

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE CIÊNCIAS DA COMPUTAÇÃO
(Bacharelado)

**PROTÓTIPO DE UMA FERRAMENTA PARA GERÊNCIA DE
CUSTOS EM PROJETOS DE SOFTWARE BASEADA NO
MODELO PMBOK**

TRABALHO DE CONCLUSÃO DE CURSO SUBMETIDO À UNIVERSIDADE
REGIONAL DE BLUMENAU PARA A OBTENÇÃO DOS CRÉDITOS NA
DISCIPLINA COM NOME EQUIVALENTE NO CURSO DE CIÊNCIAS DA
COMPUTAÇÃO — BACHARELADO

LINDOLFO PEREIRA JUNIOR

BLUMENAU, JUNHO/2003.

2003/1-43

PROTÓTIPO DE UMA FERRAMENTA PARA GERÊNCIA DE CUSTOS EM PROJETOS DE SOFTWARE BASEADA NO MODELO PMBOK

LINDOLFO PEREIRA JUNIOR

ESTE TRABALHO DE CONCLUSÃO DE CURSO FOI JULGADO ADEQUADO
PARA OBTENÇÃO DOS CRÉDITOS NA DISCIPLINA DE TRABALHO DE
CONCLUSÃO DE CURSO OBRIGATÓRIA PARA OBTENÇÃO DO TÍTULO
DE:

BACHAREL EM CIÊNCIAS DA COMPUTAÇÃO

Prof. Marcel Hugo — Orientador na FURB

Prof. José Roque Voltolini da Silva — Coordenador do TCC

BANCA EXAMINADORA

Prof. Marcel Hugo

Prof. Everaldo Artur Grahl

Prof. Ricardo Alencar de Azambuja

AGRADECIMENTOS

Agradeço, primeiramente, a Deus, por estar presente em todos os momentos e me iluminando nos momentos de dificuldade.

Ao meu orientador Marcel Hugo, pela orientação, atenção e apoio dispensados na elaboração deste trabalho.

Aos meus pais, que me apoiaram durante toda a minha vida e sem os quais eu não teria chegado a este momento.

À minha noiva Érica, pelo apoio, incentivo, compreensão e carinho que recebi durante a elaboração deste trabalho, principalmente nos momentos mais difíceis.

Finalmente, agradeço a todos que de alguma forma contribuíram para elaboração deste trabalho.

RESUMO

Este trabalho tem por objetivo um estudo da aplicação dos conceitos de gerenciamento de projetos de software definidos pelo PMBOK – *Project Management Body of Knowledge* – no âmbito da gerência de custos de projetos, aplicando também a técnica de Análise de Pontos por Função (FPA) como métrica. Para demonstrar alguns destes conceitos, foi desenvolvido o protótipo de um sistema de gerenciamento de custos em projetos de software, que atende aos padrões estabelecidos pela área de gerência de custos do PMBOK.

ABSTRACT

This work has for objective a study of the application of the concepts of management of defined software projects for PMBOK - Project Management Body of Knowledge - in the ambit of the management of costs of projects, also applying the technique of Analysis of Points for Function (FPA) as metric. To demonstrate some of these concepts, the prototype of a system of management of costs was developed in projects of software, that assists to the patterns established by the area of management of costs of PMBOK.

SUMÁRIO

| | |
|--|----|
| LISTA DE FIGURAS | 7 |
| LISTA DE TABELAS | 9 |
| LISTA DE ABREVIATURAS..... | 10 |
| 1 INTRODUÇÃO..... | 11 |
| 1.1 OBJETIVOS | 14 |
| 1.2 ESTRUTURA DO TRABALHO..... | 14 |
| 2 GERENCIAMENTO DE PROJETOS | 16 |
| 2.1 PROJETO..... | 17 |
| 2.2 MODELOS E NORMAS..... | 18 |
| 2.3 PROJECT MANAGEMENT INSTITUTE..... | 19 |
| 2.4 PMBOK – PROJECT MANAGEMENT BODY OF KNOWLEDGE..... | 20 |
| 2.5 FASES E CICLO DE VIDA DE UM PROJETO | 20 |
| 2.6 PROCESSOS | 22 |
| 2.6.1 PROCESSOS DE INICIAÇÃO | 23 |
| 2.6.2 PROCESSOS DE PLANEJAMENTO | 24 |
| 2.6.3 PROCESSOS DE EXECUÇÃO | 26 |
| 2.6.4 PROCESSOS DE CONTROLE..... | 27 |
| 2.6.5 PROCESSOS DE ENCERRAMENTO | 28 |
| 2.7 ÁREAS DE CONHECIMENTO OU DE ATUAÇÃO GERENCIAL..... | 28 |
| 2.7.1 GERÊNCIA DE CUSTOS..... | 30 |
| 2.7.1.1 PLANEJAMENTO DOS RECURSOS..... | 32 |
| 2.7.1.2 ESTIMATIVA DOS CUSTOS | 33 |
| 2.7.1.3 ORÇAMENTAÇÃO DOS CUSTOS | 34 |
| 2.7.1.4 CONTROLE DOS CUSTOS | 35 |

| | |
|--|----|
| 2.8 MÉTRICAS NO PROCESSO DE GERENCIAMENTO DE PROJETOS DE SOFTWARE..... | 36 |
| 2.9 ANÁLISE DE PONTOS DE FUNÇÃO | 38 |
| 2.9.1 COMPONENTES BÁSICOS DA ANÁLISE DE PONTOS DE FUNÇÃO..... | 39 |
| 2.9.2 ARQUIVO LÓGICO INTERNO..... | 40 |
| 2.9.3 ARQUIVO DE INTERFACE EXTERNA | 40 |
| 2.9.4 ENTRADA EXTERNA..... | 41 |
| 2.9.5 SAÍDA EXTERNA..... | 42 |
| 2.9.6 CONSULTA EXTERNA..... | 42 |
| 2.9.7 ITENS DE INFLUÊNCIA | 43 |
| 2.10 TRABALHOS CORRELATOS..... | 45 |
| 3 DESENVOLVIMENTO DO PROTÓTIPO | 46 |
| 3.1 REQUISITOS PRINCIPAIS DO PROBLEMA | 46 |
| 3.2 ESPECIFICAÇÃO | 46 |
| 3.2.1.1 DIAGRAMA DE CASOS DE USO..... | 46 |
| 3.2.1.2 DIAGRAMA DE CLASSES..... | 51 |
| 3.2.1.3 DIAGRAMA DE PACOTES | 52 |
| 3.2.1.4 DIAGRAMAS DE SEQÜÊNCIA..... | 52 |
| 3.3 IMPLEMENTAÇÃO | 56 |
| 3.4 OPERACIONALIDADE DA IMPLEMENTAÇÃO..... | 58 |
| 4 CONCLUSÕES | 67 |
| 4.1 EXTENSÕES | 68 |
| ANEXO 1 – EXEMPLO DE CÓDIGO FONTE..... | 69 |
| ANEXO 2 – EXEMPLOS DE RELATÓRIOS DISPONÍVEIS NO PROTÓTIPO. | 75 |
| REFERÊNCIAS BIBLIOGRÁFICAS | 80 |

LISTA DE FIGURAS

| | |
|--|----|
| Figura 1 – Fases de um projeto de desenvolvimento de software..... | 21 |
| Figura 2 – Relacionamento entre grupos de processos..... | 22 |
| Figura 3 – Interação entre as fases do projeto. | 23 |
| Figura 4 – Áreas de gerenciamento de projetos segundo o PMBOK..... | 30 |
| Figura 5 – Visão Geral da Gerência de custos do projeto. | 31 |
| Figura 6 – Visão Geral Planejamento dos Recursos do Projeto..... | 33 |
| Figura 7 – Visão Geral da Estimativa dos Custos. | 34 |
| Figura 8 – Orçamentação dos custos do projeto..... | 35 |
| Figura 9 – Processos de controle dos custos..... | 36 |
| Figura 10 – Análise de Pontos de Função (FPA). | 39 |
| Figura 11 – Diagrama de casos de uso Primários..... | 48 |
| Figura 12 – Diagrama de casos de uso secundários. | 49 |
| Figura 13 – Diagrama de casos de uso – Especialização de Emite Relatórios Gerenciais..... | 50 |
| Figura 14 – Diagrama de classes. | 51 |
| Figura 15 – Diagrama de Pacotes..... | 52 |
| Figura 16 – Diagrama de Seqüência Emite Relatório de Pontos de Função. | 52 |
| Figura 17 – Diagrama de Seqüência Calcula Pontos de Função. | 53 |
| Figura 18 – Diagrama de Seqüência Aloca Recursos..... | 53 |
| Figura 19 – Diagrama de Seqüência Emite Relatório de Alocação de Recursos..... | 54 |
| Figura 20 – Diagrama de Seqüência Emite Relatório do Gráfico de Custos. | 54 |
| Figura 21 – Diagrama de Seqüência Emite Relatório da Estrutura Analítica de Trabalho..... | 55 |
| Figura 22 – Diagrama de Seqüência Emite Relatório da Descrição do Quadro de Recursos. | 55 |

| | |
|--|----|
| Figura 23 – Exemplo de herança no Visual Basic.NET..... | 56 |
| Figura 24 – Diagrama Entidade-Relacionamento. | 57 |
| Figura 25 – Cadastro de Projetos..... | 58 |
| Figura 26 – Cadastro de Metodologias..... | 59 |
| Figura 27 – Cadastro de Gerentes..... | 60 |
| Figura 28 – Cadastro de Sistemas. | 60 |
| Figura 29 – Cadastro de Atividades. | 61 |
| Figura 30 – Cadastro de Recursos. | 62 |
| Figura 31 – Alocação de Recursos. | 62 |
| Figura 32 – Classificação das Funções..... | 63 |
| Figura 33 – Características Gerais do Sistema. | 63 |
| Figura 34 – Cálculo de Pontos de Função. | 64 |
| Figura 35 – Visualização de Relatórios..... | 65 |
| Figura 36 – Gráfico de Custos..... | 65 |
| Figura 37 – Itens da Gerência de Custos do Projeto abordados pelo protótipo..... | 66 |

LISTA DE TABELAS

| | |
|---|----|
| Tabela 1 – Evolução da indústria de software americana desde 1994. | 12 |
| Tabela 2 – Complexidade de arquivos lógicos internos. | 40 |
| Tabela 3 – Complexidade de arquivos de interface externa. | 41 |
| Tabela 4 – Complexidade de entradas externas. | 41 |
| Tabela 5 – Complexidade de saídas externas. | 42 |
| Tabela 6 – Complexidade de consultas externas. | 43 |

LISTA DE ABREVIATURAS

| | |
|--------|---|
| CASE | - <i>Computer Aided Software Engineering</i> |
| COCOMO | - <i>Constructive Cost Model</i> |
| CMM | - <i>Capability Model Maturity</i> |
| EAP | - <i>Estrutura Analítica do Projeto</i> |
| FPA | - <i>Function Point Analysis</i> |
| IFPUG | - <i>International Function Point User Group</i> |
| ISO | - <i>International Organization for Standardization</i> |
| MDS | - <i>Metodologia de Desenvolvimento de Sistemas</i> |
| PMBOK | - <i>Project Management Body of Knowledge</i> |
| PMI | - <i>Project Management Institute</i> |
| SDO | - <i>Software Development Organizations</i> |
| SEI | - <i>Software Engineering Institute</i> |
| UML | - <i>Unified Modeling Language</i> |

1 INTRODUÇÃO

A Engenharia de Software tem por finalidade auxiliar na construção de software da melhor maneira possível (Pressman, 1992). Desde os anos 60, quando a frase “*The Software Crisis*” foi pronunciada, muitos problemas desta área foram identificados, e muitos deles ainda persistem, tais como (Gibbs, 1994):

1. Previsão pobre – desenvolvedores não prevêm adequadamente quanto tempo e esforço serão necessários para produzir um sistema de software que satisfaça as necessidades (requisitos) dos clientes/usuários. Sistemas de software são geralmente entregues muito tempo depois do que fora planejado;
2. Programas de baixa qualidade – programas de software não executam o que o cliente deseja, conseqüência talvez da pressa para se entregar o produto. Os requisitos originais podem não ter sido completamente especificados, ou podem apresentar contradições, e isto pode ser descoberto muito tarde durante o processo de desenvolvimento;
3. Alto custo para manutenção – a manutenção pode ser corretiva, quando ocorrem enganos (erros, falhas) no sistema já entregue, ou evolutiva, quando novas características são adicionadas ao sistema de software. Ambas são caras quando o sistema original foi construído sem uma arquitetura clara e visível;
4. Duplicação de esforços – é difícil compartilhar soluções ou reusar código, em função das características de algumas linguagens adotadas, por falta de confiança no código feito por outra pessoa e até mesmo pela ausência/deficiência de documentação das rotinas e dos procedimentos já construídos.

Para solucionar alguns destes problemas, muitas empresas de desenvolvimento de software têm adotado metodologias de desenvolvimento de software. Os ambientes tradicionais das empresas geralmente têm suportado somente a engenharia do produto, assumindo um processo implícito e tendo como foco principal o produto. Esta visão tem limitado as empresas no que diz respeito à tomada de decisões, ao estabelecimento e arquivamento de metas organizacionais, à determinação de pontos para melhoria, à estipulação de prazos para entrega de produtos e à obtenção de uma certificação.

De acordo com Paulk (1993), alguns estudos e pesquisas que foram realizados nos anos 90 demonstraram que o gerenciamento de projeto é a causa mais evidente das falhas na execução e entrega de produtos e serviços de software. O *Software*

Engineering Institute (SEI) constatou, já em 1993, que o principal problema que aflige as organizações de software é gerencial e preconizou que: “as organizações precisam vencer o seu buraco negro que é o seu estilo de gerenciar de maneira informal, sem métodos e sem técnicas”.

O estudo realizado por Standish (1995), chamado de ”Relatório do Caos”, focou a indústria de software comercial. Esse estudo identificou que: nas empresas dos Estados Unidos, 31% dos projetos estudados foram cancelados antes de estarem concluídos; 53% dos projetos de software que foram concluídos excederam mais do que 50% a sua estimativa de custo; somente 16% dos projetos, em grandes empresas, foram entregues no tempo e orçamento.

Estudos mais recentes realizados por Standish (2000), revelam que: apenas 28% dos projetos de software são completados dentro da sua estimativa de tempo e com todas as características e funções originalmente especificadas; 49% dos projetos são completados acima das estimativas de tempo e orçamento, e ainda com menos características e funções que inicialmente especificados no projeto, 23% dos projetos de software são cancelados antes da conclusão ou nunca são implementados. A evolução dos projetos de software nos Estados Unidos no período de 1994 a 2000 pode ser verificada na tabela 1. Novamente o estudo indicou o gerenciamento do projeto como a principal causa para o sucesso ou fracasso de um projeto de software.

Tabela 1 – Evolução da indústria de software americana desde 1994.

| | 1994 | 1996 | 1998 | 2000 |
|----------------------------|-------------|-------------|-------------|-------------|
| Obtiveram sucesso | 16% | 27% | 26% | 28% |
| Foram Cancelados | 31% | 40% | 28% | 23% |
| Excederam os custos | 53% | 33% | 46% | 49% |

FONTE: Standish (2000).

Jones (1999) destaca que a ausência de um processo de gerenciamento apropriado, aliado às estimativas deficientes de custo e de tempo, é uma das principais causas das falhas dos projetos de software.

Apesar disso, o gerenciamento de projetos de software ainda é pouco abordado e praticado nas SDO. Diante disto, a utilização de procedimentos padronizados e de fácil

compreensão pode ser de grande valor, fornecendo uma orientação aos gerentes de projeto e dificultando a ocorrência de falhas graves de gerenciamento por falta de experiência.

O objetivo do gerenciamento de projetos é assegurar que processos particulares sejam seguidos, coordenando e monitorando as atividades da engenharia do produto. Um processo de gerenciamento de projeto deve identificar, estabelecer, coordenar e monitorar as atividades, as tarefas e os recursos necessários para um projeto produzir um produto e/ou serviço de acordo com seus requisitos.

Segundo Martins (2002), o *Project Management Institute* (PMI), entidade internacional sem fins lucrativos que congrega os profissionais que atuam em áreas relacionadas à Gerência de Projetos, é pioneiro na regulamentação e distribuição do conhecimento referente à disciplina de gerência de projetos. O PMI especificou alguns procedimentos que visam padronizar a teoria associada à gerência de projetos. A teoria de gestão definida pelo PMI está registrada em um documento chamado *Project Management Body of Knowledge* (PMBOK).

O PMBOK define algumas áreas de conhecimento da gerência de projetos, que descrevem os conhecimentos e práticas em gerência de projetos em termos dos processos que as compõe. Uma delas é a Gerência de Custos, que inclui os processos necessários para assegurar que o projeto possa ser executado dentro do orçamento aprovado. Esta área engloba o planejamento de recursos, as estimativas de custos dos recursos, a confecção do orçamento e o controle de custos.

DeMarco (1991) afirma “que não se pode controlar o que não se pode medir” e comenta que, apesar de ser óbvio o forte elo entre medição e controle, é uma idéia nova para a maioria dos gerentes de software que mantém a ilusão de que se consegue levar adiante um projeto, sob total controle, sem nunca terem medido coisa alguma. Segundo Braga (1996), através do uso de métricas, a área de informática passa a ter números para apresentar, seja em resultados seja em metas de desempenho.

De acordo com Silva (2000), um software é desenvolvido para realizar determinadas funções. Determinar estas funções segundo critérios precisos e dar-lhes uma medida é o ponto-chave da Análise de Pontos de Função. A Análise de Pontos de Função pretende fornecer uma unidade de medida para a indústria de software, independente de plataforma ou linguagem. Sendo assim, pode ser facilmente utilizada e

compreendida por profissionais de informática, e utilizada para auxiliar no gerenciamento e mensuração do tamanho funcional de um software.

As ferramentas determinam algum apoio automatizado para os procedimentos do processo. Com o intuito de fornecer meios para este apoio automatizado, este trabalho desenvolve um protótipo de ferramenta para o gerenciamento e acompanhamento de projetos de software. Esta ferramenta fornece a estrutura para que uma SDO possa de forma automatizada gerenciar os recursos necessários para que um projeto possa produzir um produto e/ou serviço de acordo com seus requisitos, controlando todos os custos relativos ao desenvolvimento do mesmo.

1.1 OBJETIVOS

O presente trabalho tem como objetivo o desenvolvimento de um protótipo de ferramenta para gerência de custos em projetos de software baseada no modelo PMBOK.

Os objetivos específicos são: gerenciamento das estimativas de custo em projetos de software através do acompanhamento das atividades do projeto, controle dos recursos alocados e geração de relatórios de acompanhamento dos custos do projeto.

1.2 ESTRUTURA DO TRABALHO

O presente trabalho está estruturado em 4 capítulos, sendo que o capítulo 1 apresenta uma introdução ao gerenciamento de projetos, sua origem, relevância e justificativa, os objetivos principais e específicos e a estrutura do mesmo.

No capítulo 2 apresenta-se uma fundamentação ao gerenciamento de projetos, descrevendo os processos definidos pelo PMI em seu modelo de gerenciamento, o PMBOK. Neste capítulo também são detalhados os processos definidos na área de gerência de custos do PMBOK. Em seguida são descritas as principais características da Análise por Pontos de Função como ferramenta de métricas de software.

No capítulo 3 mostra-se o que foi implementado no sistema, o que se propõe e que técnicas foram utilizadas. Define-se a modelagem do sistema em questão, identificando seus processos através da análise orientada a objetos e também os dados presentes no sistema, além da implementação do sistema em si, mostrando suas telas e

detalhando-as, identificando onde estão os conceitos de gerenciamento de custos do projeto descritos pelo PMBOK.

Para finalizar, no capítulo 4 encontram-se conclusões e extensões para trabalhos futuros.

2 GERENCIAMENTO DE PROJETOS

O Gerenciamento de Projetos (ou Gerência de Projetos) é um ramo das Ciências Gerenciais que trata do planejamento e controle de projetos. Gerenciar um projeto significa, resumidamente, planejar sua execução antes de iniciá-lo e, a seguir, acompanhar a sua execução (Prado, 2000). O PMI (2002, p. 6) define que “gerenciamento de projetos é aplicação de conhecimentos, experiências, ferramentas e técnicas às atividades do projeto a fim de atingir ou exceder as expectativas dos interessados/envolvidos”.

De acordo com Martins (2002), a ciência do gerenciamento de projetos surgiu no final da década de 50 e tem evoluído muito desde então. Para diversas organizações, seus negócios dependem da sua capacidade de planejar e executar projetos com eficiência. Durante anos, a disciplina de gerência de projetos foi utilizada principalmente na indústria aeroespacial e na engenharia e construção, mas agora seus conceitos vêm sendo aplicados em diversas áreas. As empresas de desenvolvimento de software estão neste grupo.

Segundo Prado (2000), na década de 70, quando as organizações de desenvolvimento de software acreditavam que, para conter a imensa desorganização existente, era necessário um sistema rígido de planejamento e acompanhamento de projetos, foram desenvolvidas as Metodologias de Desenvolvimento de Sistemas (MDS). As experiências iniciais mostraram que as MDS não diminuían os prazos e ainda se transformaram em aniquiladoras da criatividade. A partir de meados da década de 80, as MDS passaram a ser bastante flexíveis, e seu uso amadureceu, mas a má fama retardou uma maior difusão da gerência de projetos na indústria da informática.

No planejamento do projeto são estabelecidas as metas, o escopo, a identificação das tarefas a serem realizadas e a sua seqüência baseada nos recursos necessários e disponíveis. O controle do projeto, no sentido moderno do termo, significa a medição do progresso e do desempenho através de um sistema ordenado preestabelecido. Ações corretivas são tomadas sempre que necessárias.

De acordo com Presmann (1995), o gerenciamento de projetos é a primeira atividade do processo de engenharia de software e abrange todo o processo de desenvolvimento, do começo ao fim, e oferece a compreensão do escopo do trabalho a

ser realizado, os riscos em que se incorrerá, os recursos exigidos, as tarefas a serem executadas, os marcos de referência a serem acompanhados, o esforço despendido, na forma de custo, e a programação a ser seguida.

