dados_ref_ee	Integer	
Nome do arquivo de entrada externa	nm_arquivo_ee	Varchar(40)	
Quantidade de registros lógicos de entrada externa	qtd_registro_lógico_ee	Integer	

Tabela 07 – Entrada externa

<b>TABELA DE SAÍDA EXTERNA</b>			
<b>Descrição Completa</b>	<b>Cód. Atributo</b>	<b>Tipo</b>	<b>Chave</b>
Código de saída externa	cd_se	Integer	Primária
Código do sistema	cd_sistema	Integer	Estrangeira
Quantidade de dados referenciados da saída externa	qtd_dados_ref_se	Integer	
Nome do arquivo de saída externa	nm_arquivo_se	Varchar(40)	
Quantidade de registros lógicos de saída externa	qtd_registro_lógico_se	Integer	

Tabela 08 – Saída externa

<b>TABELA DE FATOR DE AJUSTE</b>			
<b>Descrição Completa</b>	<b>Cód. Atributo</b>	<b>Tipo</b>	<b>Chave</b>
Código do fator de ajuste	cd_fa	Integer	Primária
Código do sistema	cd_sistema	Integer	Estrangeira
Comunicação de dados	ComunicacaoDados	Integer	
Processamento distribuído	ProcessamentoDistribuido	Integer	
Performance	Performance	Integer	
Utilização de equipamentos	UtilizaçãoEquipamento	Integer	
Volume de Transações	VolumeTransacoes	Integer	
Entrada de dados on-line	EntradaDadosOnLine	Integer	
Eficiência de uso final	EficienciaUsoFinal	Integer	
Atualização de dados on-line	AtualizacaoOnline	Integer	
Processamento complexo	ProcessamentoComplexo	Integer	
Reutilização de código	ReutilizacaoCodigo	Integer	
Facilidade de implantação	FacilidadeImplantacao	Integer	
Facilidade operacional do sistema	FacilidadeOperacional	Integer	
Facilidade de mudança	FacilidadeMudanca	Integer	
Múltiplos locais	MultiplosLocais	Integer	

Tabela 09 – Fator de ajuste

<b>TABELA DE CONTAGEM INDICATIVA</b>			
<b>Descrição Completa</b>	<b>Cód. Atributo</b>	<b>Tipo</b>	<b>Chave</b>
Código da contagem indicativa	cd_ci	Integer	Primária
Código do sistema	cd_sistema	Integer	Estrangeira
Código da linguagem de programação	cd_linguagem	Integer	Estrangeira
Total de arquivo lógico interno	tot_ai	Integer	
Total de arquivos de interface externa	tot_aie	Integer	

Tabela 10 – Contagem indicativa

<b>TABELA DE LINGUAGEM</b>			
<b>Descrição Completa</b>	<b>Cód. Atributo</b>	<b>Tipo</b>	<b>Chave</b>
Código da linguagem de programação	cd_linguagem	Integer	Primária
Nome da linguagem de programação	nm_linguagem	Char(40)	
Valor por hora da linguagem	vlr_hora	Decimal(12,2)	
Quantidade de PF realizados por hora	Qtd_hora	Integer	

Tabela 11 – Linguagem

<b>TABELA DE MÉTODO DAS ESTIMATIVAS PERCENTUAIS</b>			
<b>Descrição Completa</b>	<b>Cód. Atributo</b>	<b>Tipo</b>	<b>Chave</b>
Código do método de estimativas percentuais	cd_mep	Integer	Primária
Código do sistema	cd_sistema	Integer	Estrangeira
Código da linguagem de programação	cd_linguagem	Integer	Estrangeira
Total de pontos por função	tot_pt_funcao	Integer	
Tipo de função	tp_funcao	Varchar(10)	

Tabela 12 – Método das estimativas percentuais

<b>TABELA DE CLIENTE</b>			
<b>Descrição Completa</b>	<b>Cód. Atributo</b>	<b>Tipo</b>	<b>Chave</b>
Código do cliente	cd_cliente	Integer	Primária
Nome do cliente	nm_cliente	Char(40)	
Nome da empresa	nm_empresa	Char(40)	
Nome do responsável	nm_responsavel	Char(40)	
Descrição do endereço	ds_endereco	Char(40)	
Número do endereço	nm_endereco	Integer	
Nome da cidade	nm_cidade	Char(40)	
Sigla do estado	sg_uf	Char(2)	
Número do telefone	nr_fone	Char(14)	
Descrição do endereço de e-mail	ds_email	Char(50)	

Tabela 13 – Cliente

<b>TABELA DE SISTEMAS</b>			
<b>Descrição Completa</b>	<b>Cód. Atributo</b>	<b>Tipo</b>	<b>Chave</b>
Código do sistema	cd_sistema	Integer	Primária
Código do cliente	cd_cliente	Integer	Estrangeira
Código do método de estimativa percentual	cd_mep	Integer	Estrangeira
Código da linguagem de programação	cd_linguagem	Integer	Estrangeira
Código do fator de ajuste	cd_fa	Integer	Estrangeira
Código da contagem indicativa	cd_ci	Integer	Estrangeira
Número do contrato	nr_contrato	Integer	
Descrição do sistema	ds_sistema	Varchar(4000)	
Data da reunião	dt_reuniao	Date	
Data de início	dt_inicio	Date	
Total de horas	tot_hrs	Date	

Tabela 14 – Sistemas

### 3.3 IMPLEMENTAÇÃO

Nesta seção são apresentadas as técnicas e ferramentas utilizadas para implementação do sistema desenvolvido, tais como Borland Delphi 7.0, Interbase.

#### 3.3.1 TÉCNICAS E FERRAMENTAS UTILIZADAS

Para a realização deste trabalho foram necessárias algumas técnicas e ferramentas para a especificação e o desenvolvimento do sistema.

O ambiente de desenvolvimento utilizado foi Delphi versão 7.0 da Borland. Para o banco de dados foi utilizado Interbase versão 1.0, também da Borland. Também foram utilizadas as ferramentas *Enterprise Architect* e *Power Designer*.

A ferramenta Delphi foi lançada pela Borland em 1995. É um ambiente de desenvolvimento de aplicações para ambiente Windows, podendo utilizar também técnicas para desenvolvimento rápido de aplicações e prototipação.