As vantagens advindas de um projeto bem administrado se resumem, basicamente, em que a execução não diferirá significativamente do planejamento. Um bom planejamento implica que um projeto poderá ser executado com prazo e custos adequados e também com a melhor qualidade possível.

Finalmente, o gerenciamento de projetos, segundo Presmann (1995), compreende atividades que envolvem, principalmente: medição e estimativa, análise dos riscos e determinação de um cronograma, além do monitoramento e controle dessas atividades.

2.1 PROJETO

Segundo Martins (2002), projeto significa “empreendimento”, e como tal é um trabalho que visa a criação de um produto ou a execução de um serviço específico, temporário, não repetitivo e que envolve um certo grau de incerteza na realização. O PMI (2002) define projeto como sendo um esforço temporário, levado a efeito para criar um produto/serviço único.

Por serem temporários, os projetos têm um momento de início e um fim estimado e as atividades diárias geralmente representam um desafio novo, visto que, geralmente não foram executadas anteriormente. Como em qualquer empreendimento, as atividades precisam ser planejadas, programadas e, durante a execução, precisam ser controladas.

Existem projetos de tipos bastante variados e alguns deles têm semelhanças com processos rotineiros. Pode-se citar algumas características típicas, que distinguem projetos de um processo repetitivo (Prado, 2000):

1. Projetos possuem um ciclo de vida constituído por etapas distintas que, por sua vez, se dividem em tarefas ou atividades, muitas delas executadas uma única vez. Na produção diária, as mesmas atividades são executadas todos os dias.
2. Em projetos é difícil o estabelecimento de padrões de produtividade, diferentemente do gerenciamento da rotina, que se baseia fundamentalmente

nestes padrões e os controla, pelo estabelecimento de procedimentos padronizados e de itens de controle.

3. Em um projeto existe um seqüenciamento das atividades, cujo planejamento é crucial para que o projeto se encerre em um prazo e custo apropriados.
4. Em um projeto existe um constante “entra e sai” de pessoas; às vezes, um profissional é “emprestado” a um determinado projeto por um certo tempo.

As características acima citadas são muito genéricas, e tem por objetivo traçar um ponto de partida para um estudo mais aprofundado do gerenciamento de projetos. Alguns projetos rotineiros costumam ser gerenciados através do uso de técnicas de gerenciamento de projetos: chama-se isto de gerenciamento por projetos.

A partir da década de noventa, percebeu-se que os conceitos “projeto” e “operações rotineiras” não eram completamente isolados. Inúmeros processos, que anteriormente eram tratados como “rotinas”, passaram a ser tratados como “projetos” nas empresas. Um novo termo foi introduzido: gerenciamento por projetos, para significar a aplicação das técnicas e métodos de gerenciamento de projetos em alguns tipos de operações rotineiras que têm como características produzir um produto/serviço único em um determinado prazo (PMI, 2002).

2.2 MODELOS E NORMAS

Existem diversas organizações mundiais e nacionais que agregam profissionais preocupados com as questões envolvidas com o gerenciamento de projetos.

A estratégia adotada pelo SEI (*Software Engineering Institute*) foi desenvolver práticas na disciplina de engenharia de software e preparar indivíduos e organizações para usar e evoluir estas práticas. Como exemplo, tem-se o desenvolvimento do modelo de capacidade e maturidade de software (CMM). Envolve o escopo do CMM apenas a área de software, não sendo usado como modelo de estrutura comercial, marketing, finanças ou até mesmo, áreas da informática como hardware e banco de dados (Paula Filho, 2001).

De acordo com ISO (1999), a ISO (*International Organization for Standardization*) é uma organização não governamental fundada em 1947, e sua missão é promover o desenvolvimento da padronização e atividades correlatas no mundo com

vistas a facilitar a troca internacional de bens e serviços e desenvolver de forma cooperativa as esferas intelectuais, científicas, tecnológicas e econômicas internacionais.

A norma ISO 10006 (ISO, 1997) se refere a atividades de gerenciamento da qualidade no tocante ao gerenciamento de projetos. Provê guias para os elementos do sistema de qualidade, conceitos e práticas para aprimorar e garantir a qualidade no processo de gerenciamento de projetos, sendo suplementar à norma ISO 9004. Foi desenvolvida para ser aplicada ao gerenciamento de projetos de qualquer porte ou complexidade, servindo de base para que profissionais de gerenciamento de projetos e auditores de qualidade possam trocar experiências a respeito do projeto.

A norma internacional NBR ISO/IEC 12207 – Tecnologia da Informação – Processos de Ciclo de Vida de Software define os processos e atividades básicos que compõe o processo de desenvolvimento de software. Essa norma é flexível e tem como objetivos principais: auxiliar os envolvidos na produção de software a definir seus papéis, por meio de processos bem definidos; e permitir um melhor entendimento das atividades a serem executadas nas operações que envolvam, de alguma forma, o software (Rocha, 2001).

2.3 PROJECT MANAGEMENT INSTITUTE

O *Project Management Institute* (PMI) é pioneiro na regulamentação e distribuição dos conhecimentos referentes à gestão de projetos. O PMI é uma entidade internacional sem fins lucrativos que congrega os profissionais que atuam em áreas relacionadas à Gerência de Projetos (*Project Management*). Foi fundado em 1969 nos EUA e hoje está presente em todo o mundo, incluindo o Brasil (Martins, 2002).

Seu principal compromisso é promover o profissionalismo e a ética em gestão de projetos. Além disso, disponibiliza aos seus associados produtos e serviços, estabelecendo a aceitação do gerenciamento de projetos como uma disciplina e uma profissão.

O PMI possui publicações (chamadas de capítulos) em quase todos os países, e sua principal publicação, o PMBOK (*Project Management Body of Knowledge*) é mundialmente conhecido. O PMBOK é um guia de orientação para os profissionais

sobre o conhecimento em gerenciamento de projetos. Seu propósito é identificar e descrever conceitos e práticas do gerenciamento, padronizando a terminologia utilizada.

2.4 PMBOK – PROJECT MANAGEMENT BODY OF KNOWLEDGE

O “*Project Management Body of Knowledge Guide*” ou “Guia para o Universo do Conhecimento de Gerenciamento de Projetos”, mais conhecido como “PMBOK Guide”, é considerado um marco na história do gerenciamento de projetos. Ele é de autoria do *Standards Committee* (Comitê de Padronização) do *Project Management Institute*, e procura contemplar os principais aspectos que podem ser abordados no gerenciamento de um projeto genérico. Não se trata de uma metodologia de gerenciamento de projetos e, sim, de uma padronização, identificando e nomeando processos, áreas de conhecimento, técnicas, regras e métodos. Ele foi reconhecido, em 1999, como um padrão de gerenciamento de projetos pelo ANSI – *American National Standards Institute*.

O PMI (2000) pressupõe que, durante o ciclo de vida, há o envolvimento com o gerenciamento do trabalho e como gerenciamento do produto (bem ou serviço que está sendo desenvolvido). Ele analisa o gerenciamento do trabalho como sendo fruto da seguinte ótica:

- a) Divisão do trabalho em etapas (ciclo de vida);
- b) Em cada etapa ocorrem processos;
- c) Em cada processo são executadas ações gerenciais que podem abranger até nove áreas de conhecimento.

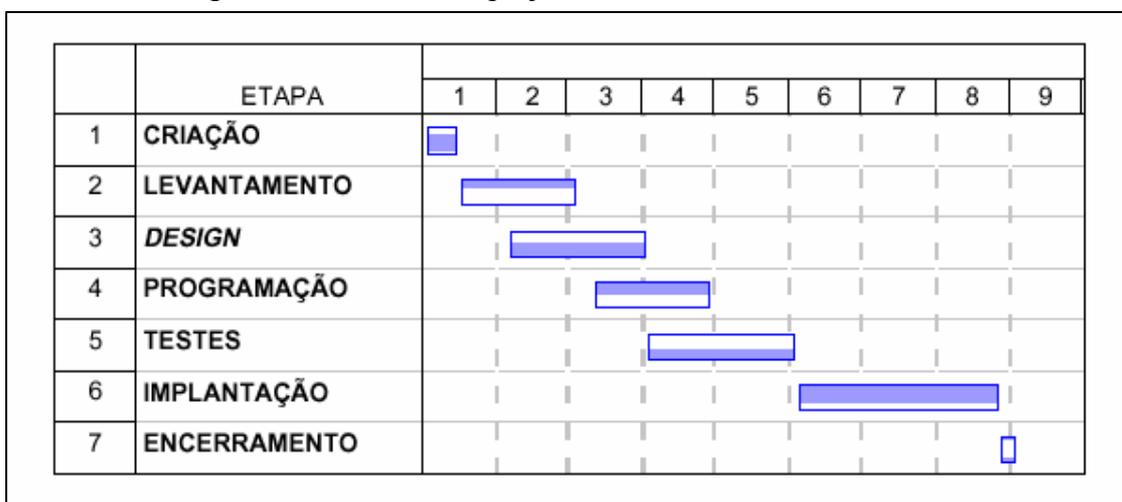
2.5 FASES E CICLO DE VIDA DE UM PROJETO

As organizações que desenvolvem projetos normalmente dividem-nos em várias fases ou etapas, visando um maior controle gerencial e uma ligação mais adequada de cada projeto com seus processos internos. O conjunto de suas fases é conhecido como “*ciclo de vida do projeto*”. Segundo Prado (2000), um ciclo de vida é caracterizado por fases distintas, sendo que genericamente, pode-se citar as seguintes:

- a) Na fase de Concepção, ocorre o início formal do projeto; são elaboradas as versões iniciais de cronogramas e orçamentos; é preparada a proposta inicial; e é nomeada a equipe do projeto.
- b) Na fase de Desenvolvimento os documentos de orçamento e cronograma são elaborados em detalhes; desenvolve-se o protótipo; são realizados os testes dos protótipos e são distribuídas as tarefas para a equipe do projeto.
- c) Na fase de Execução, os planos definidos na fase de Desenvolvimento são executados.
- d) Na fase de Conclusão, o produto ou serviço desenvolvido é entregue, e também é feita a revisão e arquivamento de todos os documentos relativos ao projeto.

Durante a montagem de um projeto, geralmente são elaboradas uma seqüência de fases ou etapas baseadas na Estrutura Analítica do Projeto (EAP) ou Estrutura de Decomposição do Projeto (*Work Breakdown Structure*). Por exemplo, em projetos de desenvolvimento de sistemas, as etapas poderiam ser as seguintes: Criação, Levantamento de Requisitos, Projeto, Programação, Testes e Encerramento, conforme ilustrado na figura 1. Ao final de cada uma delas, são produzidos um ou mais produtos ou serviços, tais como o “Manual de Especificações de um Software”, por exemplo. Uma EAP de um projeto anterior pode ser reutilizada como modelo em um outro projeto. Embora cada projeto seja único, EAP’s podem ser reutilizadas, devido ao fato que a maioria dos projetos se assemelham um ao outro.

Figura 1 – Fases de um projeto de desenvolvimento de software.



FONTE: Prado (2000)

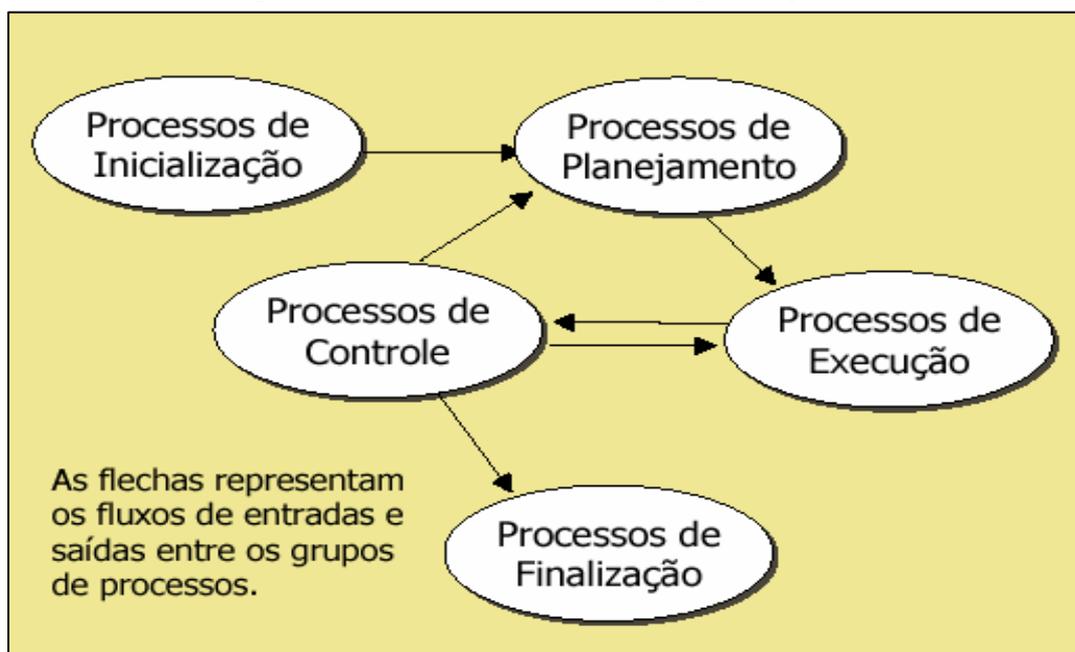
2.6 PROCESSOS

Segundo o PMI (2002), os projetos são compostos por processos. Os processos são realizados por pessoas, e normalmente se enquadram em uma das duas categorias:

- a) Processos da gerência de projetos: relacionam-se com a descrição, a organização e a conclusão do trabalho do projeto.
- b) Processos orientados ao produto: relacionam-se com a especificação e a criação do produto do projeto.

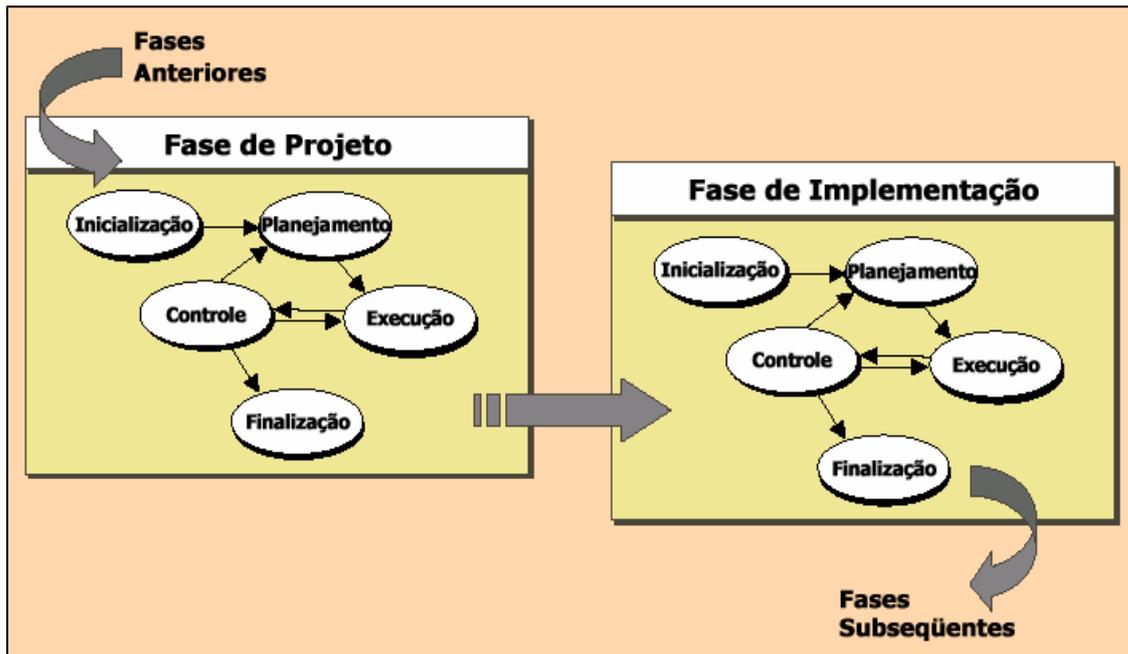
Em cada fase do projeto são executados diversos processos com o objetivo de produzir o resultado esperado daquela fase. Conforme padronização do PMI (*Project Management Institute*), estes processos se enquadram nos seguintes grupos: Processos de Inicialização, Processos de Planejamento, Processos de Execução, Processos de Controle e Processos de Encerramento, que ocorrem dentro de cada fase e estão interligados, conforme ilustrado nas figuras 2 e 3. Assim, os resultados das ações tomadas durante o processo de inicialização são utilizados como entrada para as ações a serem tomadas durante o processo de planejamento. Todos os processos ocorrem em todas as fases do projeto, mas, dependendo da fase, há uma incidência maior de alguns tipos de processos.

Figura 2 – Relacionamento entre grupos de processos.



FONTE: PMI (2002).

Figura 3 – Interação entre as fases do projeto.



FONTE: PMI (2002).

2.6.1 PROCESSOS DE INICIAÇÃO

O processo chamado de Iniciação pelo PMBOK marca o nascimento do projeto ou fase do projeto. É neste momento que são definidos o esboço do escopo do projeto ou serviço, que será entregue no final do projeto, e ao gerente do mesmo. Nesta etapa é importante obter o comprometimento da organização para o início da fase ou projeto.

Ao final deste processo os seguintes elementos deverão estar definidos:

- a) Termo de referência: descrição clara do produto ou serviço a ser executado e justificativa do negócio.
- b) Designação do gerente do projeto: definição do gerente do projeto, através de um documento formal.
- c) Identificação das restrições do projeto: geralmente impostas pelo cliente, como custo, prazo, pessoal, necessidades tecnológicas, etc.
- d) Identificação das premissas: definidas pela equipe de projeto, aumentam o risco do projeto, caso não sejam respeitadas.

Durante este processo também deve ser definida pela equipe qual é a sua missão. A missão é utilizada para estabelecer metas, objetivos e tomar decisões. Ela

deve responder a três questões: O que vamos fazer? Para quem vamos produzir? Como será produzido?

Os objetivos serão descritos com base na declaração da missão. São mais específicos, informando resultados que devem ser obtidos para que a missão seja cumprida com êxito.

2.6.2 PROCESSOS DE PLANEJAMENTO

O processo de planejamento tem uma importância fundamental no desenvolvimento de um projeto, porque executar um projeto implica em realizar algo que não tinha sido feito antes. É neste processo que a equipe do projeto é montada, o escopo é definido, o prazo e o custo são estimados, os riscos são identificados, as ações corretivas são definidas e a forma de comunicação é estabelecida. O planejamento é um processo contínuo durante toda a vida do projeto.

O PMI (2002) define dois tipos de processos de planejamento: Processos Essenciais e Processos Facilitadores.

Processos Essenciais são os processos que são executados na mesma seqüência e tem dependências bem definidas. Por exemplo, as atividades devem ser definidas antes do estabelecimento do cronograma e custos. Estes processos podem interagir várias vezes durante qualquer fase de um projeto. Os Processos Essenciais descritos pelo PMI (2002) são os seguintes:

- a) Planejamento do escopo: desenvolver uma declaração escrita do escopo, como base para futuras decisões no projeto;
- b) Detalhamento do escopo: subdividir os principais subprodutos em componentes menores e mais manuseáveis;
- c) Definição das atividades: identificar as atividades específicas que devem ser realizadas para produzir os principais subprodutos do projeto;
- d) Seqüenciamento das atividades: identificar e documentar as pendências entre as atividades;
- e) Estimativa da duração das atividades: estimar a quantidade de períodos de trabalho que serão necessários para completar as atividades individuais;

- f) Desenvolvimento do cronograma: criar o cronograma do projeto a partir da análise da seqüência das atividades, suas durações e necessidades de recursos;
- g) Planejamento da gerência de risco: decidir como abordar e planejar a gerência de risco no projeto;
- h) Planejamento dos recursos: determinar que recursos devem ser utilizados e em que quantidades para as atividades do projeto;
- i) Estimativa dos custos: desenvolver uma aproximação (estimativa) dos custos dos recursos que são necessários para completar as atividades do projeto;
- j) Orçamento dos custos: alocar a estimativa dos custos globais aos pacotes individuais de trabalho;
- k) Desenvolvimento do plano do projeto: agregar os resultados de outros processos de planejamento construindo um documento coerente e consistente.

Nos Processos Facilitadores as interações entre os demais processos de planejamento são mais dependentes da natureza do projeto. Por exemplo, em alguns projetos pode ter sido identificado apenas um pequeno risco ou mesmo nenhum, até que o planejamento tenha sido concluído e a equipe reconheça que as metas de custo e prazo são muito ousadas, envolvendo assim um custo considerável.

O PMI (2002) descreve os seguintes processos facilitadores:

- a) Planejamento da Qualidade: identificar os padrões de qualidade relevantes para o projeto e determinar ações para executá-los;
- b) Planejamento Organizacional: identificar, documentar, atribuir papéis e relações hierárquicas no projeto;
- c) Montagem da equipe: conseguir que os recursos humanos necessários sejam designados e alocados ao projeto;
- d) Planejamento das Comunicações: Determinar as necessidades das partes envolvidas quanto à informação e comunicação, como quem necessita de tal informação, quando necessita e como a informação será fornecida;

- e) Identificação dos riscos: identificar os riscos prováveis do projeto e documentar as características de cada um;
- f) Análise qualitativa dos riscos: analisar qualitativamente os riscos e condições para priorizar seus efeitos nos objetivos do projeto;
- g) Planejamento de resposta a riscos: desenvolver procedimentos e técnicas para aumentar as oportunidades e reduzir ameaças de riscos para os objetivos do projeto;
- h) Planejamento das aquisições: determinar o que contratar e quando;
- i) Preparação das aquisições: documentar os requisitos do produto/serviço a ser adquirido e as fontes possíveis de fornecimento.

2.6.3 PROCESSOS DE EXECUÇÃO

Os processos de execução são realizados a partir do que foi definido no processo de planejamento. Nesta fase, uma das principais atividades será o gerenciamento dos recursos humanos. Durante sua execução, a equipe do projeto deverá resolver conflitos sobre prioridades, custos, recursos, etc. Os processos de execução incluem os processos essenciais e os facilitadores.

O PMI (2002) descreve o seguinte processo essencial:

- a) Execução do plano do projeto: levar a cabo o plano do projeto através da realização das atividades nele incluídas.

Os processos facilitadores são os seguintes:

- a) Garantia da qualidade: avaliar regularmente o desempenho geral do projeto com o objetivo de prover confiança de que o projeto irá satisfazer os padrões estabelecidos de qualidade.
- b) Desenvolvimento da equipe: desenvolver habilidades das pessoas e da equipe, enquanto grupo, para melhorar o desempenho do projeto.
- c) Distribuição das informações: disponibilizar as informações necessárias, e no momento adequado, às partes envolvidas.
- d) Pedido de propostas: obter, conforme apropriado a cada caso, as propostas de fornecimento dos produtos/serviços pretendidos.

- e) Seleção de fornecedores: escolher entre os possíveis fornecedores.
- f) Administração dos contatos: gerenciar os relacionamentos entre os fornecedores.

2.6.4 PROCESSOS DE CONTROLE

Os processos de controle consistem no acompanhamento das atividades, com base no plano, com a finalidade de medir o progresso, comparar o previsto com o realizado e fazer os ajustes necessários no projeto. Na medida em que são identificados desvios significativos, realizam-se ajustes no plano através da repetição dos processos de planejamento que sejam adequados ao desvio encontrado. Por exemplo, quando ocorrer um atraso no término de uma atividade, pode ser necessário reajustar os recursos humanos envolvidos a fim de garantir o cumprimento dos prazos.

Os processos de controle também incluem processos essenciais e facilitadores. O PMI (2002) descreve os seguintes processos essenciais:

- a) Controle integrado de mudanças: coordenar as mudanças através de todo o projeto;
- b) Relato de Desempenho: coletar e divulgar informações de desempenho. Isto inclui relatórios de status, medidas de progresso, e novas estimativas do projeto;

Os processos facilitadores são os seguintes:

- g) Verificação do Escopo: aceitar formalmente os resultados do projeto;
- h) Controle de mudanças do escopo: controlar as mudanças no escopo do projeto;
- i) Controle do cronograma: controlar as mudanças no cronograma do projeto;
- j) Controle dos Custos: controlar as mudanças no orçamento do projeto;
- k) Controle da Qualidade: monitorar resultados específicos do projeto para determinar se eles atingem padrões adequados de qualidade, e identificar ações para eliminar as causas de desempenhos insatisfatórios;

- l) Controle e monitoração de riscos: acompanhar os riscos identificados, monitorar os riscos residuais e identificar novos riscos, garantindo a execução dos planos de risco e avaliar sua efetividade em reduzir riscos.

2.6.5 PROCESSOS DE ENCERRAMENTO

O processo de encerramento é o processo em que o produto/serviço é concluído, instalado no cliente, e aceito pelos usuários. Nesta fase, é verificada toda a documentação do produto, os resultados obtidos e se os objetivos iniciais foram alcançados.

O PMI (2002) define dois processos de encerramento:

- a) Encerramento dos Contratos: completar e liquidar o contrato, incluindo a resolução de qualquer item pendente;
- b) Encerramento Administrativo: parar, reunir e disseminar informações para formalizar o término da fase ou projeto, incluindo avaliações do projeto e compilação das lições aprendidas para uso no planejamento de futuros projetos ou fases.

2.7 ÁREAS DE CONHECIMENTO OU DE ATUAÇÃO GERENCIAL

As áreas de conhecimento da gerência de projetos descrevem os conhecimentos e práticas em gerência de projetos em termos dos processos que as compõe. O PMI define nove áreas de conhecimento gerencial para os processos de gerenciamento de projetos (Prado, 2000), ilustrados na figura 4:

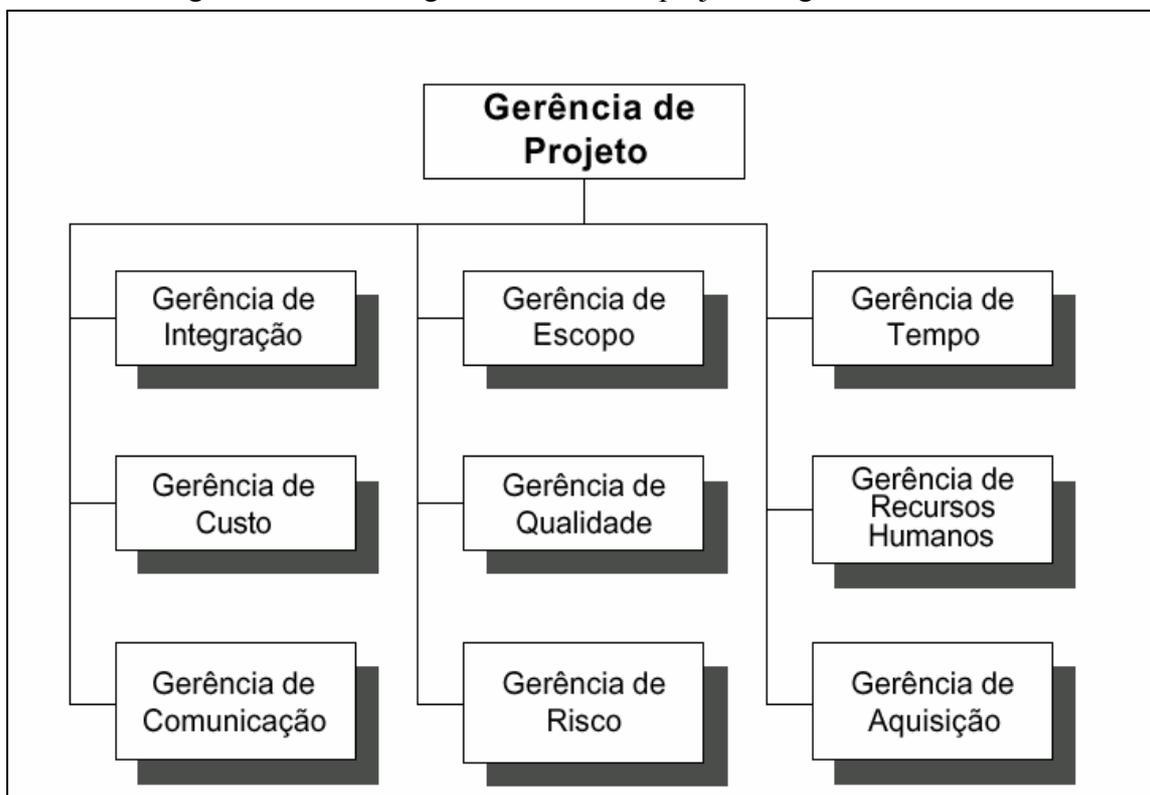
- a) A Gerência de Integração inclui os processos necessários para assegurar que os vários elementos do projeto estão adequadamente coordenados. Ela se aplica tanto ao desenvolvimento do Plano de Ação do Projeto como à sua execução e ao controle de alterações.
- b) A Gerência do Escopo inclui os processos necessários para assegurar que o projeto inclui todo o trabalho necessário, e somente o trabalho necessário, para completar o trabalho com sucesso. Por “escopo” se entende o que será feito, relativamente ao trabalho e ao produto, e esta área trata tanto da

definição do escopo e do seu controle durante a execução do projeto. A principal técnica para a definição do escopo é a confecção da Estrutura de Decomposição do Projeto (*Work Breakdown Structure*).

- c) A Gerência do Tempo inclui os processos necessários para assegurar o planejamento e execução do projeto em um prazo adequado. Esta área engloba o levantamento das atividades do projeto (definição, seqüenciamento, estimativa de duração), o agendamento (*Scheduling*) do projeto e seu controle.
- d) A Gerência de Custos inclui os processos necessários para assegurar que o projeto possa ser executado dentro do orçamento aprovado. Esta área engloba o planejamento de recursos, as estimativas de custos dos recursos, a confecção do orçamento e o controle de custos.
- e) A Gerência da Qualidade inclui os processos necessários para assegurar que o projeto vai satisfazer as necessidades para o qual foi concebido. Esta área engloba o planejamento da qualidade, a garantia da qualidade e o controle da qualidade.
- f) A Gerência de Recursos Humanos inclui os processos necessários para que se faça o melhor uso dos recursos humanos envolvidos no projeto. Esta área engloba o planejamento organizacional, a formação e desenvolvimento da equipe do projeto.
- g) A Gerência de Comunicações inclui os processos necessários para assegurar a adequada geração, disseminação e armazenamento de informações do projeto. Esta área engloba o planejamento e a distribuição de informações.
- h) A Gerência de Riscos inclui os processos relacionados com a identificação e análise dos riscos do projeto. Esta área engloba a identificação dos riscos, sua quantificação, estabelecimento de contra-medidas e acompanhamento dos fatores de risco.
- i) A Gerência de Suprimentos e Contratação inclui os processos necessários para a aquisição de bens e serviços fora da organização executora do projeto. É nesta área que é feita a confecção do plano de compras (bens e serviços), o levantamento de potenciais fornecedores, a licitação, a contratação, a administração do contrato e o fechamento do contrato.

Todas as atividades e fases do projeto são extensamente documentadas no PMBOK, organizadas pelas nove áreas de conhecimento, conforme a figura 4. No tópico seguinte, serão estudados os conceitos e características do gerenciamento de custos do projeto, que servirá como base para o desenvolvimento do protótipo.

Figura 4 – Áreas de gerenciamento de projetos segundo o PMBOK.



FONTE: PMI (2002).

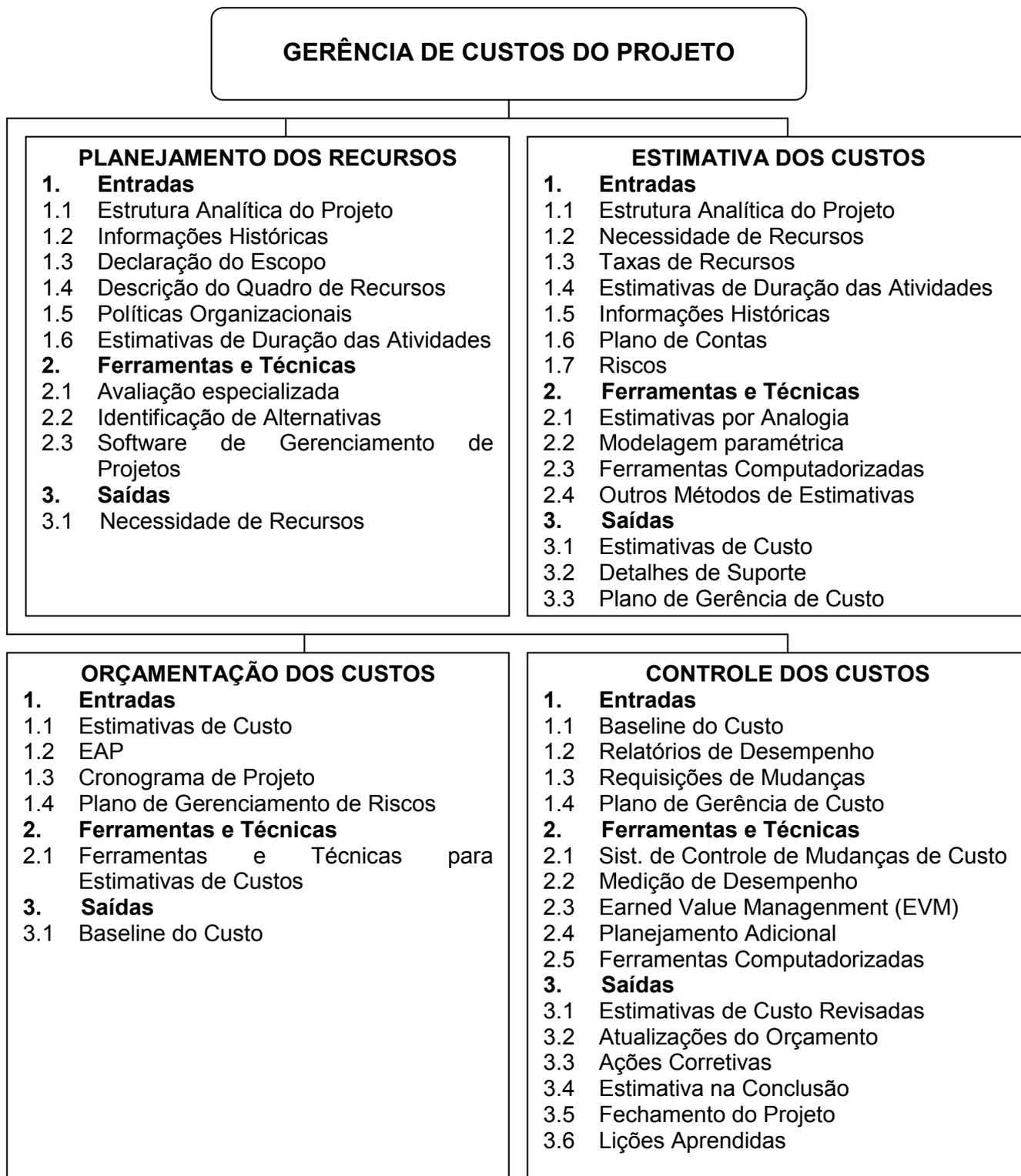
2.7.1 GERÊNCIA DE CUSTOS

A área da Gerência de Custos inclui os processos necessários para assegurar que o projeto será concluído dentro do orçamento aprovado. O PMI (2002) define os seguintes processos de gerência de custos:

- a) Planejamento dos Recursos: determinação de quais recursos (pessoal, material, equipamentos) e que quantidades devem ser utilizadas para realizar as atividades do projeto;
- b) Estimativa de custos: desenvolvimento de uma estimativa aproximada dos custos dos recursos necessários para completar as atividades do projeto;

- c) Orçamento dos custos: alocação das estimativas de custos para os itens de trabalho;
- d) Controle dos custos: controlar as modificações de custo do projeto.

Figura 5 – Visão Geral da Gerência de custos do projeto.



FONTE: PMI (2002).

A gerência do custo do projeto consiste, fundamentalmente, nos custos dos recursos necessários à implementação das atividades deste. Entretanto, a gerência do custo deve, também, considerar os efeitos das decisões do projeto no custo de utilização do produto. As técnicas de custo de ciclo de vida e engenharia de valor são usadas para reduzir custo e prazo, melhorar qualidade e desempenho e otimizar a tomada de decisão. A figura 5 traz uma visão geral da Gerência de Custos do Projeto.

2.7.1.1 PLANEJAMENTO DOS RECURSOS

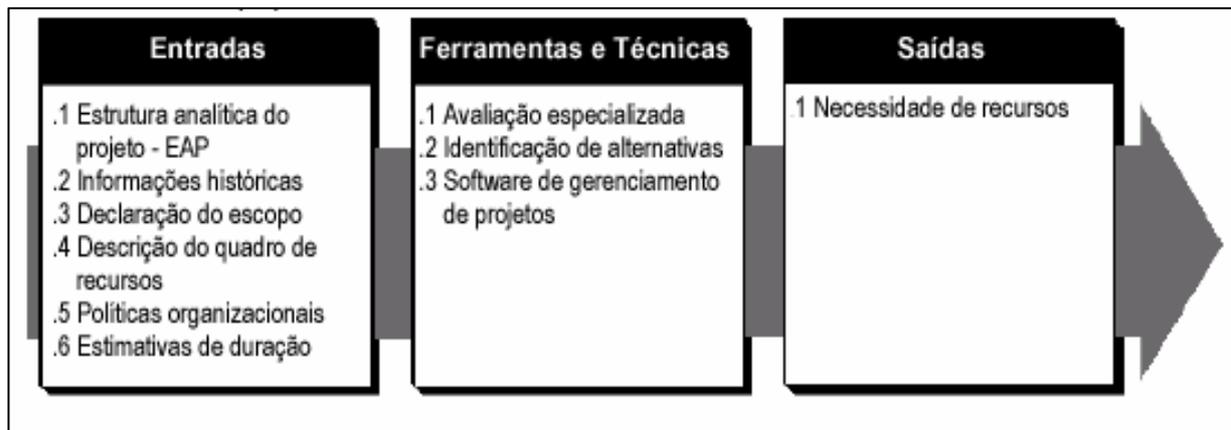
O planejamento dos recursos significa determinar quais recursos físicos (pessoas, equipamentos e materiais) são necessários, que quantidades de cada um devem ser usadas, e quando serão necessárias para a realização das atividades do projeto (PMI, 2002). Uma visão geral Planejamento dos Recursos pode ser vista na figura 6.

Dentre as entradas de dados destes processos tem-se a *EAP*, que define os elementos do projeto que necessitarão de recursos; a declaração de escopo, que contém a justificativa e os objetivos do projeto a serem considerados no planejamento dos recursos; as informações históricas, que correspondem aos tipos de recursos que foram anteriormente utilizados em projetos similares; a descrição do quadro de recursos, que consiste no conjunto de recursos que possivelmente estarão disponíveis para uso no projeto; e as políticas organizacionais, que definem as regras da organização para os recursos próprios e para o aluguel ou compra de itens de fornecimento e equipamentos de terceiros que porventura venham a ser necessários para a execução do projeto.

As ferramentas e técnicas disponíveis para este processo são a avaliação de especialistas, podendo ser provido através de empresas de consultoria, associações técnicas e de profissionais, grupos de empresas, outras áreas da empresa; a identificação de alternativas e software para gerenciamento de projetos.

Como saídas deste processo tem-se os requisitos requeridos, que incluem uma descrição de que tipos de recursos são requeridos e em que quantidade para cada elemento de trabalho da *EAP*, sendo que estes recursos poderão ser obtidos através de aquisição de pessoal ou contratação.

Figura 6 – Visão Geral Planejamento dos Recursos do Projeto.



FONTE: PMI (2002).

2.7.1.2 ESTIMATIVA DOS CUSTOS

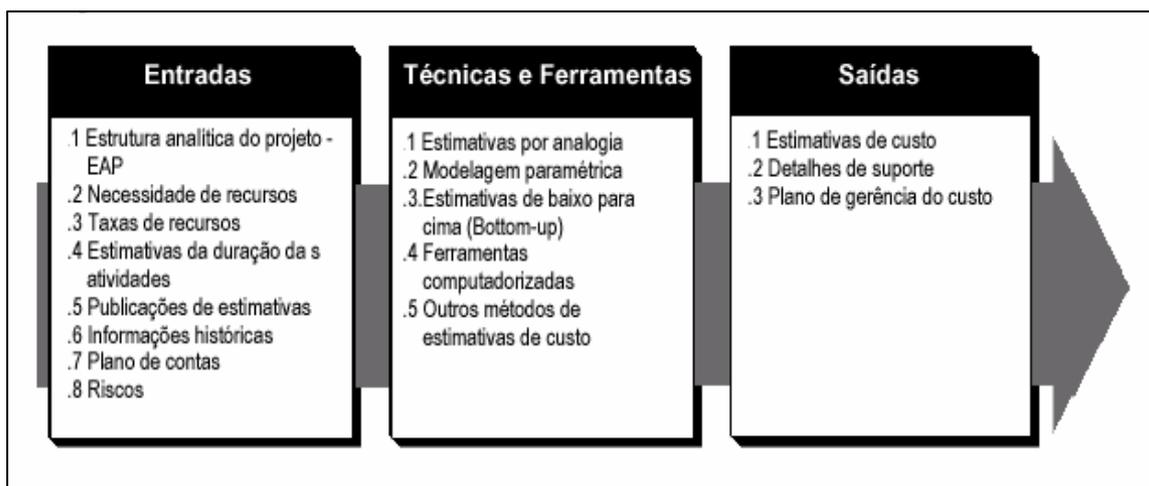
Segundo o PMI (2002), a estimativa dos custos envolve o desenvolvimento de uma aproximação (estimativa) dos custos dos recursos necessários para completar as atividades do projeto. A figura 7 exhibe uma visão geral da estimativa dos custos do projeto.

Dentre as entradas de dados destes processos tem-se a EAP, a fim de garantir que todo elemento de trabalho identificado tenha sua estimativa de custo efetuada; as necessidades de recursos; suas taxas indicadas por unidades de recurso com o intuito de se calcular o custo total do projeto; as estimativas de duração das atividades; as informações históricas, que podem ser obtidas através de arquivos de projetos anteriormente executados, de base de dados comerciais de estimativas de custo e do conhecimento dos membros do time de projeto; além do plano de contas e riscos.

As ferramentas e técnicas disponíveis são as estimativas por analogia, utilizadas especialmente em projetos onde o nível de detalhe ainda é pequeno, consistindo em utilizar as estimativas de custos já realizadas como base para a estimativa dos ainda por realizar; a modelagem paramétrica, que consiste em utilizar as características do projeto em um modelo matemático de determinação de custos; a estimativa de baixo para cima (*bottom-up*), que consiste na aferição dos custos de cada item de trabalho, na sumarização dos mesmos e na sua distribuição ao longo do projeto a fim de se calcular o custo total; além do uso de *software* específico, como ferramentas de planejamento e planilhas eletrônicas, para a estimativa total.

Como saídas esperadas tem-se a estimativa de custo, que correspondem ao valor do custo associado aos recursos necessários para a conclusão do projeto em questão; os detalhes de suporte, que incluem a descrição do escopo sobre o qual o custo foi estimado, a documentação das bases utilizadas para a estimativa, a documentação das premissas utilizadas e as margens das estimativas; além do plano de gerenciamento de custos, que descreve como as variações de custos serão administradas.

Figura 7 – Visão Geral da Estimativa dos Custos.



FONTE: PMI (2002).

2.7.1.3 ORÇAMENTAÇÃO DOS CUSTOS

A orçamentação dos custos envolve alocar as estimativas globais às atividades individuais dos pacotes de trabalho com a finalidade de estabelecer um *baseline*¹ de custo para medir o desempenho do projeto (PMI, 2002).