Possui como principal característica sua utilização para desenvolvimento de aplicações em banco de dados.

Alguns recursos que o Delphi possui são:

- a) ambiente integrado de desenvolvimento com editor de formulários, código fonte, *browser*, compilador e depurador;
- b) possibilidade de incluir componentes em formulários e alterar suas propriedades visualmente em tempo de desenvolvimento;
- c) possibilidade para liberar eventos dos componentes a código fonte visualmente;
- d) possibilidade de criar novos componentes utilizando o próprio Delphi;
- e) sistema de acesso a banco de dados como por exemplo Paradox; Interbase, SQL Servers e ODBC.

O Interbase é um Sistema Gerenciador de Banco de Dados (SGBD), desenvolvido pela Borland. Oferece suporte à linguagem *Structured Query Language* (SQL) de acordo com o padrão SQL92, além de mecanismos eficientes de segurança e integridade de dados. Permite gerenciar transações e utilizar recursos como visões (views), papéis (roles), procedimentos armazenados (*stored procedures*) e gatilhos (triggers) (NIEDERAUER, 2003).

As principais características do banco de dados Interbase são:

- a) sistema multi-gerencial: uso de um sistema otimista de "concorrência" no acesso ao banco de dados, ou seja, "guarda" várias versões dos registros mantendo assim uma visão consistente dos dados durante uma transação, independente de alguma informação ter sido alterada após a transação ter sido iniciada;
- b) configuração e manutenção: mantém a base de dados "limpa" e consistente através de rotinas automáticas de manutenção (SWEEP) executadas geralmente quando o BD está em "idle" (estado de espera);
- c) suporte a domínios : suporta o uso de domínios na definição de campos;
- d) multiplataforma: compatível com os sistemas operacionais Windows, Linux (i386), Solaris (Sparc), HP-UX (i386), MacOS X, FreeBSD, Solaris (i386) e AIX.

É uma ferramenta de análise e design UML, possibilitando o desenvolvimento de softwares a partir de um conjunto de requisitos, análise de estágios, modelos de design, testes e manutenção. Enterprise Architect é uma ferramenta multi-usuário, com base no ambiente Windows e projetada para ajudar a construir softwares robustos e eficazes. Os recursos flexíveis permitem auxiliar no desenvolvimento de suas aplicações.

O Enterprise Architect possui como principais características:

- a) Diagramas Estruturais:
  - classe;
  - objeto;
  - composição;
  - pacote;
  - componente;
  - distribuição.
- b) Diagramas de comportamento:
  - caso;
  - comunicação;
  - seqüência;
  - vista geral da interação;
  - atividade;
  - estado e sincronismo.
- c) Extensão:
  - análise (atividade simples);
  - personalizada (para requerimentos, alterações, UI).

O *Power Designer* é uma ferramenta voltada para ambiente de análise e designer de aplicações corporativas, disponibilizando recursos para modelagem de negócio, dados e objetos.

As principais características que podemos destacar para o *Power Designer* são:

- a) modelagem de processos e negócios;
- b) modelagem de dados;
- c) modelagem de objetos;
- d) modelagem integrada;
- e) repositório corporativo.

## Operacionalidade da Implementação

Para melhor compreensão do sistema são apresentadas a seguir, telas do sistema com um estudo de caso fictício: Sistema de Hotel.

O Hotel Bem Vindo pretende adquirir um sistema para controle de reservas dos clientes. Pretende através de terminais ligados ao computador central, receber e fornecer informações referentes às reservas efetuadas no momento de sua solicitação, registrando em base de dados própria e validando, se necessário, na base de dados do departamento de pessoal. O sistema deve permitir ainda, emissão de relatórios para apoio do setor de recepção e faturamento.

O sistema possuirá como funcionalidades:

- registro de reservas;
- alteração de reservas;
- exclusão de reservas;
- cadastrar clientes;
- alteração de dados cadastrais de clientes;
- exclusão de clientes;
- cadastramento de informações sobre apartamentos;
- alteração de dados de apartamentos;
- exclusão de apartamentos.

Os produtos gerados são:

Relatórios:

- reservas de um período;
- clientes ativos;

- dados dos apartamentos;

Consultas:

- informações de reservas (dado o código);
- informações de clientes;
- informações de apartamentos;

Características do sistema:

- trabalha de forma on-line, via terminais ou microcomputadores;
- atualiza informações on-line e em tempo real;
- os procedimentos de recuperação deverão minimizar a intervenção do operador;
- navegação através de menus;
- desenvolvido em linguagem Delphi;
- existem poucas restrições do caráter operacional;
- existem grandes picos de transação mensais;
- sistema voltado ao usuário final.

Para isso o analista que irá efetuar a análise de requisitos do sistema deverá:

Primeiramente acessar a tela principal do sistema para posteriormente acessar os demais menus, conforme ilustrado na figura 07.

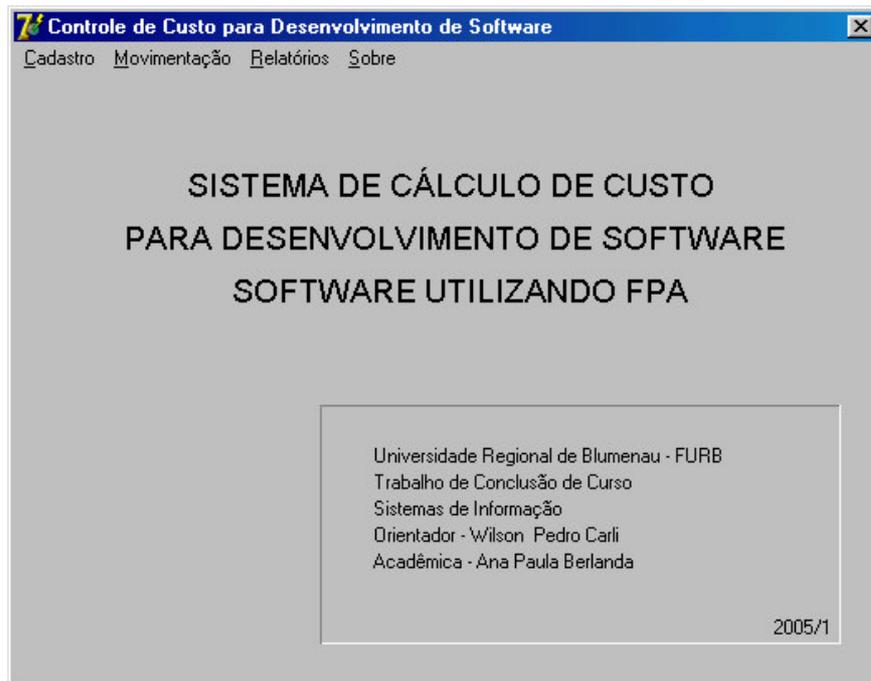


Figura 07 – Tela Principal do Sistema

#### Cadastrando Linguagem de Programação:

Primeiro, devemos cadastrar as possíveis linguagens de programação utilizadas pela empresa desenvolvedora de software para efetuar orçamento dos sistemas a serem orçados para desenvolvimento. Para isso, entre no menu Cadastro/Linguagem, onde deverá ser informado o nome da linguagem de programação, valor hora e a quantidade de PF por hora (para preenchimento deste campo verificar as tabelas 15, 16 e 17 apresentadas anteriormente).



Figura 08 – Tela de Cadastro de Linguagem

### Cadastrando Cliente:

Para cadastro de cliente, basta acessar o menu Cadastro / Cliente da tela principal do sistema. Nesta tela deverão ser preenchidas informações do cliente / empresa solicitante do orçamento, bem como nome cliente / empresa, nome do responsável (para entrar em contato caso necessário), endereço, número, cidade, UF, telefone para contato e e-mail da empresa ou responsável para fechamento de negócio (envio de relatório de proposta contratual), conforme ilustra a figura 09:

The screenshot shows a window titled "Cadastro de Clientes" with a table and a form below it. The table has three columns: "Nome do Cliente / Empresa", "Responsável", and "Telefone". The first row contains the data: "Hotel Bem Vindo", "Adalberto Nogueira", and "221 2121". Below the table is a form with the following fields:

Nome do Cliente / Empresa		Hotel Bem Vindo	
Nome do Responsável		Adalberto Nogueira	
Endereço	Número	Rua das Missões	77
Cidade	U.F.	Blumenau	SC
Telefone	E-mail	221 2121	mystery@terra.com.br

Figura 09 – Tela de Cadastro de Cliente

### Cadastrando Sistema:

Devemos registrar o sistema que deve ser orçado o custo para desenvolvimento de software, para cadastro do sistema devemos acessar o menu Cadastro / Sistema através do menu principal, conforme ilustra a figura 10, e preencher os campos nome cliente / empresa em questão, nome do sistema a ser orçado, nome da linguagem a ser utilizada para

desenvolvimento, número do contrato, data da reunião, data de início para desenvolvimento do software, hora / semana dedicado para desenvolvimento e descrição do sistema onde permita inserir observações do mesmo.

Nr. do Contrato	Sistema	Nome do Cliente / Empresa
MAS-123	Controle de Reservas de Client	Hotel Bem Vindo

Cód. Cliente: 1      Nome do Cliente / Empresa: Hotel Bem Vindo  
 Nome do Sistema: Controle de Reservas de Clientes  
 Cód. Linguagem: 3      Nome da Linguagem: Delphi  
 Número do Contrato: MAS-123      Data da Reunião: 01/02/2005      Data de Início: 01/03/2005      Horas/Semana: 8  
 Descrição do Sistema: O sistema deverá efetuar o controle de reservas de clientes.

Figura 10 – Tela de Cadastro de Sistema

Registrando as Funções:

Após ter efetuado todos os cadastros necessários (linguagem de programação, cliente e sistema), devemos registrar as funções que o aplicativo deverá implementar para então obter um valor de orçamento para desenvolvimento do sistema.

Nas figuras 11 e 12 podemos verificar como seria registrada as funções do tipo dados (ALI e AIE) do estudo de caso proposto. Para registro destas informações devemos acessar a tela de menu Movimentação / Controle Sistema.

Nas telas de Movimentações / Controle do sistema, ao gravar o registro, a ferramenta calcula a complexidade da função e o número de pontos por função correspondentes.