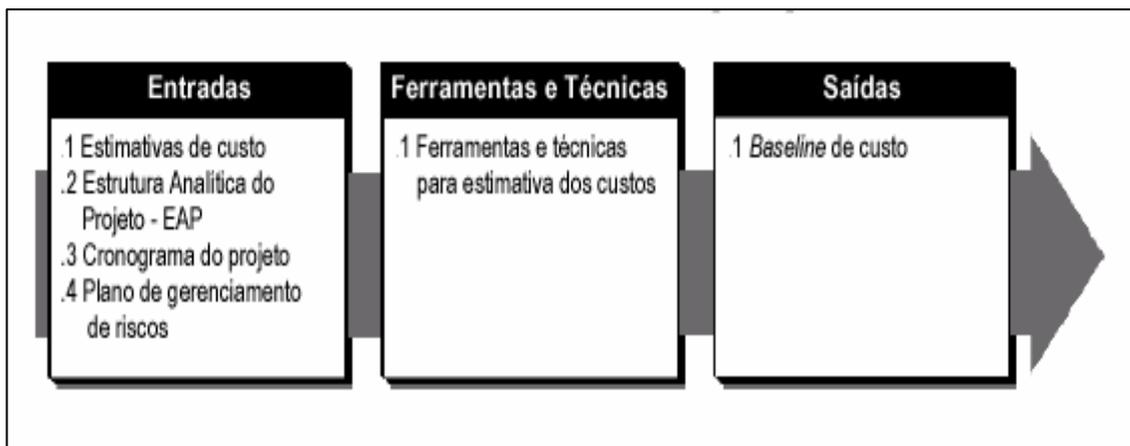
Dentre as entradas de dados destes processos tem-se a EAP, as estimativas de custo; o cronograma do projeto, incluindo as datas planejadas de início e término de cada elemento ao qual serão alocados custos, e o plano de gerenciamento de riscos.

As ferramentas e técnicas para orçar custos são as mesmas utilizadas para as estimativas de custos.

Como saída esperada tem-se o *baseline* de custo, que corresponde à base de custos a ser utilizada para medir e monitorar o desempenho de custos do projeto, conforme se pode verificar na figura 8.

¹ Uma situação inicial de referência de planejamento normalmente utilizada para comparação do planejamento com o real.

Figura 8 – Orçamentação dos custos do projeto.



FONTE: PMI (2002).

2.7.1.4 CONTROLE DOS CUSTOS

De acordo com o PMI (2002), o controle dos custos está associado a: influenciar os fatores que criam as mudanças no *baseline* de custo de forma a garantir que estas mudanças sejam benéficas; determinar que o *baseline* de custo foi alterado; gerenciar as mudanças reais quando e como elas surgirem.

O controle dos custos inclui: monitorar o desempenho do custo para detectar e entender as variações do plano; assegurar que todas as mudanças apropriadas estão registradas corretamente no *baseline* de custo; informar as partes envolvidas afetadas sobre as mudanças autorizadas; atuar no sentido de manter os custos esperados dentro de limites aceitáveis.

Dentre as entradas de dados destes processos tem-se o *baseline* de custo; os relatórios de desempenho, que contém informações sobre quais itens estiveram de acordo com o orçado e quais não estiveram; as requisições de mudanças; além do plano de gerenciamento de custos.

As ferramentas e técnicas para o controle dos custos definidos pelo PMBOK são o sistema de controle de mudanças de custo, que define o procedimento a ser seguido para atender às requisições de mudanças que causam algum impacto no orçamento; as medições de desempenho, que auxiliam a identificar possíveis desvios entre o planejado e o realizado até o momento; o gerenciamento de valor agregado, que mede continuamente o desempenho do projeto; planejamento adicional, que pode ser necessário em função de mudanças solicitadas ou a fim de corrigir algum eventual

desvio do projeto e revisão dos custos anteriormente estimados e o uso de ferramentas computadorizadas para o seu controle.

Como saídas esperadas para o controle de custos tem-se as estimativas de custo revisadas, que contém as modificações efetuadas em função dos controles apresentados; as ações corretivas, a fim de evitar novos problemas neste projeto; as estimativas para conclusão (*estimate at completion - EAC*), que correspondem ao cálculo do custo total do projeto baseado no desempenho do mesmo e lições aprendidas; as atualizações do orçamento, que são as mudanças no *baseline* aprovado; o fechamento do projeto, que estabelece processos e procedimentos para o encerramento ou cancelamento dos projetos; e lições aprendidas. Uma visão geral dos processos de controle dos custos de um projeto pode ser vista na figura 9.

Figura 9 – Processos de controle dos custos.



FONTE: PMI (2002).

2.8 MÉTRICAS NO PROCESSO DE GERENCIAMENTO DE PROJETOS DE SOFTWARE

O gerenciamento de projetos, segundo Pressman (1995), compreende atividades que envolvem, principalmente: medição, estimativa e análise dos riscos e determinação de um cronograma, além do monitoramento e controle dessas atividades.

Quando se considera boa parte dos empreendimentos técnicos, verifica-se que as medições e as métricas permitem um melhor entendimento do processo utilizado para desenvolver um produto, assim como uma melhor avaliação do próprio produto. (Pressman, 1995).

A quantificação dos aspectos relacionados ao processo de obtenção de um produto, assim como do produto, é importante, pelas seguintes razões:

- a) No caso do processo de desenvolvimento, as medições podem permitir melhorias no processo, aumentando a sua produtividade;
- b) No caso do produto, as medições podem proporcionar informações a respeito de sua qualidade.

De acordo com Fernandes (1995), existem várias características importantes associadas ao emprego das métricas de software. Sua escolha, para fins de gerenciar através de fatos e dados, requer alguns pré-requisitos importantes, quais sejam:

- a) A utilização de métricas em uma organização que se dedica ao desenvolvimento de software, seja como atividade-fim, seja como de suporte a uma empresa, dependendo do estágio de maturidade em que se encontram essas atividades;
- b) Devido às diversas audiências (destinatários dos produtos das medições), as métricas devem ser simples de entender e de serem utilizadas para verificar se os objetivos foram atingidos para, assim, subsidiarem as tomadas de decisão;
- c) As métricas devem ser objetivas, visando minimizar a influência do julgamento pessoal na coleta, cálculo e análise dos resultados;
- d) O valor da informação obtida como resultado das medições deve exceder o custo de coletar, armazenar e calcular as métricas. A efetividade no custo deve ser analisada em função do estágio de maturidade do ambiente de desenvolvimento que se encontra a empresa no contexto de uma estratégia de melhoria contínua;
- e) As métricas selecionadas devem propiciar informação que possibilite avaliar acertos ou não de decisões e ações realizadas no passado, evidenciar a ocorrência de eventos presentes que subsidiem decisões tempestivas, bem como provar a possibilidade de ocorrência de eventos futuros.

Existem dois tipos de métricas de software: métricas orientadas a função e métricas orientadas ao tamanho. Segundo Pressman (1995), Métricas de software orientadas ao tamanho são medidas diretas do software e do processo por meio do qual

ele é desenvolvido. As métricas orientadas ao tamanho provocam controvérsias e não são universalmente aceitas como a melhor maneira de se medir o processo de desenvolvimento de software.

Métricas de software orientadas à função são medidas indiretas do software e do processo por meio do qual ele é desenvolvido. Em vez de contar as linhas do código, por exemplo, a métrica orientada à função concentra-se na funcionalidade ou utilidade do software, ou seja, mede o tamanho do software pela quantificação de sua funcionalidade externa, baseada no projeto lógico ou a partir do modelo de dados. De acordo com Choose (2000), o ponto-por-função representa efetivamente a métrica de estimativa mais largamente utilizada na indústria para dimensionamento de projetos de software.

2.9 ANÁLISE DE PONTOS DE FUNÇÃO

De acordo com Silva (2000), a Análise de Pontos de Função (FPA) é uma unidade de medida que procura definir o tamanho do que faz o software, independente de como possa ser produzido e implementado.

Surgiu no final dos anos 70, e seus conceitos foram introduzidos por Allan J. Albrecht, da IBM. Em 1986, uma comunidade de usuários forma o Grupo Internacional de Usuários de Pontos de Função (IFPUG), que se tornou líder na publicação de documentos relacionados ao assunto, incluindo o Manual de Práticas de Contagem, hoje na versão 4.1.

Segundo Braga (1996), o objetivo principal da Análise de Pontos de Função é medir a funcionalidade de um software ou aplicativo, baseando-se primeiramente no desenho lógico, e de acordo com a perspectiva do usuário.

De acordo com Choose (2000), as principais vantagens do uso de pontos por função são:

- a) Pode ser aplicada com certa facilidade, logo nas etapas iniciais do projeto de software;
- b) Permite medir o tamanho do projeto de software, o qual é expresso em termos de quantidade de pontos-por-função;

- c) Atualmente, pode-se dispor de várias estatísticas de produtividade, esforços e prazos baseados na técnica de pontos-por-função.

2.9.1 COMPONENTES BÁSICOS DA ANÁLISE DE PONTOS DE FUNÇÃO

De acordo com Silva (2000), a análise de pontos de função segue o método humano de resolver um problema: dividi-lo em partes para que seja melhor entendido e mais facilmente resolvido.

A análise de pontos de função decompõe e classifica o software em cinco componentes básicos, enquadrando-os em duas categorias: funções do tipo dados e funções do tipo transação. A figura 10 ilustra a funcionalidade do FPA vista segundo a perspectiva do usuário.

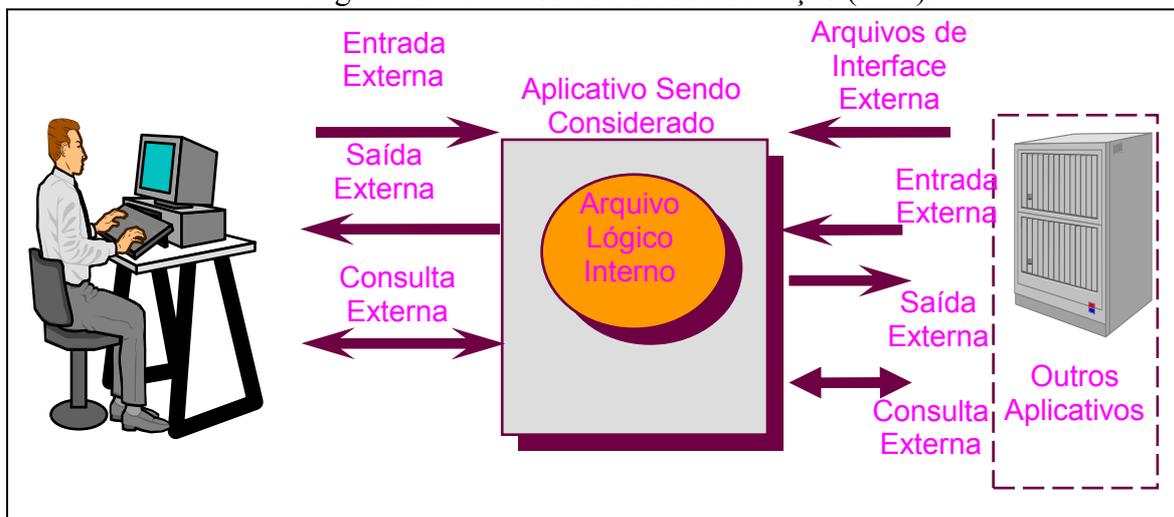
As funções do tipo dados são as seguintes:

- a) Arquivo Lógico Interno (ALI);
- b) Arquivo de Interface Externa (AIE).

As funções do tipo transação incluem:

- a) Entrada Externa (EE);
- b) Saída Externa (SE);
- c) Consulta Externa (CE).

Figura 10 – Análise de Pontos de Função (FPA).



FONTE: BFPUG (2001).

2.9.2 ARQUIVO LÓGICO INTERNO

De acordo com Braga (1996), um Arquivo Lógico Interno (ALI) é um grupo de dados logicamente relacionados, ou informações de controle, identificado pelo usuário e utilizado dentro das fronteiras da aplicação.

O processo de contagem de um ALI é baseado em Registros Lógicos Referenciados (RLR) e Dados Elementares Referenciados (DER).

Um Registro Lógico Referenciado é um subgrupo de dados que devem ser enxergados como um elemento único. Se não existem subgrupos de dados, considera-se que o arquivo lógico interno apresenta um registro lógico.

Arquivos Lógicos Referenciados são os dados que dão suporte às necessidades do usuário, segundo sua própria visão. A tabela 2 exibe a tabela de complexidade de arquivos lógicos internos.

Tabela 2 – Complexidade de arquivos lógicos internos.

| | 1 a 19 DER | 20 a 50 DER | 51 DER em diante |
|---------------|-------------|---------------|------------------|
| 1 LRL | Simples (7) | Simples (7) | Média (10) |
| 2 a 5 RLR | Simples (7) | Média (10) | Complexa (15) |
| 6 RLR ou mais | Média (10) | Complexa (15) | Complexa (15) |

FONTE: Silva (2000).

2.9.3 ARQUIVO DE INTERFACE EXTERNA

De acordo com Silva (2000, p. 13), “um Arquivo de Interface Externa (AIE) é um grupo lógico de dados relacionados, identificados pelo usuário, usados apenas como referência pelo aplicativo. Os dados residem inteiramente fora da fronteira do aplicativo e são mantidos por outros aplicativos”.

Estes arquivos não podem sofrer nenhuma alteração, pois se isto ocorrer está-se diante de uma função do tipo transação. Um Arquivo de Interface Externa para um aplicativo é um Arquivo Lógico Interno em outro aplicativo.

As regras para contagem de um AIE são muito parecidas com o processo de contagem de Arquivos Lógicos Internos. Baseia-se em Registros Lógicos Referenciados

(RLR) e Dados Elementares Referenciados (DER), conforme se pode visualizar na tabela 3.

Tabela 3 – Complexidade de arquivos de interface externa.

| | 1 a 19 DER | 20 a 50 DER | 51 DER em diante |
|---------------|-------------|---------------|------------------|
| 1 LRL | Simples (5) | Simples (5) | Média (7) |
| 2 a 5 RLR | Simples (5) | Média (7) | Complexa (10) |
| 6 RLR ou mais | Média (7) | Complexa (10) | Complexa (10) |

FONTE: Silva (2000).

2.9.4 ENTRADA EXTERNA

Segundo Dekkers (1998, p.9), “uma entrada externa é um processo lógico do negócio que mantém os dados em um ou mais arquivos lógicos internos, ou é um processo de controle que direciona o software para atender os requisitos de negócio do usuário”.

Para a contagem de entradas externas é preciso conhecer os campos de dados envolvidos bem como os arquivos lógicos internos e arquivos de interface externa, envolvidos no processo.

Telas de inclusão, alteração e exclusão de alunos em um colégio seriam bons exemplos de entradas externas. Não podem ser consideradas entradas externas telas que não mantêm arquivos lógicos internos.

A complexidade de uma Entrada Externa é calculada de acordo com a quantidade de Arquivos Lógicos Referenciados (ALR) e de Dados Elementares Referenciados (DER), conforme ilustrado na tabela 4.

Tabela 4 – Complexidade de entradas externas.

| | 1 a 4 DER | 5 a 15 DER | 16 DER em diante |
|---------------|-------------|--------------|------------------|
| 0 ou 1 ALR | Simples (3) | Simples (3) | Média (4) |
| 2 ALR | Simples (3) | Média (4) | Complexa (6) |
| 3 ALR ou mais | Média (4) | Complexa (6) | Complexa (6) |

FONTE: Silva (2000).

2.9.5 SAÍDA EXTERNA

SILVA (2000, p. 19) define Saída Externa como “um processo elementar através do qual dados derivados são enviados para ele, da fronteira do aplicativo. Os dados podem ser relatórios ou arquivos de saída que são enviados para outros aplicativos. Os relatórios ou arquivos de saída são criados a partir de arquivos lógicos internos e arquivos de interface externa”.

Se todos os dados apresentados por uma saída externa estiverem na forma original, está se tratando de outro tipo de função: a consulta externa.

A finalidade básica de uma saída externa é a de recuperar dados, processá-los e enviá-los para além da fronteira do aplicativo. Todavia uma saída externa pode, eventualmente, atualizar arquivos lógicos internos.

A complexidade de uma saída externa é calculada de acordo com a quantidade de arquivos lógicos referenciados (ALR) e de dados elementares referenciados (DER), melhor visualizadas através da tabela 5.

Tabela 5 – Complexidade de saídas externas.

| | 1 a 5 DER | 6 a 19 DER | 20 DER em diante |
|---------------|------------|--------------|------------------|
| 0 ou 1 ALR | Simple (4) | Simple (4) | Média (5) |
| 2 ou 3 ALR | Simple (4) | Média (5) | Complexa (7) |
| 4 ALR ou mais | Média (5) | Complexa (7) | Complexa (7) |

FONTE: Silva (2000).

2.9.6 CONSULTA EXTERNA

De acordo com Dekkers (1998), uma consulta externa é o último tipo de função lógica de usuário contada através de pontos de função. Uma consulta externa consiste em um par pergunta-resposta através do qual a pergunta ou solicitação de dados entra no aplicativo vindo de fora (tipicamente a partir de um usuário ou de outro aplicativo), os dados são recuperados para atender à solicitação e enviados para fora.

Uma Consulta Externa (CE) é um processo elementar com componentes de entrada e saída recuperados de arquivos lógicos internos e arquivo de interface externa. Os dados são enviados para além da fronteira do aplicativo e os dados não sofrem nenhuma alteração.

A complexidade de uma consulta externa é calculada de acordo com a quantidade de arquivos lógicos referenciados (ALR) e de dados elementares referenciados (DER).

Os elementos da parte de entrada devem ser somados com os elementos da parte de saída, exceto se um elemento aparece como elemento de entrada e como elemento de saída. Neste caso deve ser contado apenas uma vez.

Tabela 6 – Complexidade de consultas externas.

| | 1 a 5 DER | 6 a 19 DER | 20 DER em diante |
|---------------|-------------|--------------|------------------|
| 0 ou 1 ALR | Simples (3) | Simples (3) | Média (4) |
| 2 ou 3 ALR | Simples (3) | Média (4) | Complexa (6) |
| 4 ALR ou mais | Média (4) | Complexa (6) | Complexa (6) |

FONTE: Silva (2000).

2.9.7 ITENS DE INFLUÊNCIA

A Análise de Pontos de Função prevê 14 características, conhecidas como Itens de Influência, que devem ser analisadas para ajustar o tamanho funcional do aplicativo.

As características gerais do aplicativo variam para mais ou menos 35%, o que nos leva a concluir que há um fator entre 0,65 e 1,35 a ser utilizado no ajuste do tamanho do aplicativo.

As características podem receber pontos que variam de 0 a 5. Assim, as características poderão totalizar entre 0 e 70 pontos.

A fórmula seguinte calcula o Fator de Ajuste:

$$\mathbf{Fa = (ni * 0,01) + 0,65}$$

Onde:

Fa = fator de ajuste (0,65 a 1,35)

Ni = somatório dos pontos atribuídos as características (0-70).

O Fator de Ajuste determina os pontos finais do aplicativo (ou pontos ajustados), e as funções do tipo dados e do tipo transação fornecem os pontos brutos (não-ajustados).

Segundo Silva (2000), de acordo com a versão 4.1 do manual de práticas de contagem, os itens de influência para análise de pontos de função são os seguintes:

- a) Comunicação de dados: refere-se aos aspectos relacionados à comunicação de dados do aplicativo. É importante determinar que protocolos são utilizados pelo aplicativo para a transmissão e recepção de informações;
- b) Funções distribuídas: aspectos relacionados com processamento e funções distribuídas;
- c) Performance: aspectos relacionados a parâmetros estabelecidos pelo usuário quanto a tempos de resposta;
- d) Utilização do equipamento: aspectos relacionados com o nível de utilização dos equipamentos necessários à execução do aplicativo. A avaliação visa a identificar a carga de trabalho exigida pelo aplicativo quando em produção;
- e) Volume de transações: aspectos relacionados com a capacidade do sistema em relação ao volume de transações esperado;
- f) Entrada de dados on-line: aspectos relacionados com a quantidade de entrada de dados on-line do aplicativo;
- g) Interface com o usuário: aspectos relacionados com a eficiência do aplicativo na interação com o usuário;
- h) Atualizações on-line: aspectos relacionados com o modo de atualização dos arquivos lógicos internos do aplicativo;
- i) Processamento complexo: aspectos relacionados com a complexidade do processamento;
- j) Reusabilidade: aspectos relacionados à reutilização do código do aplicativo;
- k) Facilidade de implantação: aspectos relacionados com o grau de dificuldade de implementação do aplicativo;
- l) Facilidade operacional: aspectos relacionados com a facilidade de operação do aplicativo. Avalia procedimentos operacionais automáticos e mecanismos de iniciação, salvamento e recuperação de dados;
- m) Múltiplos locais: aspectos relacionados com a arquitetura do aplicativo e a necessidade de instalação em vários lugares;

- n) Facilidade de mudanças: aspectos relacionados com o grau de flexibilidade do aplicativo com relação a mudanças nos requisitos do usuário.

2.10 TRABALHOS CORRELATOS

Outros trabalhos de conclusão de curso já foram desenvolvidos na área de gerenciamento de projetos e métricas de software, entre outros destacam-se: Kienen (2000), que desenvolveu um protótipo de uma ferramenta de software baseada na metodologia PERT/CPM para o planejamento de projetos de desenvolvimento de sistemas, que a partir de informações como processos a serem desenvolvidos, tempos de execução, custo e recursos, calcula o caminho crítico e o custo do sistema. Bauer (2001) apresentou um protótipo de software de apoio às atividades iniciais de gerência de riscos de software. O protótipo permite identificar e analisar os riscos a partir do uso de *check-list* envolvendo perguntas básicas sobre riscos técnicos, riscos de projeto e riscos de negócio. Valcanaia (1998) desenvolveu um estudo sobre métricas de software, tendo como foco principal a métrica FPA (Análise de Pontos de Função), contemplando a especificação e implementação de um protótipo para contabilização e cálculo de pontos de função em sistemas desenvolvidos no banco de dados Access 97. Lonzetti (2001) propôs um software que une as vantagens de uma ferramenta CASE, automatizando o processo de cálculo e apresentação das métricas, a um tutorial com o forte intuito de estimular o uso de técnicas de métricas de software pelos estudantes dos cursos da área de Informática para que empreguem efetivamente as métricas de software em seu ambiente de trabalho. Este trabalho utilizou as métricas orientadas ao tamanho (LOC), métricas orientadas à função (FPA) e COCOMO.

3 DESENVOLVIMENTO DO PROTÓTIPO

De acordo com os objetivos propostos por este trabalho, desenvolveu-se um protótipo que auxilia as atividades de gerenciamento de custos de software de acordo com as técnicas definidas na área de gerência de custos do PMBOK, utilizando como métrica de software a Análise de Pontos de Função.