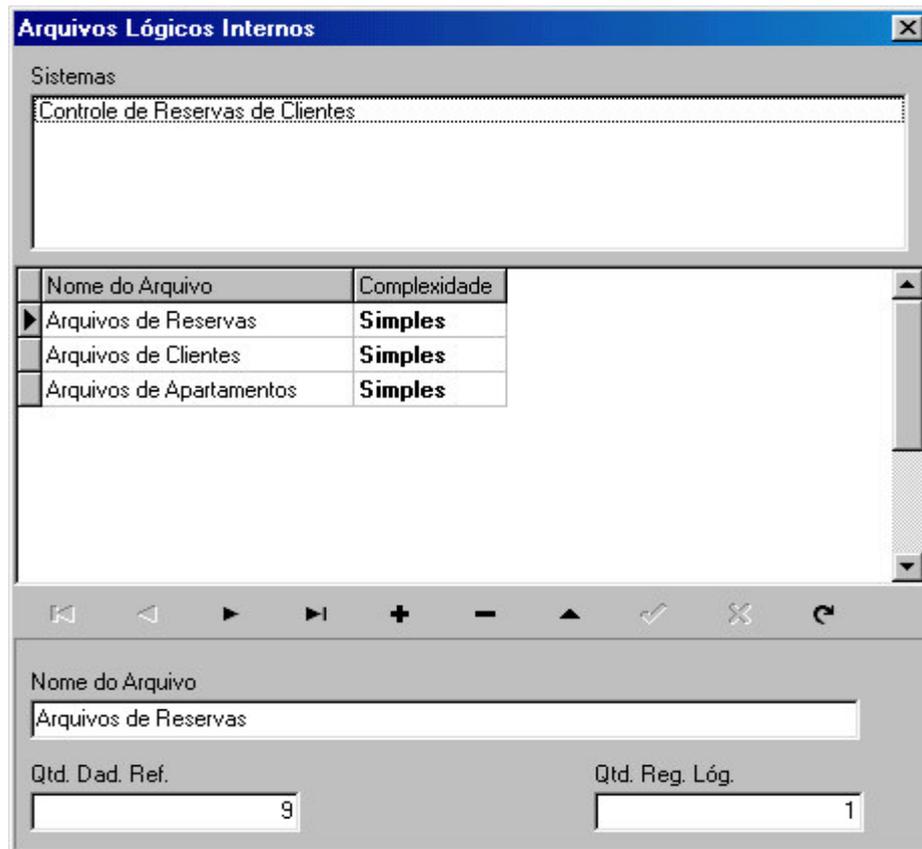


Figura 11 – Tela de Movimentação de Arquivos Lógicos Internos

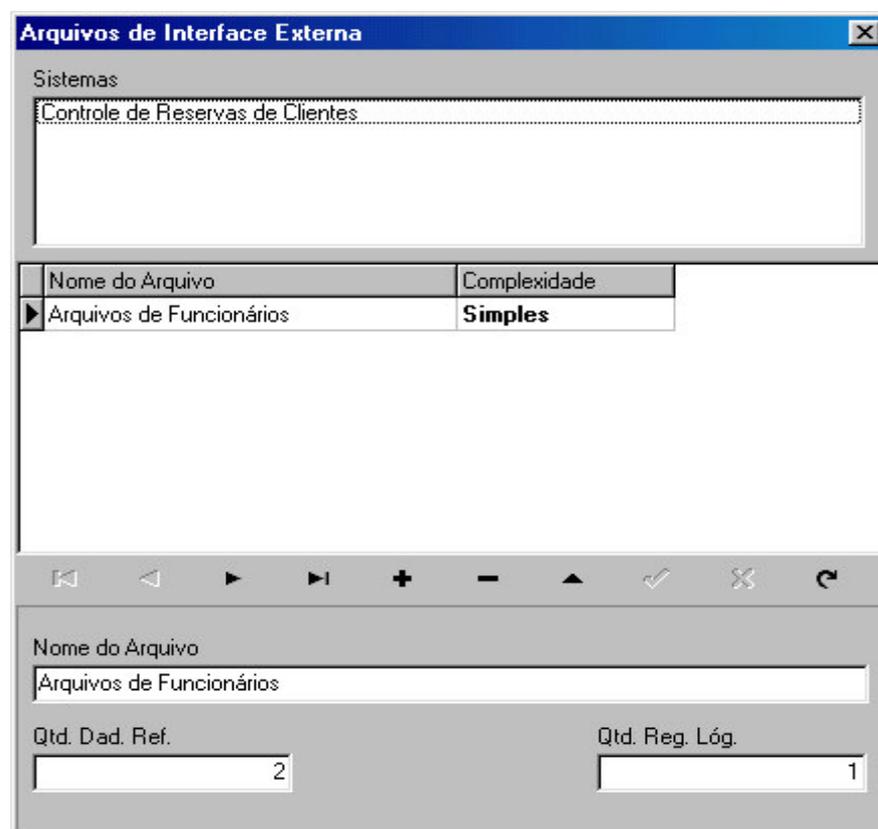


Figura 12 – Tela de Movimentação de Arquivos de Interface Externa

Nas figuras 13, 14 e 15 podemos verificar como seriam registradas as funções do tipo transação (CE, EE e SE) do estudo de caso proposto. Para registro destas informações devemos acessar a tela de menu Movimentação / Controle Sistema.



The screenshot shows a window titled "Consultas Externas" with a close button. It contains a list of systems under the heading "Sistemas", with "Controle de Reservas de Clientes" selected. Below this is a table with two columns: "Nome do Arquivo" and "Complexidade". The table lists three queries: "Consulta de Apartamentos" (Simple), "Consulta de Clientes" (Média), and "Consulta de Reservas" (Complexa). At the bottom, there is a control bar with navigation icons and a data entry section. The "Nome do Arquivo" field contains "Consulta de Apartamentos". The "Qtd. Dad. Ref." field contains the value "4", and the "Qtd. Reg. Lóg." field contains the value "2".

Nome do Arquivo	Complexidade
Consulta de Apartamentos	<b>Simple</b>
Consulta de Clientes	<b>Média</b>
Consulta de Reservas	<b>Complexa</b>

Nome do Arquivo: Consulta de Apartamentos

Qtd. Dad. Ref.: 4

Qtd. Reg. Lóg.: 2

Figura 13 – Tela de Movimentação de Consultas Externas

**Entradas Externas**

Sistemas

Controle de Reservas de Clientes

Nome do Arquivo	Complexidade
Exclusão de Clientes	<b>Simple</b>
Exclusão de Apartamentos	<b>Simple</b>
Alteração de Apartamentos	<b>Simple</b>
Inclusão de Apartamentos	<b>Simple</b>
Exclusão de Reservas	<b>Simple</b>
Alteração de Clientes	<b>Média</b>
Inclusão de Clientes	<b>Média</b>
Registro de Reservas	<b>Complexa</b>

Nome do Arquivo

Exclusão de Clientes

Qtd. Dad. Ref.  Qtd. Reg. Lóg.

Figura 14 – Tela de movimentação de Entradas Externas

**Saídas Externas**

Sistemas

Controle de Reservas de Clientes

Nome do Arquivo	Complexidade
Relatório de Apartamentos	<b>Simple</b>
Relatório de Clientes	<b>Simple</b>
Relatório de Reservas	<b>Simple</b>

Nome do Arquivo

Relatório de Apartamentos

Qtd. Dad. Ref.  Qtd. Reg. Lóg.

Figura 15 – Tela de Movimentação de Saídas Externas

A figuras 16 ilustra onde podemos verificar como seria registrada o fator de ajuste do estudo de caso proposto. Para registro destas informações devemos acessar a tela de menu Movimentação / Controle Sistema / Fator Ajuste.

The screenshot shows a window titled "Fatores de Ajustes" with a list of systems. The system "Controle de Reservas de Clientes" is selected. Below the list is a table with two columns: "Nível de Influência" and "Fator de Ajuste". The selected system has a value of 18 in the first column and 0.83 in the second. Below the table are various adjustment factors, each with a dropdown menu.

Nível de Influência	Fator de Ajuste
18	0.83

Comunicação de Dados 4 - Influência Significativa	Atualização On-Line 5 - Grande Influência
Processamento Distribuído 0 - Nenhuma influência	Complexidade Processamento 0 - Nenhuma influência
Performance 0 - Nenhuma influência	Reutilização de Código 0 - Nenhuma influência
Utilização de Equipamentos 1 - Influência Mínima	Facilidade de Implantação 0 - Nenhuma influência
Volume de Transações 1 - Influência Mínima	Facilidade Operacional 1 - Influência Mínima
Entrada de Dados On-Line 5 - Grande Influência	Facilidade de Mudanças 0 - Nenhuma influência
Eficiência - Uso Final 1 - Influência Mínima	Multiplos Locais 0 - Nenhuma influência

Figura 16 – Tela de Movimentação de Fator de Ajuste

Utilizando Variações de FPA:

Caso o cliente desejar efetuar uma avaliação do custo para desenvolvimento do software antes de ter conhecimento de todas as funções existentes no sistema em questão, pode-se utilizar o cálculo das variações de FPA que são os métodos de CI e MEP. Sendo que para a CI, ilustrada na figura 17 (a tela pode ser acessada através do menu principal Movimentação / Contagem Indicativa), é necessário ter conhecimento apenas dos ALI e/ou AIE. Para utilização do MEP que pode ser acessada pelo menu principal Movimentação / Método Percentual), ilustrada na figura 18, basta saber apenas uma das funções.

Tot. Arq. Lóg. Int.	Tot. Arq. Int. Ext.	Total de P.F.
5	75	

Figura 17 – Tela de Movimento de Contagem Indicativa

Tipo de Função	Total P.F.	Total Arquivo	Total Interface	Total Consulta	Total Entrada	Total Saída	Total Geral
Interface	5	420	35	92	200	235	982

Figura 18 – Tela de Movimento de Estimativas Percentuais

Emitindo Relatórios:

Relatório de resumo da classificação das funções: este relatório tem por finalidade apresentar ao usuário a totalização do resumo das funções, total do fator de ajuste, total de pontos de funções ajustados e não-ajustados, linguagem selecionada para estimar o valor do desenvolvimento do software, total de PF por hora, valor da hora por ponto de função / linguagem selecionada e o valor total do custo para desenvolvimento de software.

Para acessar este modelo de relatório, clique no menu Relatórios / Resumo da Classificação das Funções, onde será apresentada a tela do filtro do relatório, conforme ilustra a figura 19, selecione o sistema desejado para apresentação dos dados, posteriormente clique no botão Imprimir. A ilustração do modelo do relatório é apresentado na figura 20.



Figura 19 – Tela do Filtro de Relatório de Resumo da Classificação das Funções

**Resumo da Classificação das Funções** 18/07/05

**Sistema:** Controle de Reservas de Clientes  
**Linguagem:** Delphi

	Simple	Média	Complexa	Totais
<b>Arquivos</b>	21	0	0	21
<b>Interfaces</b>	5	0	0	5
<b>Entradas</b>	15	8	12	35
<b>Saídas</b>	12	0	0	12
<b>Consultas</b>	3	4	6	13
<b>Total de P.F não ajustados</b>				<b>86</b>

**Fator de Ajuste:** 0,83  
**Total de P.F ajustados:** 71,38 pontos  
**P.F por Hora:** 3 pontos  
**Total de Horas:** 23,79 Horas  
**Valor Hora:** R\$ 6,67  
**Valor Final:** R\$ 475,87

0% Page 1 of 1

Figura 20 – Tela do Relatório de Resumo da Classificação das Funções

Relatório fluxo de orçamentos: este relatório tem por finalidade ilustrar através de relatório gráfico comparativo os totais de PF calculados e estimados entre FPA, MEP e CI. Para acessar este modelo de relatório, clique no menu Relatórios / Fluxo de Orçamentos, onde será apresentada a tela do filtro do relatório, conforme ilustra a figura 21, devendo ser selecionado o sistema a ser efetuado a análise comparativa, posteriormente clique no botão Imprimir. A ilustração do modelo deste relatório é apresentado na figura 22.