3.1 REQUISITOS PRINCIPAIS DO PROBLEMA

O protótipo deve auxiliar nas atividades de gerência de custos para o desenvolvimento de sistemas. A atividade de gerência de custos foi desenvolvida de acordo com as características definidas pelo PMBOK, utilizando a técnica de Análise de Pontos de Função como métrica de software.

3.2 ESPECIFICAÇÃO

A especificação do protótipo foi realizada utilizando a linguagem de modelagem *Unified Modeling Language* (UML). A UML é a padronização da linguagem de desenvolvimento orientado a objetos para visualização, especificação, construção e documentação de sistemas (Martins, 2002).

Para a modelagem foram utilizados os diagramas de casos de uso, diagramas de seqüência e diagrama de classes. Para a especificação do protótipo, utilizou-se a ferramenta CASE *Power Designer 9*.

3.2.1.1 DIAGRAMA DE CASOS DE USO

A figura 11 demonstra o diagrama de casos de uso do protótipo, onde se tem como ator o gerente de projeto. Nesta figura estão representados apenas os Casos de uso primários. A figura 12 exhibe os casos de uso secundários e a figura 13 mostra uma especialização dos casos de uso de Emite Relatórios Gerenciais.

A seguir são descritos os casos de uso primários e secundários, especificando em seguida o item da Gerência de Custos do PMBOK que está sendo atendido em cada caso, de acordo com a figura 5.

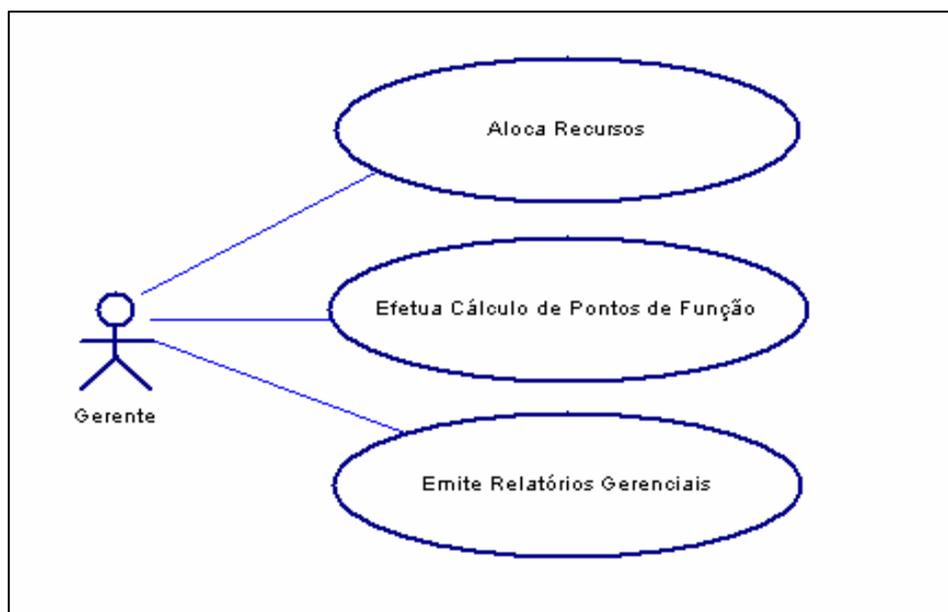
Os casos de uso primários são os seguintes:

- a) Aloca recursos: o gerente define os recursos que serão utilizados para cada atividade, que pode ser um equipamento (ou material), ou ainda uma pessoa. Atende ao item 1.2 da Estimativa de Custos.
- b) Efetua cálculo de pontos de função: o usuário efetua a classificação das funções, que podem ser: Arquivo Lógico Interno, Arquivo Interface Externo, Entradas Externas, Saídas Externas e Consultas Externas. O usuário deve informar as funções de cada tipo, assim como o número de registros lógicos e itens de dados referenciados. A complexidade de cada função é gerada, a medida em que o usuário vai informando os valores. Em seguida, devem ser informados os níveis de influência de cada característica do software. Após terem sido informados todos os dados necessários, o sistema poderá efetuar o cálculo dos pontos de função. O aplicativo mostra as funções, a complexidade de cada um, o total de pontos de cada função, o total de pontos brutos e o total de pontos de função ajustados. Atende ao item 2.2 (Modelagem Paramétrica) da Estimativa de Custos.
- c) Emite Relatórios Gerenciais: emite relatórios gerenciais para o acompanhamento dos custos de execução do projeto, conforme a figura 13:
 - a) Relatório de pontos de função: relaciona todas as informações relevantes ao cálculo de pontos de função dos sistemas cadastrados, como pontos de função brutos, características gerais do sistema, níveis de influência, fator de ajuste e pontos ajustados. Atende às saídas da estimativa de custos, referente às estimativas de custo e detalhes de suporte, e também ao item 2.2 (Modelagem Paramétrica) da Estimativa de Custos.
 - b) Alocação de recursos: relaciona os recursos alocados pelos projetos cadastrados, especificando o projeto, metodologia, fase, atividade e recursos alocados. Atende aos itens 1.2 e 3.1 do Planejamento de Recursos, 1.2, 1.3 e 1.5 da Estimativa de Custos, além do item 1.1 da orçamentação dos custos.
 - c) Gráfico de custos dos projetos: compara graficamente os custos totais dos projetos cadastrados. Atende aos itens 1.1 e 1.2 do Controle de

Custos, e também às saídas da orçamentação dos custos, no item baseline de custo.

- d) Estrutura Analítica de Trabalho (EAT): fornece uma ilustração detalhada do escopo do projeto, listando os projetos cadastrados por metodologia, fase e atividade. Atende aos itens descritos como Estrutura Analítica do Projeto (EAP), para Planejamento, Estimativa e Orçamentação dos Recursos, e também aos detalhes de suporte na estimativa dos custos.
- e) Estimativas de duração das atividades: relaciona as atividades dos projetos cadastrados por metodologia, fase e atividade, listando as datas planejadas e realizadas, separadas por 2 sub-relatórios. Atende ao item 1.4 da Estimativa de Custos, 1.6 do Planejamento dos Recursos, 1.3 da Orçamentação dos Custos, Informações Históricas e saídas da estimativa de custos.
- f) Descrição do Quadro de Recursos: lista os recursos disponíveis ao gerente de projetos, separados em dois sub-relatórios: por pessoa e equipamento, listando o nome, a descrição (ou cargo) e o custo de cada recurso. Atende ao item 1.4 do Planejamento dos Recursos.

Figura 11 – Diagrama de casos de uso Primários.



Os casos de uso secundários descrevem as atividades relacionadas ao cadastro das informações necessárias ao gerente de projetos para possibilitar o gerenciamento do projeto, de acordo com as premissas definidas pelo PMBOK.

A seguir, são descritos os casos de uso secundários:

- a) Cadastra gerentes: o usuário efetua o cadastro dos gerentes de projeto.
- b) Cadastra recursos: o usuário efetua o cadastro dos recursos disponíveis para a execução das atividades do projeto, que podem ser pessoas ou equipamentos.
- c) Cadastra projetos: cadastra um novo projeto de desenvolvimento.
- d) Cadastra atividades: o usuário cadastra as atividades de cada projeto, que pertencem a uma determinada fase da metodologia de desenvolvimento do projeto, e podem alocar recursos.
- e) Cadastra plano de contas: o usuário pode cadastrar um plano de contas para relacionar posteriormente com os recursos disponíveis.
- f) Cadastra sistema: cadastra os sistemas que serão desenvolvidos em cada projeto.
- g) Cadastra metodologia: cadastra uma metodologia de desenvolvimento de sistemas, e possui nome e descrição da metodologia.
- h) Cadastra fases: cadastra as fases de uma metodologia de desenvolvimento de sistemas.

Figura 12 – Diagrama de casos de uso secundários.

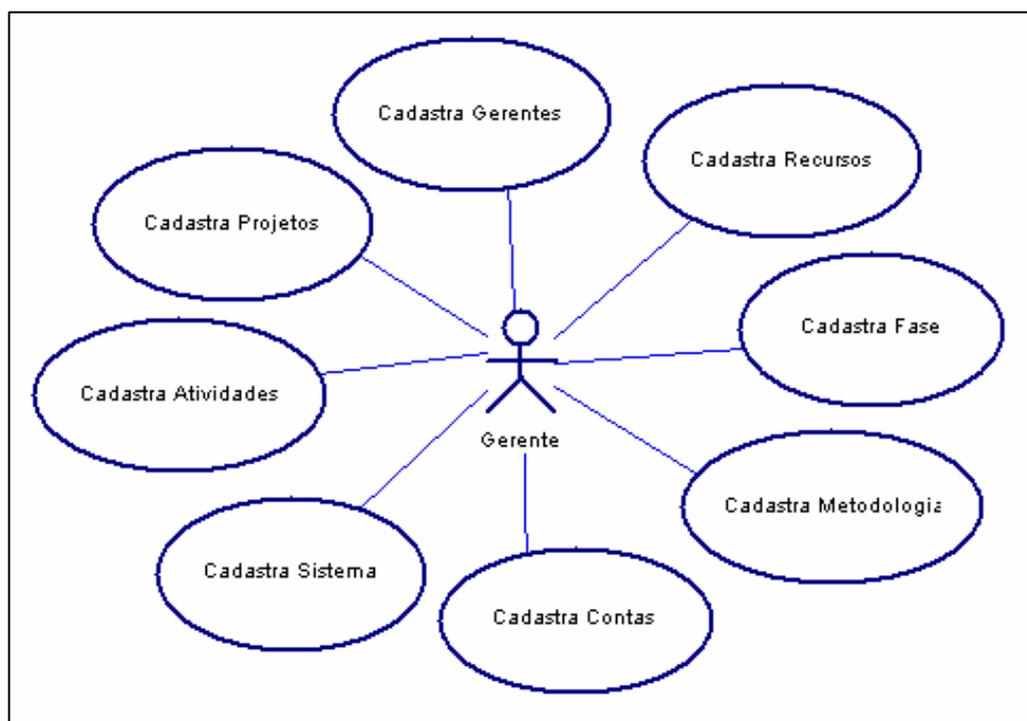
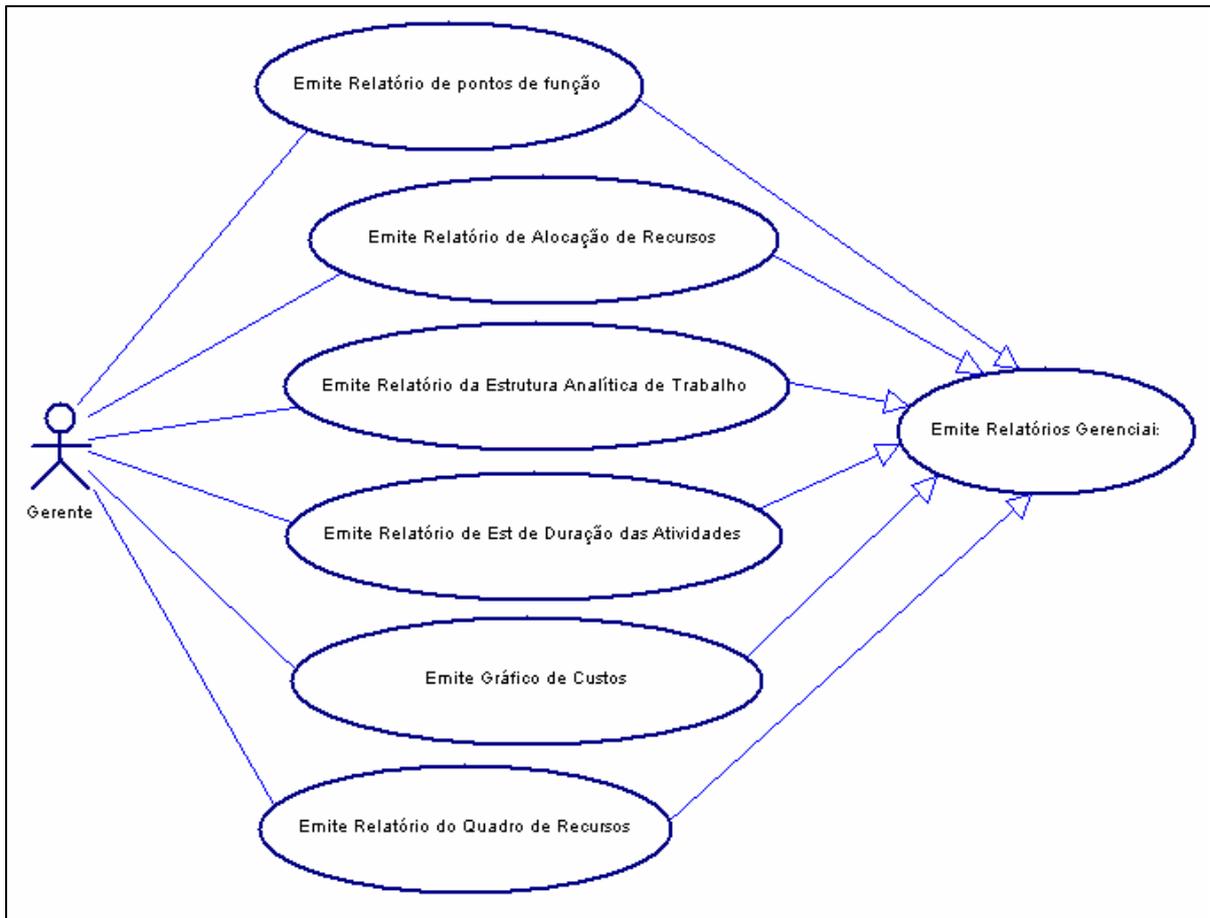


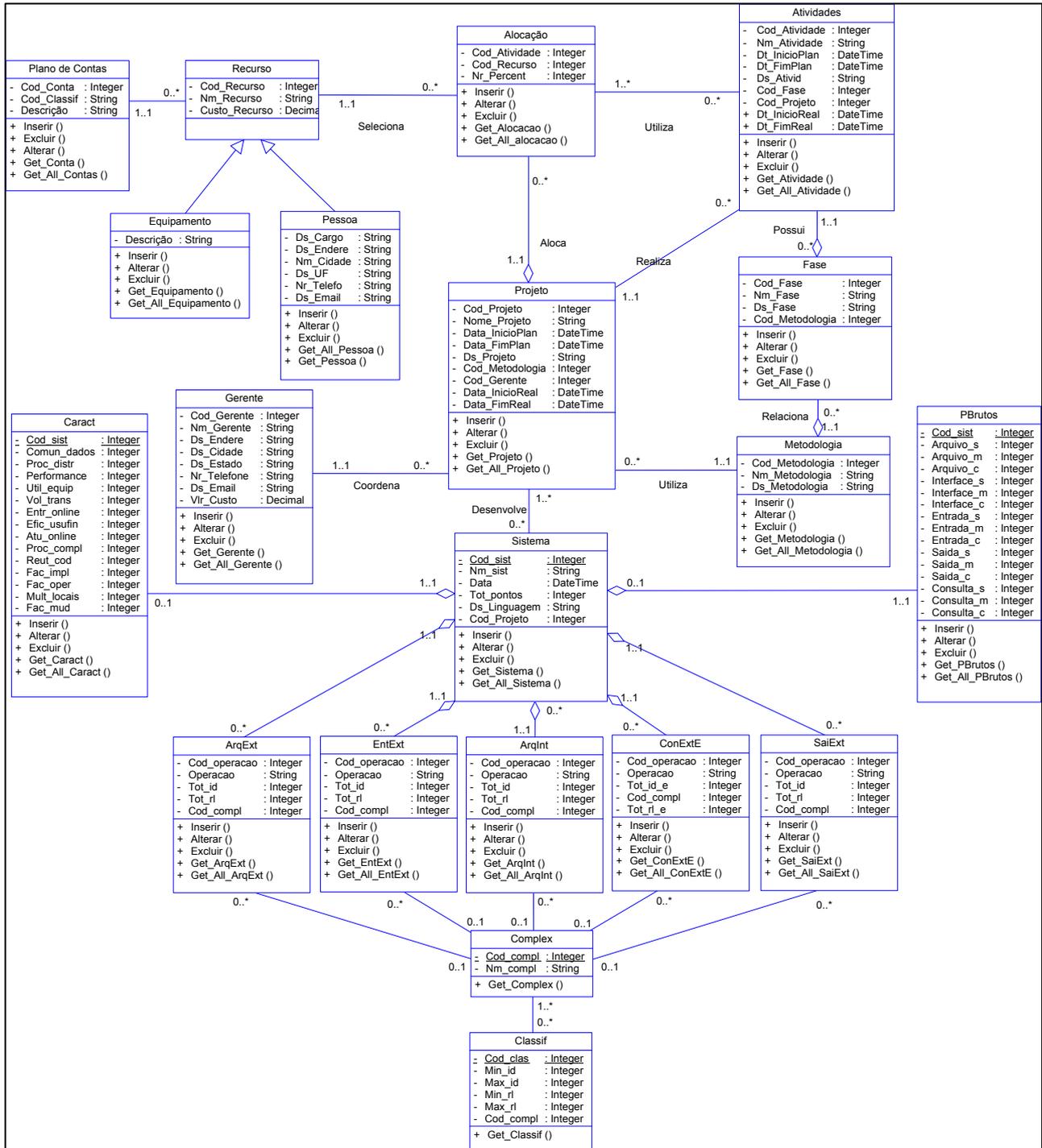
Figura 13 – Diagrama de casos de uso – Especialização de Emite Relatórios Gerenciais.



3.2.1.2 DIAGRAMA DE CLASSES

A figura 14 mostra o diagrama de classes desenvolvido para especificar o protótipo.

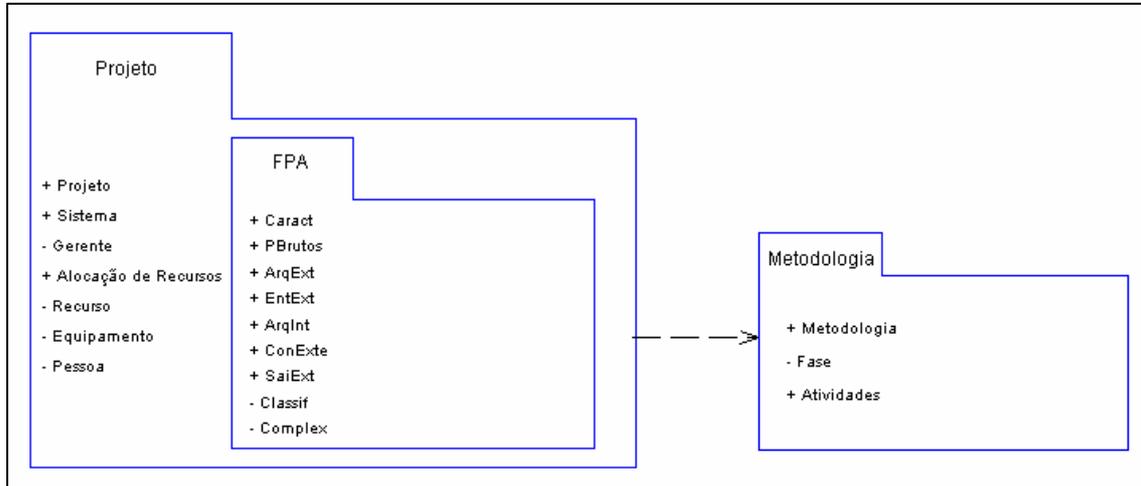
Figura 14 – Diagrama de classes.



3.2.1.3 DIAGRAMA DE PACOTES

A figura 15 demonstra o Diagrama de Pacotes do protótipo, que contém a lista de classes contidas em cada pacote.

Figura 15 – Diagrama de Pacotes.



3.2.1.4 DIAGRAMAS DE SEQÜÊNCIA

A seguir as figuras 16, 17, 18, 19, 20, 21 e 22 demonstram os diagramas de seqüência do protótipo os quais representam as opções disponíveis ao gerente de projetos.

Figura 16 – Diagrama de Seqüência Emite Relatório de Pontos de Função.

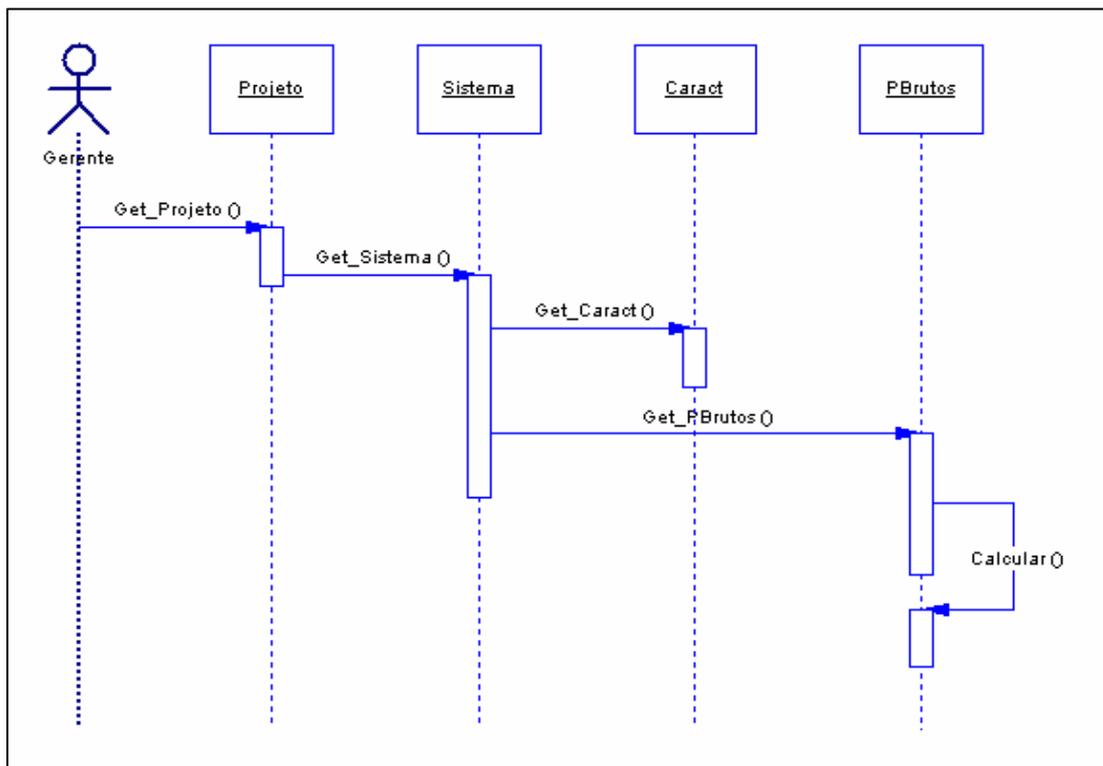


Figura 17 – Diagrama de Seqüência Calcula Pontos de Função.