Figura 21 – Tela do Filtro de Relatório Fluxo de Orçamento

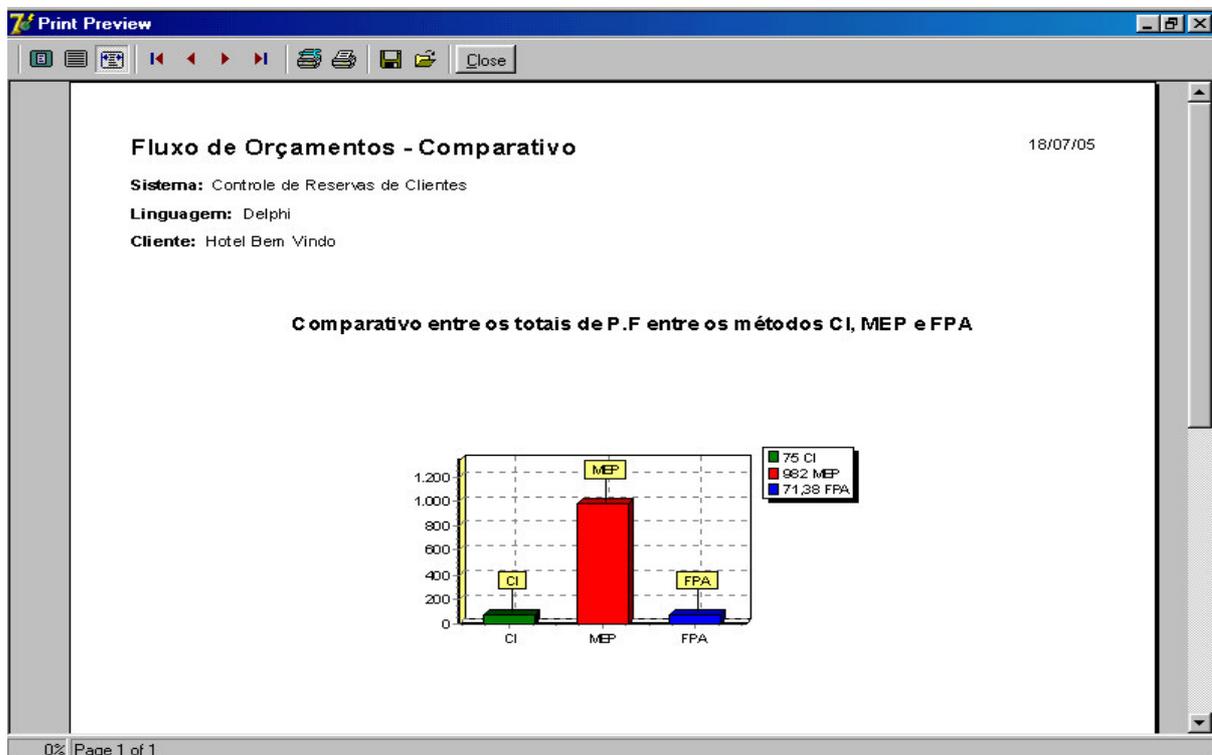


Figura 22 – Tela do Relatório Fluxo de Orçamento

Relatório linguagem x custo do sistema: a finalidade deste relatório é demonstrar o custo para desenvolvimento de software para cada linguagem de programação cadastrada no sistema com base nos valores de totais de PF estimados pelo MEP e CI, auxiliando o usuário na escolha da linguagem de programação a ser utilizada para desenvolvimento. A figura 23 ilustra o filtro do relatório de linguagens de programação x custo sistema, neste filtro deverá ser selecionado o método a ser utilizado para a comparação (Contagem Indicativa ou Método de Estimativas percentuais).

Este modelo de relatório pode ser acessado através do menu Relatórios / Linguagem x Custo Sistema. O modelo deste relatório é apresentado na figura 24, onde na primeira coluna é apresentado o nome das linguagens de programação, na segunda coluna é listado o valor da hora correspondente a linguagem, a terceira coluna apresenta o total de PF produzido por hora com a determinada linguagem e última coluna é a estimativa do valor total.

Nome do Sistema	Nome do Cliente
Controle de Reservas de Clientes	Hotel Bem Vindo

Estimativa de Custo

Contagem Indicativa       Método de Estimativas Percentuais

Imprimir

Figura 23– Tela do Filtro Relatório Linguagens X Custo Sistema

**Linguagens X Custo do Sistema** 18/07/05

**Sistema:** Controle de Reservas de Clientes  
**Cliente:** Hotel Bem Vindo

**Estimativa de Custo - Contagem Indicativa - Total de 75 P.F.**

Linguagem	Valor Hora	P.F por Hora	Estimativa do Valor Total
Delphi	R\$ 20,00	3	R\$ 500,00
PHP	R\$ 12,00	2,7	R\$ 333,33
Visual Basic	R\$ 15,00	2,7	R\$ 416,67

300% Page 1 of 1

Figura 24– Tela do Relatório Linguagens X Custo Sistema

Relatório proposta comercial: a finalidade deste relatório é listar todas as informações necessárias ao cliente para “fechamento de contrato”, deixando evidente o valor do custo cobrado para o desenvolvimento do software. Também possui o intuito de evitar o desperdício de materiais de escritório.

Este relatório pode ser acessado através do menu Relatórios / Proposta Comercial. A figura 25 apresenta a tela do filtro do relatório, onde possui como escolha as opções de imprimir (para listar as informações em vídeo) ou enviar por e-mail. Caso a opção de envio por e-mail seja selecionada, o relatório será enviado para o endereço de e-mail cadastrado na tela de cadastro de cliente, caso não possui e-mail cadastrado ou o computador não estiver conectado a internet o relatório emitirá mensagem: “Erro ao enviar e-mail!”, conforme demonstrado na figura 26. O modelo deste relatório é apresentado na figura 27.



Figura 25– Tela do Filtro do Relatório Linguagens X Custo Sistema

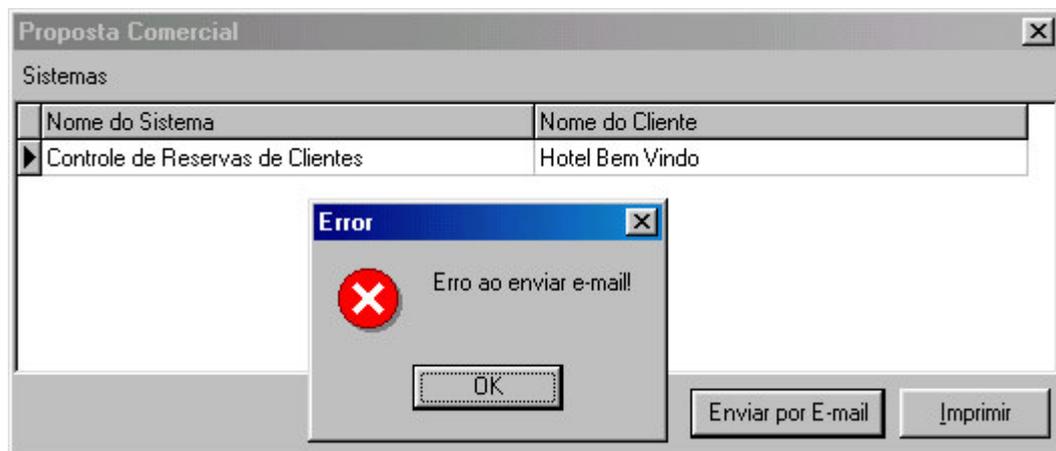


Figura 26– Tela do Filtro do Relatório Linguagens X Custo Sistema

<b>Proposta Comercial</b>		18/07/05
Sistema:	Controle de Reservas de Clientes	
Linguagem:	Delphi	
Cliente:	Hotel Bem Vindo	
Contrato:	MAS-123	
Responsável:	Adalberto Nogueira	
<b>Descrição dos Serviços Solicitados</b>		
O sistema deverá efetuar o controle de reservas de clientes, bem como		
<b>Tempo Previsto</b>		
Início do Sistema:	01/03/2005	
Tempo estimado:	0 meses	
Término do Sistema:	22/03/2005	
<b>Valores Orçados</b>		
Total de P.F.:	71,38 pontos	

Figura 27– Tela do Relatório Linguagens X Custo Sistema

### 3.4 RESULTADOS E DISCUSSÃO

Os métodos de contagem indicativa e das estimativas percentuais mostraram-se eficazes quanto ao fornecimento de informações relevantes ao processo decisório para escolha de linguagens de programação a ser utilizada no desenvolvimento do sistema. Sendo que o m

Com base na figura 22, pode-se verificar que o total de PF que mais se aproximou do FPA foi o método de contagem indicativa. Também deve ser levado em consideração que o total de PF ajustados foram 71,38 e os não ajustados aproximadamente 86.