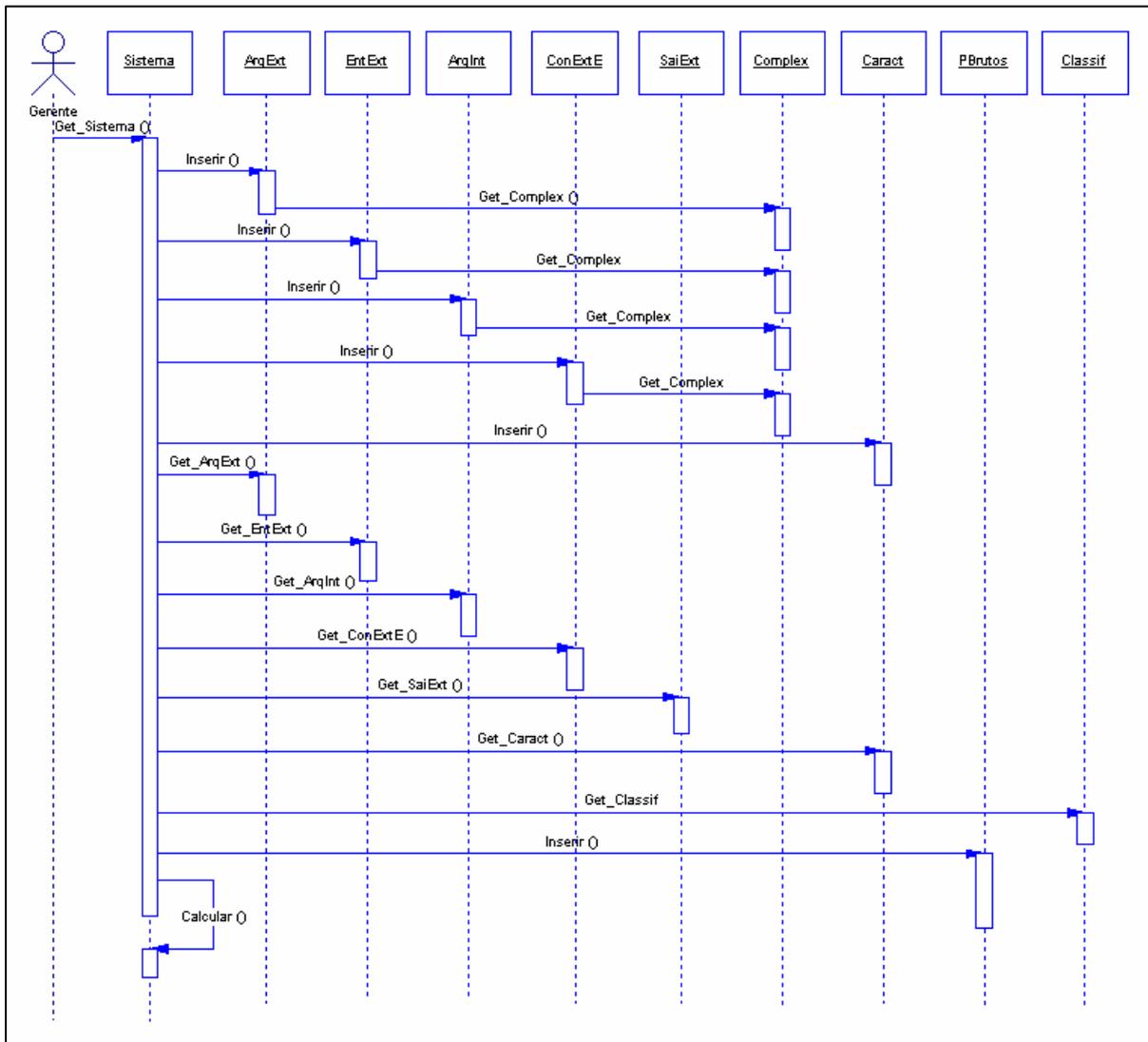


Figura 18 – Diagrama de Seqüência Aloca Recursos.

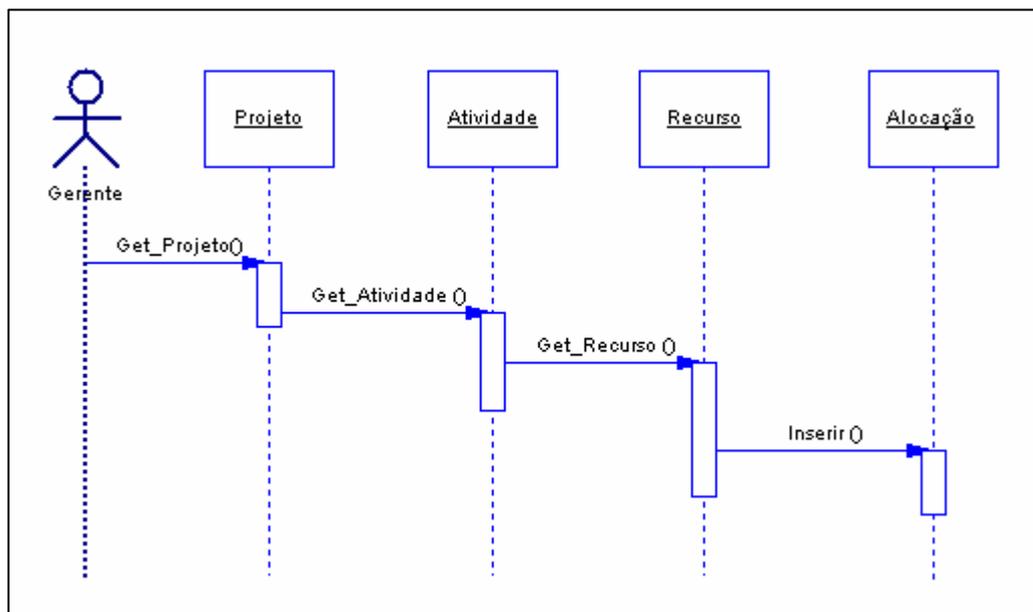


Figura 19 – Diagrama de Seqüência Emite Relatório de Alocação de Recursos.

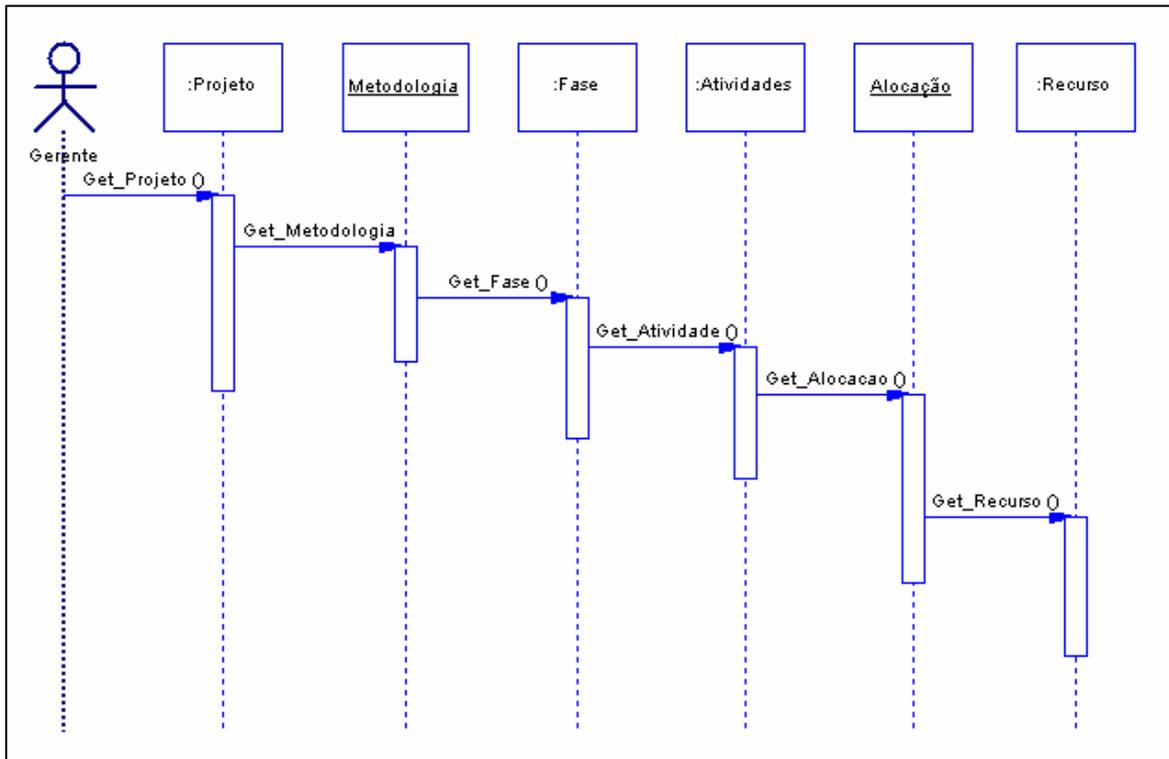


Figura 20 – Diagrama de Seqüência Emite Relatório do Gráfico de Custos.

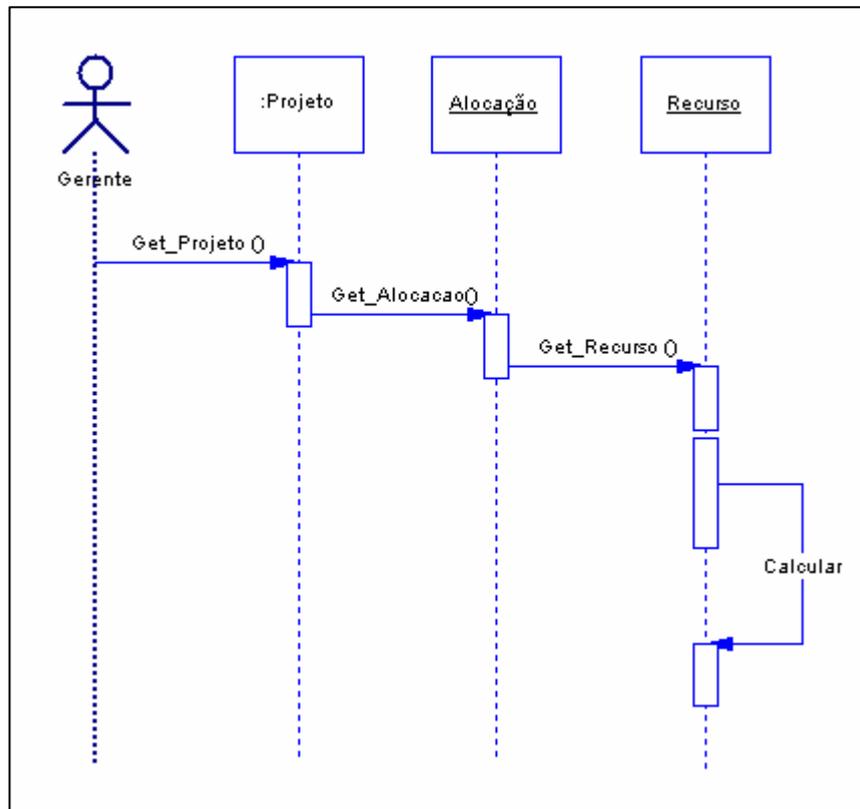


Figura 21 – Diagrama de Seqüência Emite Relatório da Estrutura Analítica de Trabalho.

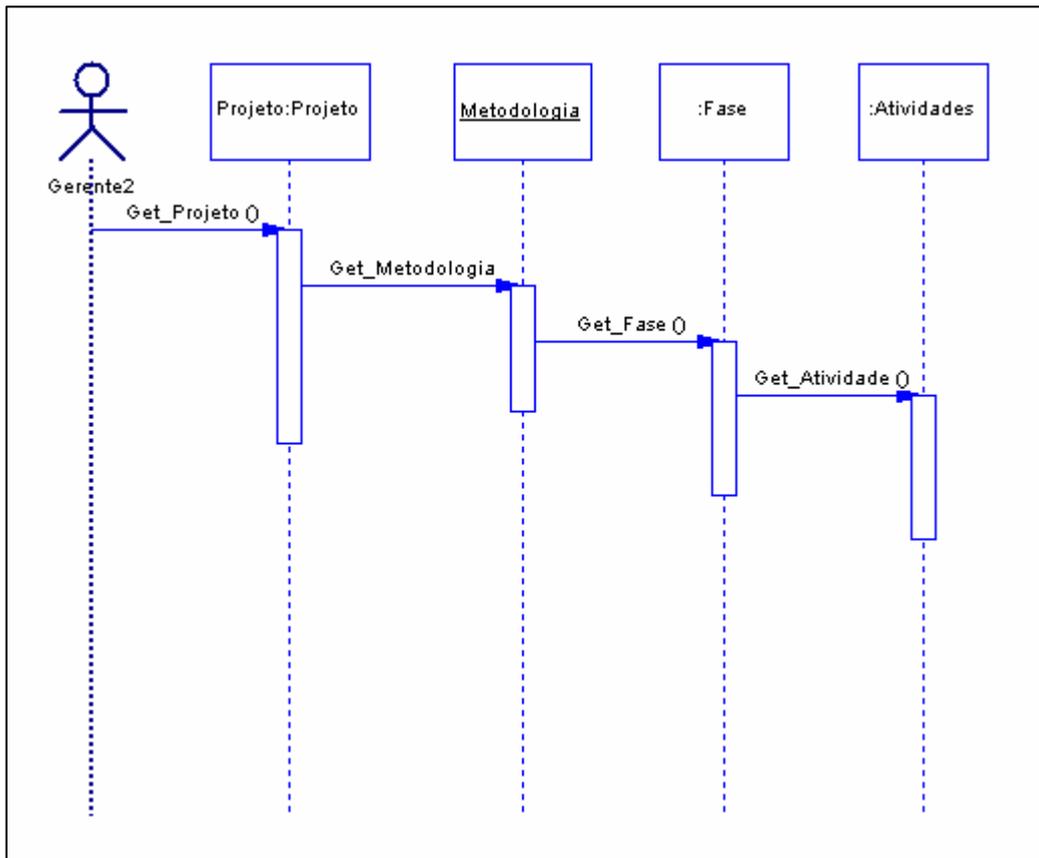
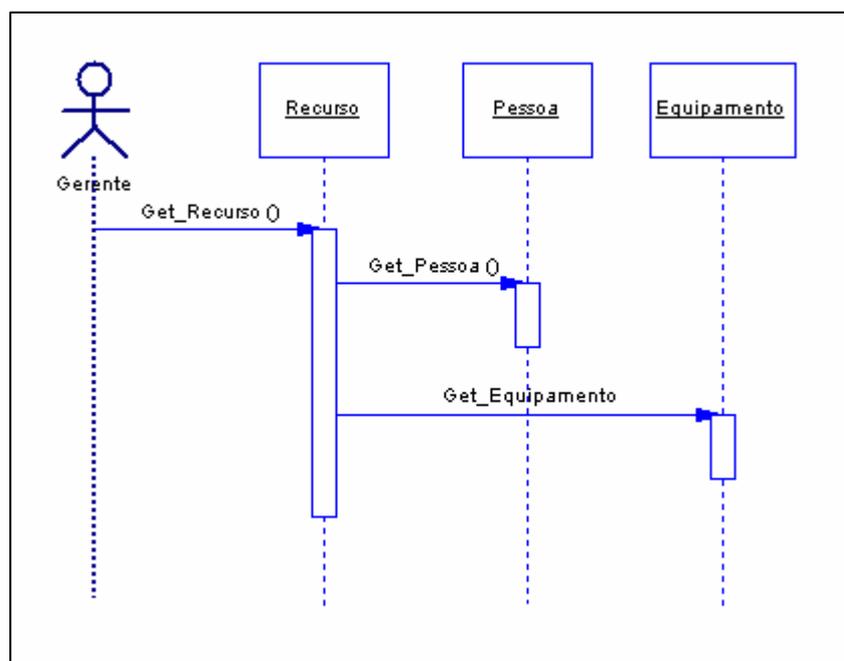


Figura 22 – Diagrama de Seqüência Emite Relatório da Descrição do Quadro de Recursos.



3.3 IMPLEMENTAÇÃO

Para o desenvolvimento do presente trabalho utilizou-se como ferramenta de desenvolvimento o ambiente *Visual Basic.NET* (MacDonald, 2002). O sistema gerenciador de banco de dados escolhido foi o *Microsoft Access 2000* (Callahan, 2000).

Para o mapeamento do diagrama de classes para o modelo físico de dados foi utilizada a ferramenta CASE *Power Designer 9*, que possibilita a geração do modelo físico a partir de um diagrama de classes e vice-versa. As tabelas foram geradas com as mesmas características das classes definidas no diagrama de classes do protótipo. Cada classe originou uma tabela no banco de dados, com os mesmos atributos. A figura 24 mostra o Diagrama Entidade-Relacionamento gerado a partir do diagrama de classes com a ferramenta *Power Designer*.

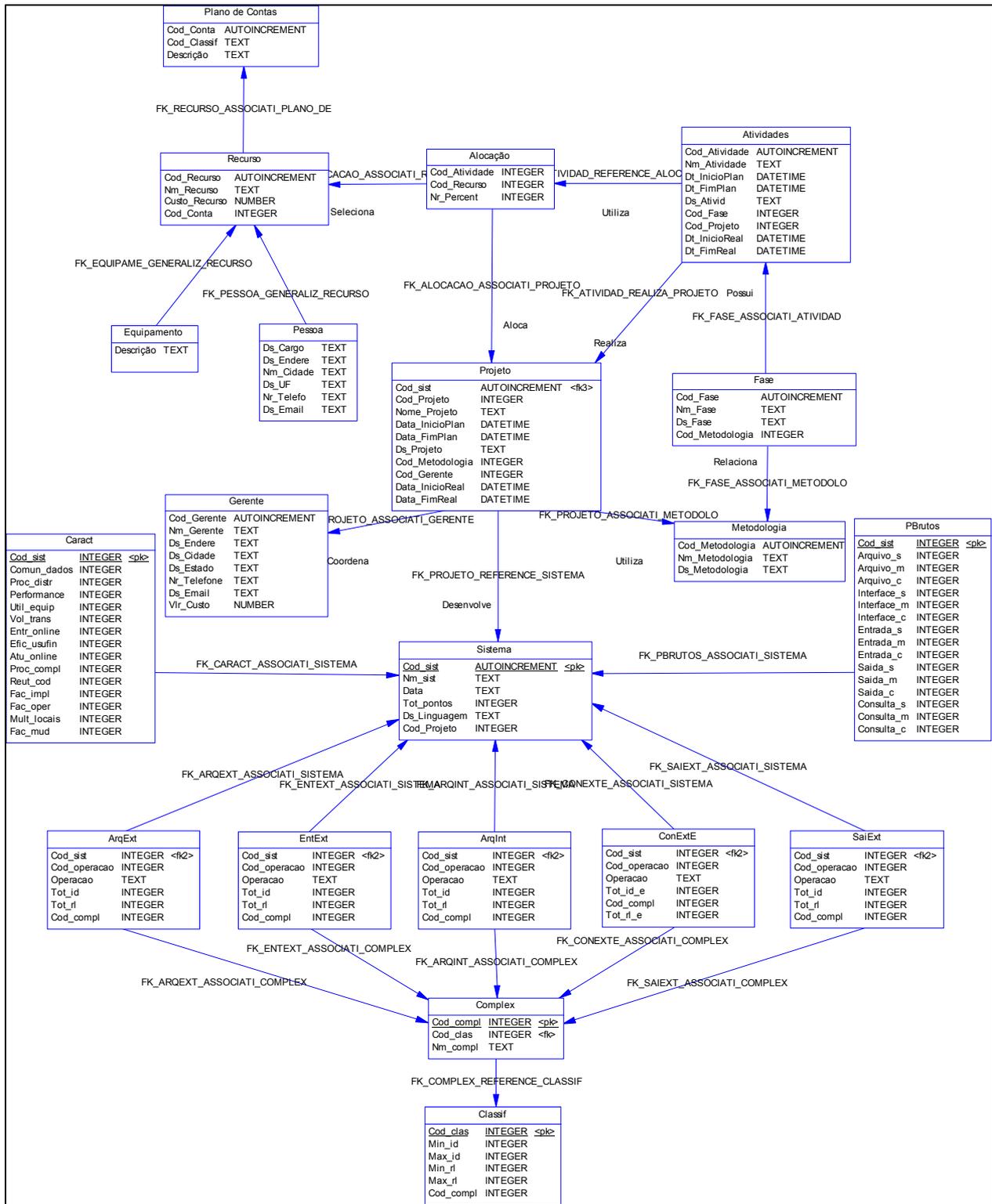
O *Power Designer 9* também possibilita a geração de código fonte básico (definição de atributos e métodos) para diversas linguagens, inclusive para o Visual Basic.NET. Desta forma, a geração do código fonte das classes pode ser feita diretamente a partir da ferramenta case. O anexo 1 descreve um exemplo de herança no VB.NET, citando a classe Pessoa, que herda os atributos e métodos da classe Recurso. Para fazer da classe Recurso uma subclasse Pessoa, é utilizada a palavra *Inherits*, conforme exemplificado na figura 23.

Figura 23 – Exemplo de herança no Visual Basic.NET.

```
Public Class Pessoa
    Inherits Recurso
    Private DS_CARGO As String
    Private DS_ENDERE As String
    Private NM_CIDADE As String
    Private DS_UF As String
    Private NR_TELEFO As String
    Private DS_EMAIL As String
```

A seguir serão demonstradas as principais telas do protótipo, seguidas de uma breve explicação do funcionamento das mesmas, através do uso de exemplos práticos.

Figura 24 – Diagrama Entidade-Relacionamento.



3.4 OPERACIONALIDADE DA IMPLEMENTAÇÃO

A seguir serão apresentadas algumas telas do sistema com suas respectivas funcionalidades.

A figura 25 mostra o Cadastro de Projetos, onde o gerente pode realizar o cadastro de um novo projeto ou editar as informações de um projeto pré-cadastrado.

Figura 25 – Cadastro de Projetos.

| CODIGO | PROJETO | GERENTE |
|--------|------------------------------|-------------------|
| 1 | Projeto Datawarehouse | Roberto Deschamps |
| 2 | Projeto Businnes Inteligence | Marcel Hugo |
| 6 | Projeto Fábrica de Software | Roberto Deschamps |
| 12 | Projeto OLAP | Roberto Deschamps |

Na figura 26, pode-se visualizar o Cadastro de Metodologias. Nesta tela, o gerente de projetos pode cadastrar uma metodologia de desenvolvimento, que poderá ser selecionada posteriormente como metodologia padrão para o desenvolvimento de um projeto. A partir da metodologia, serão definidas suas fases, na tela de Cadastro de Fases.

Figura 26 – Cadastro de Metodologias.

| Código | Nome | Descrição |
|--------|--------------|------------------------|
| 1 | Cascata | Método em Cascata |
| 2 | Prototipação | Método da Prototipação |
| 3 | Espiral | Metodologia Modelo em |

Um outro cadastro disponibilizado pelo sistema é o Cadastro de Gerentes (figura 27). O Cadastro de Gerentes possui as informações referentes aos gerentes de projetos, como nome, endereço, telefone, e-mail, etc. Para o sistema proposto, cada projeto pode possuir apenas um gerente.

A figura 28 ilustra o Cadastro de Sistemas, onde o usuário poderá efetuar o cadastro de um sistema, ou alterações de seus dados. Um projeto pode possuir vários sistemas, mas um sistema pode estar relacionado a apenas um projeto.

No Cadastro de Atividades o usuário poderá cadastrar as atividades referentes a um projeto, informando o nome, a descrição e as datas planejadas e/ou realizadas para a execução da atividade, além de selecionar a qual fase da metodologia do projeto esta atividade está relacionada. O Cadastro de Atividades pode ser visualizado na figura 29.

Figura 27 – Cadastro de Gerentes.

Cadastro de Gerentes

Código: 1 Nome do Gerente: Lindolfo Pereira Junior
 Vlr Custo: 650 Endereço: Rua José Arnaldo Pamplona, 122
 Cidade: Blumenau Estado: RS
 Telefone: (047)323-4039 E-mail: lindolfo.junior@lince.com.br

| | Código | Nome | Endereço | Cidade | UF |
|---|--------|-------------------------|---------------|----------|----|
| | 3 | Roberto Deschamps | Rua Anfilóqui | Gaspar | AC |
| | 4 | José Roque | Rua das Cabr | Blumenau | SC |
| ▶ | 1 | Lindolfo Pereira Junior | Rua José Arn | Blumenau | RS |
| | 2 | Marcel Hugo | Rua São Paul | Blumenau | SC |

Figura 28 – Cadastro de Sistemas.

Cadastro de Sistemas

Código: 2 Nome do Sistema: DecisionMaker
 Projeto: Projeto Businnes Inteligence
 Data de Início: 1/1/2003 Data de Conclusão: 31/12/2003 Linguagem: JAVA

Descrição:
 Sistema para planejamento gerencial

| | Código | Nome | Início | Final |
|---|--------|-------------|-----------|------------|
| ▶ | 2 | DecisionMak | 1/1/2003 | 31/12/2003 |
| | 3 | OLAP Report | 16/2/2003 | 25/10/2003 |

Figura 29 – Cadastro de Atividades.

Cadastro de Atividades

Projeto:

Código: Atividade:

Descrição:

Fase:

Planejado: Data de Início: Data de Conclusão:

Realizado: Data de Início: Data de Conclusão:

| | Código | Atividade | Descrição | Fase |
|---|--------|---------------|----------------|---------------|
| | 15 | Validação | Validação | Especificação |
| | 2 | Desenho de | Desenho das | Testes |
| ▶ | 16 | Especificação | Especificação | Projeto |
| | 17 | Projeto | Projeto de tel | Projeto |
| | 7 | Teste | Teste | Projeto |

Para a execução das atividades do projeto, normalmente são necessários alguns recursos, que podem ser pessoas ou equipamentos. Para o cadastro destes recursos, o sistema disponibiliza uma tela de Cadastro de Recursos (figura 30). Nesta tela, o usuário irá informar qual o tipo de recurso (Pessoa ou Equipamento) e em seguida cadastrar as informações referentes a cada tipo de recurso. Para o cadastro do recurso o usuário precisa também selecionar a classificação contábil em que o recurso está enquadrado. Esta classificação contábil pode ser cadastrada através da tela de Cadastro de Contas.

A figura 31 exibe a tela de Alocação de Recursos, que relaciona um recurso a uma determinada atividade. Para isto, o usuário deve selecionar o projeto, e em seguida a atividade e o recurso que deseja alocar, informando também o percentual que deseja alocar para aquela atividade.

Figura 30 – Cadastro de Recursos.

Cadastro de Recursos

Tipo
 Pessoa
 Equipamento

Código: 3 Nome do Recurso: João Paulo Ferreira
 Descrição:
 Custo: R\$ 1.200,00 Endereço: Rua São Paulo, 1535
 Cidade: Blumenau Estado: SC Cargo: Analista de Sistemas
 Telefone: (0047)326-8954 E-mail: joao@inet.com.br
 Classificação Contábil: 1.1.1 - Bancos

| Código | Nome | Custo | Descrição | Endereço |
|--------|--------------|-------|----------------|--------------|
| 3 | João Paulo F | 1200 | Analista de Si | Rua São Paul |

Figura 31 – Alocação de Recursos.

Alocação de Recursos

Projeto: Projeto Business Intelligence

Atividade: Desenho de Telas
 Recurso: Scanner HP 3600P
 % Alocado: 25

| Cod_Ativ | Cod_Rec | Cod_Proj | Atividade | Recurso | Projeto |
|----------|---------|----------|---------------|--------------|---------|
| 1 | 1 | 2 | Especificação | Computador | Projeto |
| 2 | 1 | 2 | Desenho de | Computador | Projeto |
| 15 | 3 | 2 | Validação | João Paulo F | Projeto |
| 5 | 4 | 2 | Teste 2 | Scanner HP 3 | Projeto |
| 2 | 4 | 2 | Desenho de | Scanner HP 3 | Projeto |

O PMBOK define como uma das ferramentas e técnicas para Estimativa dos Custos a Modelagem Paramétrica, que neste protótipo é representada por Pontos de Função. A figura 32 mostra a tela de Classificação das Funções, onde o usuário poderá inserir as funções para cada componente. Na figura 33, pode-se visualizar a tela de cadastro das Características Gerais do Sistema, onde o usuário deve informar os níveis de influência para cada item de influência.

Figura 32 – Classificação das Funções.

The screenshot shows a window titled "Classificação das Funções". At the top, there are dropdown menus for "Projeto" (Projeto Business Intelligence) and "Sistema" (OLAP Report), along with a "Selecionar" button. Below this, there are input fields for "Código" (7), "Função" (Exclusão de Métodos), "Ítems de dados ref." (3), "Reg. Lógicos" (3), and "Complexidade" (Simple). To the right of these fields are buttons for "Inserir", "Salvar", "Excluir", and "Limpar". Below the form is a tabbed interface with tabs for "Arquivo Lógico Interno", "Arquivo Interface Externo", "Entradas Externas", "Saídas Externas", and "Consultas Externas". The "Arquivo Lógico Interno" tab is active, showing a table with the following data:

| Código | Função | Itens | Registros | Complexidad |
|--------|----------------------|-------|-----------|-------------|
| 5 | Inclusão de Métodos | 3 | 3 | Simple |
| 7 | Exclusão de Métodos | 3 | 3 | Simple |
| 8 | Alteração de Métodos | 3 | 3 | Simple |

At the bottom of the window is a "Fechar" button.

Figura 33 – Características Gerais do Sistema.

The screenshot shows a window titled "Características Gerais do Sistema". At the top, there are dropdown menus for "Projeto" (Projeto Business Intelligence) and "Sistema" (OLAP Report), along with a "Selecionar" button. Below this, there are two columns of characteristics, each with a "Peso" (Weight) dropdown menu. The characteristics and their weights are:

| Características | Peso | Características | Peso |
|---------------------------------|---------------------|--------------------------------|--------------------------|
| 1 - Comunicação de Dados | Grande influência | 8 - Atualização On-line | Influência significativa |
| 2 - Processo Distribuído | Grande influência | 9 - Processamento Complexo | Influência média |
| 3 - Performance | Grande influência | 10 - Reutilização do Código | Influência moderada |
| 4 - Utilização do Equipamento | Grande influência | 11 - Facilidade de Implantação | Influência mínima |
| 5 - Volume de Transações | Influência mínima | 12 - Facilidade Operacional | Influência significativa |
| 6 - Entrada de Dados Online | Influência moderada | 13 - Múltiplos Locais | Grande influência |
| 7 - Eficiência do Usuário Final | Influência média | 14 - Facilidade de Mudanças | Grande influência |

At the bottom of the window are buttons for "Salvar", "Cancelar", and "Fechar".

Na tela de Cálculo de Pontos de Função, o usuário pode visualizar os cálculos de pontos de função brutos, nível de influência, fator de ajuste e pontos de função ajustados, além de relacionar o total de pontos para cada tipo de função (figura 34).

Figura 34 – Cálculo de Pontos de Função.

| Tipo de Função | Complexidade | Total |
|----------------|--------------|-------------|
| Arquivo | Simple | 1 × 7 = 7 |
| | Média | 2 × 10 = 20 |
| | Complexa | 3 × 15 = 45 |
| Interface | Simple | 4 × 5 = 20 |
| | Média | 5 × 7 = 35 |
| | Complexa | 6 × 10 = 60 |
| Entrada | Simple | 7 × 3 = 21 |
| | Média | 8 × 4 = 32 |
| | Complexa | 9 × 6 = 54 |
| Saída | Simple | 0 × 4 = 0 |
| | Média | 1 × 5 = 5 |
| | Complexa | 2 × 7 = 14 |
| Consulta | Simple | 3 × 3 = 9 |
| | Média | 4 × 4 = 16 |
| | Complexa | 5 × 6 = 30 |

Pontos de Função Brutos
368

Nível de Influência
50

Fator de Ajuste
FA = (NI * 0,01) + 0,65
1,15

Pontos de Função Ajustados
PF Ajustados = PF Brutos * Fator de Ajuste
423.2

Calcular Imprimir

Fechar

O sistema disponibiliza também diversos relatórios. Entre eles, o Relatório de Pontos de Função, Alocação de Recursos, Quadro de Recursos, Estrutura Analítica de Trabalho, Estimativas de Duração das Atividades e um Gráfico de Custos Gerais dos Projetos. Um exemplo de interface de visualização dos relatórios pode ser visualizado na figura 35, com o Relatório de Alocação de Recursos Analítico, e também na figura 36, com o Gráfico de Custos. O anexo 3 contém alguns exemplos de relatórios disponibilizados pelo protótipo.

Figura 35 – Visualização de Relatórios.

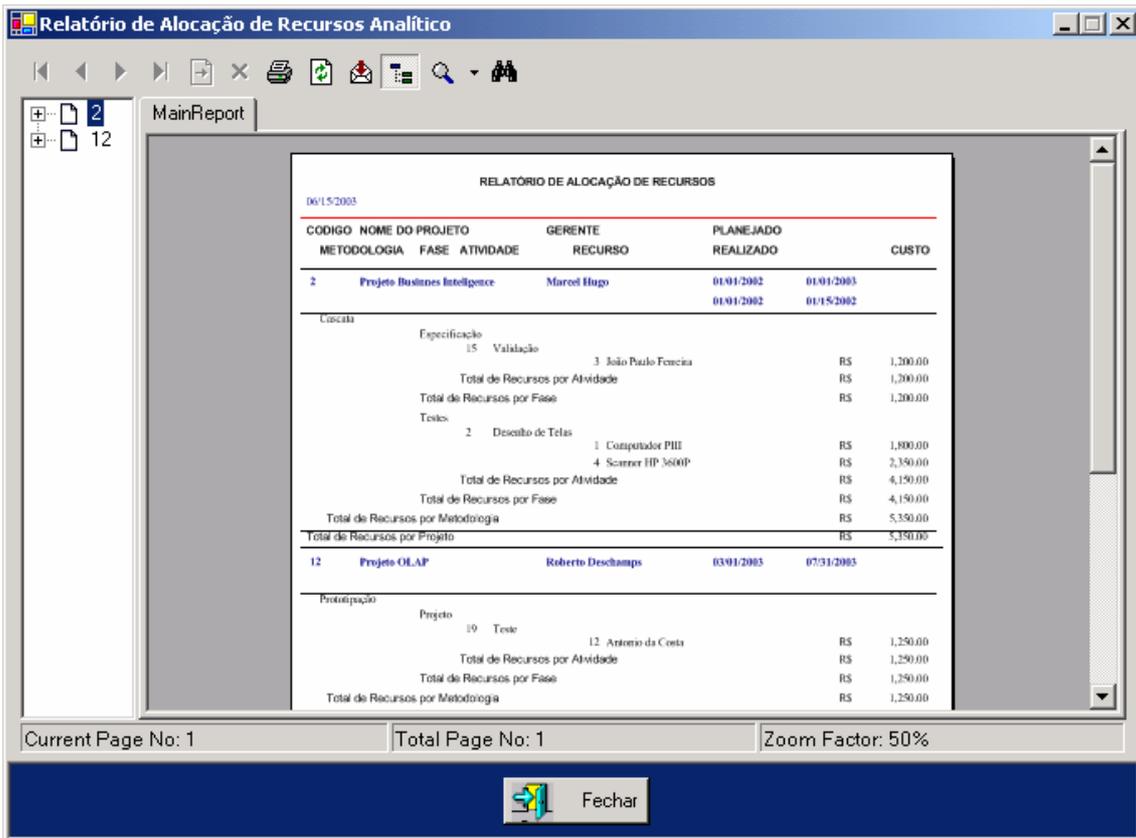
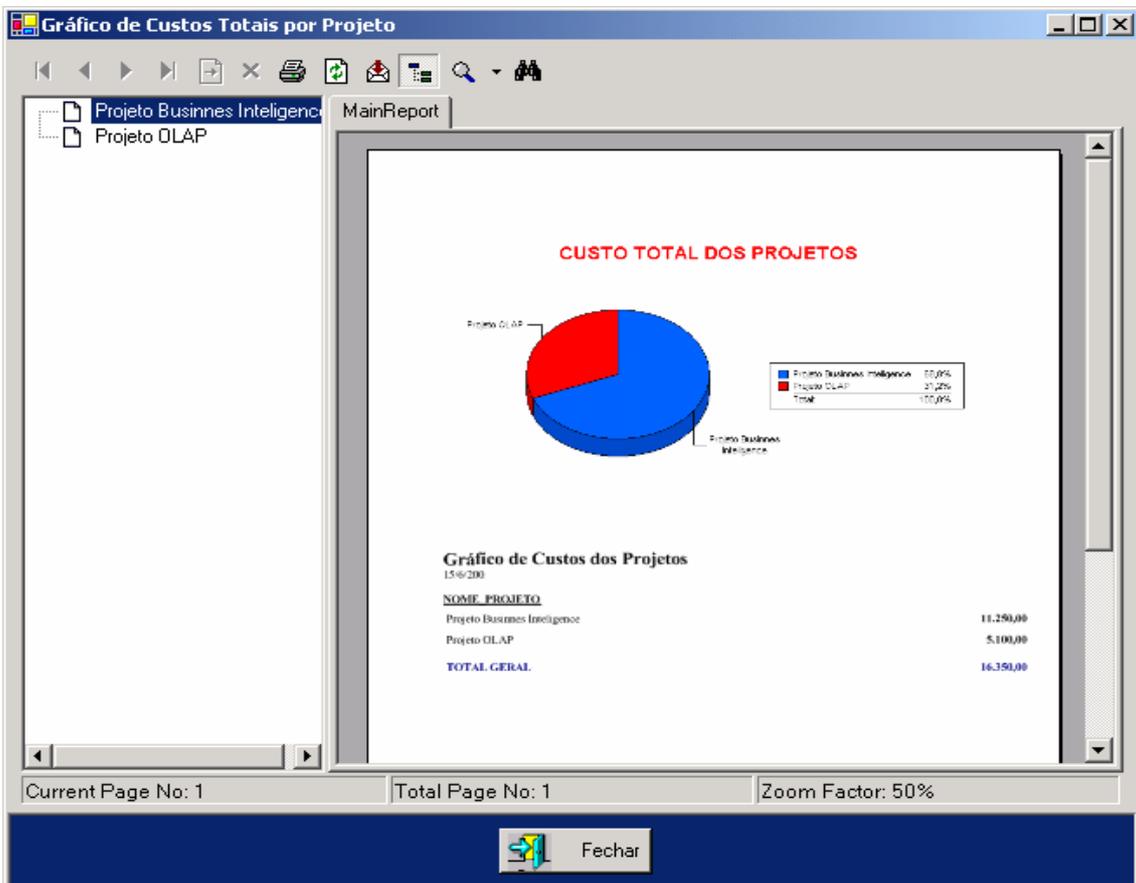


Figura 36 – Gráfico de Custos.



A figura 37 destaca os itens da área de Gerência de Custos do PMBOK que foram abordados pelo protótipo desenvolvido.

Figura 37 – Itens da Gerência de Custos do Projeto abordados pelo protótipo.



4 CONCLUSÕES

Durante o desenvolvimento deste trabalho foram apresentados os principais conceitos e características do Gerenciamento de Projetos, de acordo com o modelo definido pelo PMI, o *Project Management Body of Knowledge* (PMBOK). De forma mais aprofundada, foram estudadas as técnicas de Gerência de Custos definidas pelo PMBOK, sendo esta área de conhecimento a base para especificação e desenvolvimento do protótipo.

O objetivo do gerenciamento de custos de projetos é assegurar que o projeto possa ser executado dentro do orçamento aprovado. As atividades da Gerência de Custos iniciam-se com o Planejamento dos Recursos, Estimativas de Custos, Orçamentação dos Custos e Controle dos Custos. Estas atividades consistem fundamentalmente em gerenciar os custos dos recursos necessários à implementação das atividades do projeto.

Neste trabalho, foram identificadas todas as atividades da Gerência de Custos, e a maioria delas foi implementada no protótipo, exceto por alguns itens do processo de controle dos custos, e também por alguns outros itens como, por exemplo, controle de riscos, pois já foi abordado em outros trabalhos. Para atender ao item Modelagem Paramétrica definida no processo de Estimativa de Custos, foi utilizada a técnica de Análise de Pontos de Função (FPA), que é uma unidade de medida que procura definir o tamanho do que faz o software, independente de como possa ser produzido e implementado. Esta técnica utiliza características do projeto para criar modelos matemáticos que auxiliam a prever seus custos.

O ambiente de programação *Visual Basic.NET* atendeu plenamente às necessidades de implementação tendo em vista os objetivos do trabalho, assim como o *Microsoft Access 2000*, que foi utilizado como sistema gerenciador de banco de dados.

Em relação aos principais objetivos do trabalho (desenvolvimento de um protótipo de ferramenta para gerenciamento de projetos de software), e também aos objetivos específicos (gerenciamento das estimativas de custo em projetos de software através do acompanhamento das atividades do projeto, controle dos recursos alocados, geração de gráficos e relatórios de acompanhamento do projeto e do uso de métricas de software), conclui-se que estes, foram alcançados, pois o sistema disponibiliza

informações importantes aos gerentes de projeto, com efetivo auxílio no processo de gerenciamento de custos em projetos de software.

O protótipo possui também potencial didático, pois aborda assuntos como gerenciamento de projetos e métricas de software, podendo desta forma servir como material de apoio em disciplinas de engenharia de software, gerência de projetos, etc.

4.1 EXTENSÕES

A fim de continuar a abordagem aqui utilizada vários aspectos poderiam vir a ser trabalhados. A seguir é apresentada uma lista com algumas sugestões para o desenvolvimento de trabalhos futuros:

- a) Avaliar, através de pesquisas, em que medida o mercado atual vem adotando as práticas de gerenciamento de projetos para o desenvolvimento de software, e desenvolver um estudo das outras áreas de conhecimento e atuação gerencial do PMBOK, ou de outros modelos de gerenciamento de projetos, desenvolvendo um protótipo que auxilie na execução de tais atividades;
- b) Verificar possível convergência entre os conceitos de gerenciamento de projetos descritos pelo PMI e ISO, comparando os dois modelos e selecionando as melhores características dos dois modelos para desenvolver um modelo de gerenciamento de projetos de software;
- c) Desenvolver um adendo ao PMBOK referente às peculiaridades relacionadas com o gerenciamento de processos de desenvolvimento de *software*, seguindo da implementação de um tutorial para utilização do modelo desenvolvido;
- d) Aprimorar o protótipo a partir da inclusão de novas funções envolvendo outras atividades do gerenciamento de custos de software, ou utilizando outras métricas de software para modelagem paramétrica.

Apesar de ser uma área ainda pouco explorada, existe uma forte tendência ao surgimento de novos trabalhos na área de gerenciamento de projetos de software. Isso pode ser observado em publicações realizados nos últimos anos, que demonstram a necessidade das empresas de buscarem alternativas para gerenciar melhor seus projetos.