Quanto à linguagem de programação utilizada para desenvolvimento, o analista deve levar em consideração não apenas o valor, mas a quantidade de pontos por função que é possível ser desenvolvido por hora. Também cabe ao analista dominar as funcionalidades de cada linguagem para posteriormente indicar a melhor linguagem para desenvolvimento de um determinado sistema.

Outra funcionalidade importante é o relatório de proposta contratual para auxílio e agilidade de informações podendo ser enviado por e-mail.

## 4 CONCLUSÕES

Atualmente, as organizações necessitam de informações para auxiliar no processo decisório. Com a utilização da métrica FPA é possível estabelecer um processo padronizado para efetuar o levantamento de custo para desenvolvimento de software. Os métodos de contagem indicativa e métodos de estimativas percentuais são bons para inicialmente prever o valor do custo final. Com a inclusão do tipo de linguagem de programação utilizada também foi possível aproximar o tempo para realização do desenvolvimento do mesmo, uma vez que dá suporte às funções de planejamento, controle e operação de uma organização, fornecendo informação uniforme.

Foi visto que utilizando cadastro de linguagem de programação é possível efetuar um levantamento de tempo para desenvolvimento de software, permitindo efetuar comparações entre si. A especificação do custo utilizando o método de contagem indicativa e método das estimativas percentuais também é de grande importância para se obter um princípio de dimensão do projeto e valores iniciais aproximados. Com a utilização do modelo de proposta comercial obtem-se redução de custos de materiais de escritório, agilidade para fechamento de contrato além de expor ao cliente de forma detalhada informações do que será desenvolvido e pago.

O trabalho desenvolvido foi de grande importância para a autora, pois possibilitou experiência de aprendizagem antes não conhecida tão profundamente na linguagem e ferramentas utilizadas para desenvolvimento do mesmo. Também adquiriu prática de conceitos aprendidos durante o curso de Sistemas de Informação.

### 4.1 EXTENSÕES

Como sugestão de extensão deste trabalho pode-se estudar a viabilidade de integrar o sistema com gerência de projetos, desenvolvimento para utilização WEB, além de efetuar o aperfeiçoamento do mesmo inserindo *help*, podendo também inserir restrições ao acesso de usuários, ou ainda, incluir alguma métrica para avaliação de qualidade.

## REFERÊNCIAS BIBLIOGRÁFICAS

BEZERRA, Eduardo. **Princípios de análise e projetos de sistemas com uml**. Rio de Janeiro: Campus, 2002.

DEIMLING, Jean R. Silva. **Protótipo para gerenciamento de custos de desenvolvimento de software em pequenas e médias empresas de software**. Trabalho de Conclusão de Curso de Ciências da Computação. Universidade Regional de Blumenau, 1999.

BFPUG, **Perguntas e Dúvidas Frequentes**, 1999. Disponível em <<http://www.bfpug.com.br/FAQ.htm>>. Acessado em 11 set. 2004.

FERNANDEZ, Aguinaldo A. **Gerência de software através de métricas**. São Paulo:Atlas,1995.

FUCK, Mônica Andréa. **Estudo e aplicação de métricas da qualidade do processo de desenvolvimento de aplicações em banco de dados**. Trabalho de Conclusão de Curso Ciências da Computação. Universidade Regional de Blumenau, 1994.

HAZAN, Claudia. Análise de pontos por função: uma abordagem gerencial. In: Integração Universidade – Empresa. Petrópolis. **Anais da XIX jornada de atualização em informática**. Curitiba: UCPR, 2000. p. 287-325.

NIEDERAUER, Juliano. **Tutorial de Delphi**, Porto Alegre, [20032]. Disponível em: <<http://www.novateceditora.com.br/guias/interbase/>>. Acesso em: 14 ago. 2005.

PAULA FILHO, Wilson de Pádua. **Engenharia de software: fundamentos, métodos e padrões**. LTC: Rio de Janeiro, 2001.

PRESSMAN, Roger S. **Engenharia de software**. Makron Books: São Paulo, 1995.

SEIBT, Patricia R. Ramos. **Ferramenta para cálculo de métricas em softwares orientados a objetos codificados em delphi**. Trabalho de Conclusão de Curso Ciências da Computação. Universidade Regional de Blumenau, 2001.

SILVA, Ivan J. de Mecnas. **Delphi 5: análise de pontos de função**. Book Express: Rio de JaneiroBook, 2000.

SOMMERVILLE, Ian. **Engenharia de software**. Addison Wesley: São Paulo, 2003.

VALCANAIA, Tibério César. **Protótipo de ferramenta de cálculo de fpa sobre microsoft access 97**. Trabalho de Conclusão de Curso Ciências da Computação. Universidade Regional de Blumenau, 1998.

VAZQUEZ, Carlos Eduardo c. et al. **Análise de pontos de função: medição estimativas e gerenciamento de projetos de software**. São Paulo: Érika, 2003.

WILLEMANN, Débora. **Avaliação da qualidade em pacotes de software utilizando a norma br iso/iec 12119**. 2004. 94 f. Trabalho de Conclusão de Curso (Bacharelado em Sistemas de Informação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

## APÊNDICE A – DESCRIÇÃO DOS CASO DE USO

### RF01- Cadastrar Cliente

#### *Cenários*

##### Cadastrar Clientes {Principal}.

- 1) Analista seleciona opção do Menu “Cadastro / Cliente”.
- 2) Sistema solicita que as informações necessárias (nome do cliente, nome do responsável, endereço, número, cidade, uf, telefone, e-mail), sejam cadastradas.
- 3) Analista cadastra os dados.
- 4) Os dados são gravados no banco de dados.