ANEXO 1 – EXEMPLO DE CÓDIGO FONTE

```
Imports System
Imports System.Data
Imports System.Data.OleDb

Public Class Recurso
    Private COD_RECURSO As Integer
    Private COD_CONTA As Integer
    Private NM_RECURSO As String
    Private CUSTO_RECURSO As String
    Public mErr As String
    Public SQL_CONNECTION_STRING = Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=C:\Proger\DbCustos.mdb"

    Public Property CODRECURSO() As Integer
        Get
            Return COD_RECURSO
        End Get
        Set(ByVal Value As Integer)
            COD_RECURSO = Value
        End Set
    End Property

    Public Property CODCONTA() As Integer
        Get
            Return COD_CONTA
        End Get
        Set(ByVal Value As Integer)
            COD_CONTA = Value
        End Set
    End Property

    Public Property NMRECURSO() As String
        Get
            Return NM_RECURSO
        End Get
        Set(ByVal Value As String)
            NM_RECURSO = Value
        End Set
    End Property

    Public Property CUSTORECURSO() As String
        Get
            Return CUSTO_RECURSO
        End Get
        Set(ByVal Value As String)
            CUSTO_RECURSO = Value
        End Set
    End Property
End Class
```

```
Imports System
Imports System.Data
Imports System.Data.OleDb

Public Class Pessoa
    Inherits Recurso
    Private DS_CARGO As String
    Private DS_ENDERE As String
    Private NM_CIDADE As String
    Private DS_UF As String
    Private NR_TELEFO As String
    Private DS_EMAIL As String
    Public TabelaPessoa As New DataTable

    Public Property DSCARGO() As String
        Get
            Return DS_CARGO
        End Get
        Set(ByVal Value As String)
            DS_CARGO = Value
        End Set
    End Property
    Public Property DSENDERE() As String
        Get
            Return DS_ENDERE
        End Get
        Set(ByVal Value As String)
            DS_ENDERE = Value
        End Set
    End Property
    Public Property NMCIDADE() As String
        Get
            Return NM_CIDADE
        End Get
        Set(ByVal Value As String)
            NM_CIDADE = Value
        End Set
    End Property
    Public Property DSUF() As String
        Get
            Return DS_UF
        End Get
        Set(ByVal Value As String)
            DS_UF = Value
        End Set
    End Property
    Public Property NRTELEFO() As String
        Get
            Return NR_TELEFO
        End Get
        Set(ByVal Value As String)
            NR_TELEFO = Value
        End Set
    End Property
    Public Property DSEMAIL() As String
        Get
            Return DS_EMAIL
        End Get
        Set(ByVal Value As String)
            DS_EMAIL = Value
        End Set
    End Property
End Class
```



```

Public Sub Alterar(ByVal CODRECURSO As Integer, ByVal NMRECURSO As
String, ByVal CUSTORECURSO As String, ByVal DSCARGO As String, ByVal
DSENDERE As String, ByVal NMCIDADE As String, ByVal DSUF As String,
ByVal NRTELEFO As String, ByVal DSEMAIL As String, ByVal CODCONTA As
Integer)
    Dim conn As New OleDbConnection(SQL_CONNECTION_STRING)
    Dim cmd As New OleDbCommand
    Try
        conn.Open()
        cmd.Connection = conn
        cmd.CommandTimeout = 0
        cmd.CommandText = "UPDATE RECURSO SET NM_RECURSO =
'" & NMRECURSO & "', CUSTO_RECURSO = '" & CUSTORECURSO & "',
COD_CONTA = '" & CODCONTA & "' where COD_RECURSO = " &
CODRECURSO
        cmd.CommandType = CommandType.Text
        cmd.ExecuteNonQuery()
        cmd.CommandText = "UPDATE PESSOA SET DS_CARGO = '"
& DSCARGO & "', DS_ENDERE = '" & DSENDERE & "', NM_CIDADE = '"
& NMCIDADE & "', DS_UF = '" & DSUF & "', NR_TELEFO = '" &
NRTELEFO & "', DS_EMAIL = '" & DSEMAIL & "' where COD_RECURSO
= " & CODRECURSO
        cmd.CommandType = CommandType.Text
        cmd.ExecuteNonQuery()
        conn.Close()
    Catch ex As Exception
        mErr = mErr.ToString
        If conn.State.Open Then
            conn.Close()
        End If
    End Try
End Sub

Public Sub Get_All_Pessoa()
    Dim Cnn As New OleDbConnection(SQL_CONNECTION_STRING)
    Dim Cmd As New OleDbCommand
    Dim PessoaAdapter As New OleDbDataAdapter
    Try
        Cnn.Open()
        Cmd.Connection = Cnn
        Cmd.CommandTimeout = 0
        Cmd.CommandText = "select RECURSO.COD_RECURSO as
Código, RECURSO.NM_RECURSO as Nome, RECURSO.CUSTO_RECURSO as
Custo, PESSOA.DS_CARGO as Descrição," & _
"PESSOA.DS_ENDERE as Endereço, PESSOA.DS_UF as UF,
PESSOA.NR_TELEFO as Telefone," & _
"PESSOA.DS_EMAIL as Email from RECURSO, PESSOA WHERE
RECURSO.COD_RECURSO = PESSOA.COD_RECURSO"
        Cmd.CommandType = CommandType.Text
        TabelaPessoa.Clear()
        Cmd.ExecuteNonQuery()
        PessoaAdapter.SelectCommand = Cmd
        PessoaAdapter.Fill(TabelaPessoa)
        Cnn.Close()
    Catch mExceptionTabelaPessoa As System.Exception
        mErr = mExceptionTabelaPessoa.ToString
        If Cnn.State.Open Then
            Cnn.Close()
        End If
    End Try
End Sub

```

```

Public Sub Get_Pessoa(ByVal Cod_ID As String)
    Dim Cnn As New OleDbConnection(SQL_CONNECTION_STRING)
    Dim Cmd As New OleDbCommand
    Dim DataReader As OleDbDataReader
    With Cmd
        .Connection = Cnn
        .CommandTimeout = 0
        .CommandText = "SELECT RECURSO.COD_RECURSO,
RECURSO.COD_CONTA, RECURSO.NM_RECURSO, RECURSO.CUSTO_RECURSO,
PESSOA.DS_CARGO, PESSOA.DS_ENDERE, " & _
"PESSOA.NM_CIDADE, PESSOA.NR_TELEFO, PESSOA.DS_EMAIL,
PESSOA.DS_UF FROM PESSOA, RECURSO WHERE RECURSO.COD_RECURSO =
" & Cod_ID & " and RECURSO.COD_RECURSO = PESSOA.COD_RECURSO"
        .CommandType = CommandType.Text
    End With
    Try
        Cnn.Open()
        DataReader = Cmd.ExecuteReader
        While DataReader.Read
            If Not IsDBNull(DataReader.Item("COD_RECURSO")) Then
                CODRECURSO = DataReader.Item("COD_RECURSO")
            End If
            If Not IsDBNull(DataReader.Item("COD_CONTA")) Then
                CODCONTA = DataReader.Item("COD_CONTA")
            End If
            If Not IsDBNull(DataReader.Item("NM_RECURSO")) Then
                NMRECURSO = DataReader.Item("NM_RECURSO")
            End If
            If Not IsDBNull(DataReader.Item("CUSTO_RECURSO")) Then
                CUSTORECURSO = DataReader.Item("CUSTO_RECURSO")
            End If
            If Not IsDBNull(DataReader.Item("DS_CARGO")) Then
                DSCARGO = DataReader.Item("DS_CARGO")
            End If
            If Not IsDBNull(DataReader.Item("DS_ENDERE")) Then
                DSENDERE = DataReader.Item("DS_ENDERE")
            End If
            If Not IsDBNull(DataReader.Item("NM_CIDADE")) Then
                NMCIDADE = DataReader.Item("NM_CIDADE")
            End If
            If Not IsDBNull(DataReader.Item("DS_UF")) Then
                DSUF = DataReader.Item("DS_UF")
            End If
            If Not IsDBNull(DataReader.Item("NR_TELEFO")) Then
                NRTELEFO = DataReader.Item("NR_TELEFO")
            End If
            If Not IsDBNull(DataReader.Item("DS_EMAIL")) Then
                DSEMAIL = DataReader.Item("DS_EMAIL")
            End If
        End While
        Catch mError As System.Exception
            mErr = mError.ToString
            If Cnn.State.Open Then
                Cnn.Close()
            End If
        End Try
        Cnn.Close()
    End Sub

```

```
Public Sub Excluir(ByVal Cod_Id As String)
    Dim conn As New OleDbConnection(SQL_CONNECTION_STRING)
    Dim cmd As New OleDbCommand
    Try
        conn.Open()
        cmd.Connection = conn
        cmd.CommandTimeout = 0
        cmd.CommandText = "DELETE FROM RECURSO WHERE COD_RECURSO = "
        " & Cod_Id
        cmd.CommandType = CommandType.Text
        cmd.ExecuteNonQuery()
        cmd.CommandText = "DELETE FROM PESSOA WHERE COD_RECURSO = "
        " & Cod_Id
        cmd.CommandType = CommandType.Text
        cmd.ExecuteNonQuery()
        conn.Close()
    Catch ex As Exception
        mErr = mErr.ToString
        If conn.State.Open Then
            conn.Close()
        End If
    End Try
End Sub
End Class
```

ANEXO 2 – EXEMPLOS DE RELATÓRIOS DISPONÍVEIS NO PROTÓTIPO.

| RELATÓRIO DE ALOCAÇÃO DE RECURSOS | | | | | | |
|-----------------------------------|-------------------------------------|-------------------------------------|------------------------------------|-------------------|-------------------|---------------|
| 06/01/2003 | | | | | | |
| CODIGO | NOME DO PROJETO | | GERENTE | PLANEJADO | | CUSTO |
| METODOLOGIA | FASE | ATIVIDADE | RECURSO | REALIZADO | | |
| 2 | Projeto Businnes Inteligence | | Marcel Hugo | 01/01/2002 | 01/01/2003 | |
| | | | | 01/01/2002 | 01/15/2002 | |
| Cascata | | | | | | |
| | | Análise de Requisitos | | | | |
| | | 15 Validação | 3 João Paulo Ferreira | | | R\$ 1,200.00 |
| | | Total de Recursos por Atividade | | | | R\$ 1,200.00 |
| | | Total de Recursos por Fase | | | | R\$ 1,200.00 |
| | | Análise | | | | |
| | | 2 Desenho de Telas | 1 Computador PIII | | | R\$ 1,800.00 |
| | | | 4 Scanner HP 3600P | | | R\$ 2,350.00 |
| | | | 12 Antonio da Costa | | | R\$ 1,250.00 |
| | | Total de Recursos por Atividade | | | | R\$ 5,400.00 |
| | | 16 Especificação | 14 Antonio da Silva | | | R\$ 1,500.00 |
| | | Total de Recursos por Atividade | | | | R\$ 1,500.00 |
| | | Total de Recursos por Fase | | | | R\$ 6,900.00 |
| | | Testes | | | | |
| | | 7 Teste | 1 Computador PIII | | | R\$ 1,800.00 |
| | | | 3 João Paulo Ferreira | | | R\$ 1,200.00 |
| | | Total de Recursos por Atividade | | | | R\$ 3,000.00 |
| | | Total de Recursos por Fase | | | | R\$ 3,000.00 |
| | | Total de Recursos por Metodologia | | | | R\$ 11,100.00 |
| | | Total de Recursos por Projeto | | | | R\$ 11,100.00 |
| 12 | Projeto OLAP | | José Roque | 03/01/2003 | 07/31/2003 | |
| Prototipação | | | | | | |
| | | Avaliação do Protótipo pelo cliente | | | | |
| | | 19 Treinamento dos usuários | 12 Antonio da Costa | | | R\$ 1,250.00 |
| | | | 17 Microcomputador AMD K 7 850 Mhz | | | R\$ 1,200.00 |
| | | Total de Recursos por Atividade | | | | R\$ 2,450.00 |
| | | Total de Recursos por Fase | | | | R\$ 2,450.00 |
| | | Total de Recursos por Metodologia | | | | R\$ 2,450.00 |
| | | Total de Recursos por Projeto | | | | R\$ 2,450.00 |
| Total Geral: | | | | | | R\$ 13.550,00 |

1/6/2003

RELATÓRIO DE PONTOS DE FUNÇÃO POR PROJETO/SISTEMA**PROJETO:** Projeto Business Intelligence**SISTEMA:** DecisionMaker**PONTOS BRUTOS**

| | | | | | |
|------------------|----|---|----|---|-----------|
| Arquivo | 1 | X | 7 | = | 7 |
| | 2 | X | 10 | = | 20 |
| | 3 | X | 15 | = | 45 |
| Interface | 4 | X | 5 | = | 20 |
| | 5 | X | 7 | = | 35 |
| | 6 | X | 10 | = | 60 |
| Entrada | 7 | X | 3 | = | 21 |
| | 8 | X | 4 | = | 32 |
| | 9 | X | 6 | = | 54 |
| Saída | 10 | X | 4 | = | 40 |
| | 11 | X | 5 | = | 44 |
| | 12 | X | 7 | = | 84 |
| Consulta | 13 | X | 3 | = | 39 |
| | 14 | X | 4 | = | 56 |
| | 15 | X | 6 | = | 90 |

TOTAL DE PONTOS BRUTOS 627**CARACTERÍSTICAS GERAIS DO SISTEMA**

| | | | |
|------------------------------|--------------------------|-----------------------------|--------------------------|
| Comunicação de | Influência moderada | Atualizações On-line | Influência significativa |
| Processamento | Influência média | Processamento | Grande influência |
| Performance | Influência significativa | Reusabilidade | Influência mínima |
| Utilização do | Grande influência | Facilidade de | Influência moderada |
| Volume de | Influência mínima | Facilidade | Influência média |
| Entrada de Dados | Influência moderada | Múltiplos Locais | Influência significativa |
| Eficiência do Usuário | Influência média | Facilidade de | Grande influência |

NIVEL DE INFLUÊNCIA: NI = 44**FATOR DE AJUSTE 1,09**

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

PONTOS AJUSTADOS**(PONTOS BRUTOS * PONTOS AJUSTADOS)****683.43**

| ESTRUTURA ANALÍTICA DE TRABALHO | | | | |
|---------------------------------|-------------------------------------|--------------------|--------------------------------------|-------------------|
| 06/01/2003 | | | | |
| CODIGO | NOME DO PROJETO | GERENTE | PLANEJADO | REALIZADO |
| | METODOLOGIA | | | |
| | FASE | ATIVIDADE | | |
| 2 | Projeto Businnes Inteligence | Marcel Hugo | 01/01/2002 | 01/01/2003 |
| | | | 01/01/2002 | 01/15/2002 |
| | Cascata | | | |
| | Análise de Requisitos | | | |
| | | 15 | Validação | |
| | Análise | | | |
| | | 2 | Desenho de Telas | |
| | | 16 | Especificação | |
| | Testes | | | |
| | | 7 | Teste | |
| 12 | Projeto OLAP | José Roque | 03/01/2003 | 07/31/2003 |
| | Prototipação | | | |
| | Coleta dos Requisitos | | | |
| | | 21 | Levantamento dos requisitos | |
| | Construção do Protótipo | | | |
| | | 22 | Desenvolvimento das rotinas de carga | |
| | Avaliação do Protótipo pelo cliente | | | |
| | | 19 | Treinamento dos usuários | |
| | Refinamento do Protótipo | | | |
| | | 23 | Testes do sistema | |

CUSTO TOTAL DOS PROJETOS

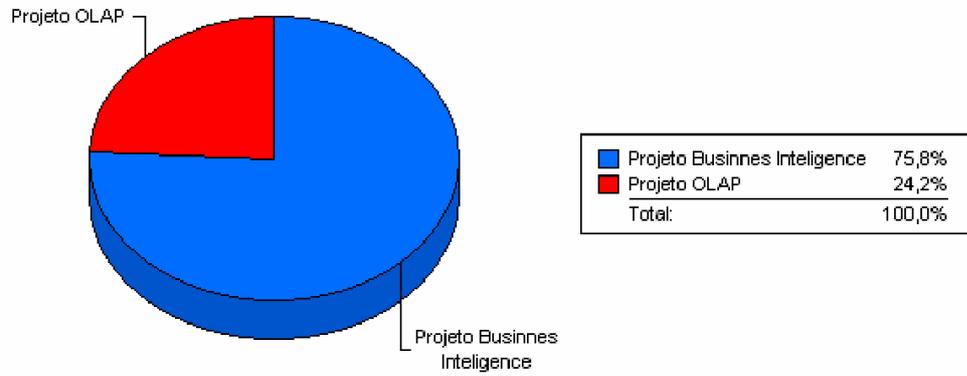


Gráfico de Custos dos Projetos

1/6/2003

NOME PROJETO

| | |
|-------------------------------|------------------|
| Projeto Business Intelligence | 12.350,00 |
| Projeto OLAP | 3.950,00 |
| TOTAL GERAL | 16.300,00 |

1/6/2003

DESCRIÇÃO DO QUADRO DE RECURSOS - EQUIPAMENTOS

| CÓDIGO | NOME DO RECURSO | DESCRIÇÃO | CUSTO |
|---------------|-----------------------------------|-----------------------------------|--------------|
| 1 | Computador PIII | Microcomputador Intel Pentium III | R\$ 1,800.00 |
| 2 | Impressora | Impressora HP 695C | R\$ 500.00 |
| 6 | Scanner | Scanner HP 3600P 1200x1200 Dpi | R\$ 1,650.00 |
| 17 | Microcomputador AMD K7 850 | Microcomputador AMD K7 850 | R\$ 1,200.00 |
| 18 | Servidor HP Intel Xeon 1.0 Ghz | 2 Processadores Intel 1.0 Ghz | R\$ 8,500.00 |
| 19 | Impressora Epson Stylus Color | Impressora Epson 1440 Dpi color | R\$ 560.00 |
| 20 | Microcomputador Intel Pentium 4 | Processador Intel Pentium 4 2.2 | R\$ 2,500.00 |
| 21 | Microcomputador Intel Pentium III | Processador Intel Pentium III 933 | R\$ 1,600.00 |

TOTAL GERAL:

R\$ 18,310.00

[RecursPessoal.rpt](#)

REFERÊNCIAS BIBLIOGRÁFICAS

BAUER, Corina. **Protótipo de software de apoio a identificação e análise de riscos e software**. 2001. 50 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

BFPUG. **Introdução ao International Function Point User Group**. 2001. Disponível em: < <http://www.bfpug.com.br/artigos/introifpug.zip> > Acesso em: 15 abr. 2003.

BRAGA, A. **Análise de pontos de função**. Rio de Janeiro: IBPI, 1996. 188 p.

CALLAHAN, E. **Microsoft Access 2000: Visual Basic for Applications: fundamentos**. São Paulo: Makron Books, 2000. 359 p.

CHOOSE - CHOOSE TECHNOLOGIES. **Metodologia para uso de métrica de pontos por função**. 2000. Disponível em: <<http://www.choose.com.br/servicos/metricas.htm>>. Acesso em: 25 mar 2003.

DEKKERS, Carol A. **Desmistificando Pontos de Função: entendendo a terminologia**. IT Metrics Strategies, October 1998. Disponível em: < <http://www.bfpug.com.br/Artigos/Desmistificando%20Pontos%20de%20Função.pdf> > Acesso em: 12 abr. 2003.

DEMARCO, Tom. **Controle de Projetos de Software**. Editora Campus, 1991.

FERNANDES, Aguinaldo Aragon. **Gerência de Software Através de Métricas**, São Paulo: Atlas, 1995.

GIBBS W. *Software's chronic crisis*. **Scientific American Magazine**, set. 1994. Disponível em: <<http://www.di.ufpe.br/~java/graduacao/referencias/SciAmSept1994.html>>. Acesso em: 27 fev. 2003.

ISO. *ISO - International Organization for Standardization. Introduction to ISO*. 1999. Disponível em <<http://www.iso.ch>>. Acesso em: 23 mar 2003.

ISO. *ISO - International Organization for Standardization. ISO 10006 – Quality management - Guidelines to quality in project management*. Genebra. 1997.

JAEGER NETO, José Inácio. **CMM & PMBOK**. 2001. Disponível em: < <http://www.pucrs.br/inf/prosergs1.pdf> >. Acesso em: 15 jun 2003.

JONES, C. Software project management in the 21st century. **American Programmer**, New York, v. 11, n. 2, fev. 1999. Disponível em: <<http://www.spr.com/news/SoftwareProjectArticle.pdf>>. Acesso em: 27 fev. 2003.

KIENEN, Anísio. **Protótipo de uma ferramenta de software baseada na metodologia PERT/CPM para o planejamento de projetos de desenvolvimento de sistemas**. 2000. 63 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

LONZETTI, Ricardo. **Ferramenta case para métricas de projetos de software**. 2001. 91 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Educação Superior de Ciências Tecnológicas da Terra e do Mar, Universidade do Vale do Itajaí.

MACDONALD, M. **O livro de VB.NET**. Rio de Janeiro: Ciência Moderna, 2002. 459 p.

MANOEL, Marcus Teli. **Capability Maturity Model**. 2002. Disponível em <<http://mtelli.dominodeveloper.net/members/mtelli/home.nsf/CMM!OpenPage>>. Acesso em: 21 abr de 2003.

MARTINS, J. C. C. **Gestão de projetos de desenvolvimento de software PMI - UML**. Rio de Janeiro: Brasport, 2002. 189 p.

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

PAULK, M. et al. **Capability maturity model for software**, Pittsburgh, fev., 1993. Disponível em: <<http://www.sei.cmu.edu/publications/documents/93.reports/93.tr.024.html>>. Acesso em: 27 fev. 2003.

PETERS, James F. **Engenharia de software: Teoria e Prática**. Campus, 2001.

PMI- Project Management Institute. **PMBOK: project management body of knowledge**, Belo Horizonte, 2002. Disponível em: <www.pmimg.org.br>. Acesso em: 25 fev. 2002

PRADO, D. S. do. **Gerenciamento de projetos nas organizações**. Belo Horizonte: Editora de Desenvolvimento Gerencial, 2000. 203 p.

PRESSMAN, R. S. **Software engineering**: a practitioner's approach. New York: McGraw-Hill, 3 ed., 1992. 793 p.

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

ROCHA, Ana Regina Cavalcanti et al. **Qualidade de Software**. São Paulo: Prentice Hall, 2001.

SEI. *SEI - Software Engineering Institute*. History. 1999. Disponível em: <<http://www.sei.cmu.edu/about/overview/sei/history.html>>. Acesso em: 17 mar. 2003.

SILVA, I. J. de M. **Delphi 5**: análise de pontos de função. Rio de Janeiro: Book Express. 2000. 277 p.

STANDISH. **The standish group, "Chaos"**, Estados Unidos, 1995. Disponível em: <<http://www.standishgroup.com>>. Acesso em: 16 fev. 2003.

_____. **The standish group, "Micro Projects Cause Constant Change"**, Estados Unidos, 2000. Disponível em: <<http://www.standishgroup.com>>. Acesso em: 27 fev. 2003.

VALCANAI, Tibério César. **Protótipo de ferramenta de cálculo de FPA sobre Microsoft Access 97**. 1998. 126 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.