##### Altera Dados Cadastrados de Cliente {Alternativo}.

Poderá alterar os dados cadastrados.

##### Exclui Cliente{Alternativo}.

O sistema deverá permitir a exclusão de cliente, solicitando confirmação de exclusão.

### RF02- Cadastrar Linguagem

#### *Cenários*

##### Cadastrar Linguagem {Principal}.

- 1) Analista seleciona opção do Menu “Cadastro / Linguagem”.
- 2) Sistema solicita que as informações necessárias (nome da linguagem, valor hora e quantidade de PF por hora), sejam cadastradas.
- 3) Analista cadastra os dados.
- 4) Os dados são gravados no banco de dados.

##### Altera Dados Cadastrados de Linguagem {Alternativo}.

Poderá alterar os dados cadastrados..

### Exclui Linguagem{ Alternativo}.

O sistema deverá permitir a exclusão de linguagem já cadastrada, devendo solicitar confirmação para exclusão.

RF03- Cadastrar Sistema.

### ***Cenários***

#### Cadastrar Sistema {Principal}.

1. Usuário seleciona opção do Menu “Cadastro Sistema”.
2. Sistema solicita que os campos (Número do Contrato, Data de Início, Data reunião, Nome Software, Descrição do sistema, Horas / Semana) sejam preenchidos.
3. Usuário preenche os campos.
4. Sistema solicita que os campos (Nome do Cliente / Empresa e Nome da Linguagem) sejam selecionados.
5. Usuário seleciona os campos.
6. Os dados são gravados no banco de dados.

#### Altera Dados Cadastrados de Sistema { Alternativo}.

Poderá alterar os dados cadastrados.

#### Exclui Sistema

O sistema deverá permitir a exclusão de software já cadastrado. Caso já houver algum tipo de movimentação (ALI, AIE, CE, EE, SE, MEP ou CI), o sistema deverá excluir automaticamente as informações destas movimentações.

RF04- Movimento da Métrica FPA.

### ***Cenários***

#### Movimento da Métrica FPA {Principal}.

1. Usuário seleciona opção do Menu “Movimento / Controle Sistema /”.
2. Solicita o sub-menu desejado (Arquivos Lógicos Internos, Arquivos de Interface Externa, Consultas Externas, Entradas Externas, Saídas Externas, Fator Ajuste).
3. Sistema solicita que o campo (nome do Sistema) seja selecionado.
4. Usuário seleciona o campo.

5. Sistema solicita que os campos necessários sejam preenchidos.
6. Usuário preenche os campos.
7. Os dados são gravados no banco de dados.

Altera Dados Cadastrados / Movimento Sistema { Alternativo }.

Poderá alterar os dados cadastrados.

Exclui Movimento / Sistema

O sistema deverá permitir a exclusão do movimento, solicitando confirmação de exclusão.

RF05- Movimento de Variantes FPA.

### ***Cenários***

Movimento de Variantes FPA { Principal }.

- 1) Analista seleciona opção do Menu “Movimento / Contagem Indicativa e / ou Estimativa Percentual”.
- 2) Sistema solicita que o campo (nome do Sistema) seja selecionado.
- 3) Analista seleciona o campo.
- 4) Analista informa os campos desejados
- 5) Os dados são gravados no banco de dados.

Altera Dados Cadastrados / Movimento Sistema { Alternativo }.

Poderá alterar os dados cadastrados.

Exclui Movimento / Sistema

O sistema deverá permitir a exclusão da movimentação efetuada, solicitando confirmação de exclusão.

RF06- Solicitar Relatório Resumo da Classificação das Funções

Solicitar Relatório Resumo da Classificação das Funções{ Principal }.

### ***Cenários***

- 1) Analista seleciona opção do Menu “Relatórios / Resumo da Classificação das Funções”
- 2) Analista seleciona o sistema a ser comparado.
- 3) Analista clica na opção “imprimir”.

- 4) O sistema deverá listar o total de complexidades detalhadas, que foram informados para o devido sistema. Também deverá listar o total do custo para desenvolvimento do sistema, bem como a linguagem utilizada e total de horas para desenvolvimento do sistema. Também deverá permitir a impressão do relatório.

#### RF07- Gera Relatórios – “Fluxo de Orçamento”

##### ***Cenários***

##### Relatórios – Emite Relatório Fluxo de Orçamento {Principal}.

- 1) Analista seleciona opção do Menu “Relatório / Fluxo de Orçamento”.
- 2) Analista deverá escolher o sistema a ser feito a comparação entre o total de PF do FPA e suas variações CI e MEP.
- 3) Analista deverá clicar no botão imprimir.
- 4) Sistema deverá listar gráfico comparativo entre os totais dos PF’s dos itens mencionados anteriormente.
- 5) O sistema permitirá a impressão do relatório

#### RF09- Solicitar Relatório – “Linguagem x Custo Projeto”

##### ***Cenários***

##### Relatórios – Solicitar Relatório Linguagem x Custo Projeto {Principal}.

- 1) Analista seleciona opção do Menu “Relatórios / Linguagem x Custo Projeto”
- 2) Analista seleciona o sistema a ser comparado.
- 3) Analista seleciona o checkbox contagem indicativa ou método de estimativa percentual para efetuar comparação de estimativa de custo por linguagem de programação.
- 4) O analista deverá clicar no botão imprimir.
- 5) O sistema deverá listar as linguagens de programação cadastradas no sistema com suas devidas informações (como valor hora, tempo para desenvolvimento de PF), e efetuar comparação de valor total entre MEP e / ou CI.
- 6) Com bases nas informações do item anterior o usuário será capaz de verificar qual linguagem de programação utilizar para o determinado sistema a ser desenvolvido.

7) Sistema deverá permitir a impressão do relatório

RF09- Gera Relatório Contratual

***Cenários***

Relatórios – Gera Relatório / Contratual {Principal}.

- 1) Analista seleciona opção do Menu “Relatório”
- 2) O analista deverá selecionar o software a ser listado para proposta
- 3) O analista deverá selecionar a opção de imprimir o relatório (para visualizar em vídeo as informações da proposta), ou enviar por e-mail (sendo que o sistema irá enviar o relatório da proposta por e-mail).