

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

**DESENVOLVIMENTO DE UMA MÉTRICA PARA ESTIMAR
UM PROJETO DE IMPLANTAÇÃO DE SOFTWARE DE
RECURSOS HUMANOS**

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

CARLOS GOUVEIA LIMA FILHO

BLUMENAU, JUNHO/2003.

2003/1-07

DESENVOLVIMENTO DE UMA MÉTRICA PARA ESTIMAR UM PROJETO DE IMPLANTAÇÃO DE SOFTWARE DE RECURSOS HUMANOS

CARLOS GOUVEIA LIMA FILHO

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. Maurício Capobianco Lopes

Prof. Evaristo Baptista

AGRADECIMENTOS

Agradeço ao meu orientador Marcel Hugo, pelo auxílio que me deu durante o desenvolvimento do trabalho.

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

A Deus, por estar presente em todos os momentos e me dando força nos momentos de dificuldade.

Agradeço a meus familiares por tudo que me ensinaram.

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

A todos os meus colegas de trabalho e de faculdade, que contribuíram para o meu crescimento e tornaram este período de faculdade inesquecível. E também contribuíram e muito na elaboração deste trabalho.

Agradeço também aos Amigos do Barney, que estiveram a meu lado durante toda a faculdade.

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

LISTA DE FIGURAS

Figura 1 - Áreas de conhecimento na Gerência de Projetos.....	26
Figura 2 - Caso de uso do sistema	38
Figura 3 – Diagrama de classes do software	39
Figura 4 – Diagrama de seqüência CalcularPontosImplantação	41
Figura 5 – Tela principal do sistema	42
Figura 6 – Tela de Cadastro de Clientes.....	42
Figura 7 – Tela para cadastrar estimativas	43
Figura 8 – Tela para adicionar módulos	44
Figura 9 – Tela do módulo Ponto e Acesso.....	44
Figura 10 – Relatório de Pontos de Implantação.....	45

LISTA DE QUADROS

Quadro 1 – Perguntas sobre a etapa de Levantamento Inicial.....	30
Quadro 2 – Questionamento sobre os módulos do Sistema de RH.....	30
Quadro 3 – Perguntas sobre a etapa de Instalação.....	31
Quadro 4 – Perguntas sobre o Módulo Folha de Pagamento	33
Quadro 5 – Perguntas sobre o módulo de controle de Ponto e Acesso.	35
Quadro 6 – Pergunta sobre o módulo RS	36
Quadro 7 – Pergunta sobre o módulo TR.....	36
Quadro 8 – Pergunta sobre o módulo CS.....	37
Quadro 9 – Perguntas sobre treinamento.....	37
Quadro 10 – Pergunta sobre Validação e Aceite.....	37
Quadro 11 – Validação do Software.....	46

LISTA DE ABREVIATURAS E SIGLAS

FPA	<i>Function Point Analysis</i>
IFPUG	<i>International Function Point User Group</i>
PI	Ponto de Implantação
PMBOK	<i>Project Management Body of Knowledge</i>
RH	Recursos Humanos
TI	Tecnologia de Informação
UML	<i>Unified Modeling Language</i>

SUMÁRIO

1	INTRODUÇÃO	11
1.1	OBJETIVOS	12
1.2	ESTRUTURA.....	13
2	ENGENHARIA DE SOFTWARE E FPA.....	14
2.1	ENGENHARIA DE SOFTWARE	14
2.2	FPA.....	15
2.2.1	HISTÓRICO	15
2.2.2	OBJETIVOS	16
2.2.3	TIPOS DE CÁLCULO DA TÉCNICA FPA.....	16
2.2.4	VISÃO GERAL DE UMA APLICAÇÃO.....	17
2.2.5	RESUMO DO CÁLCULO DE PONTOS POR FUNÇÃO	17
3	IMPLANTAÇÃO DE SOFTWARE	19
3.1	A IMPORTÂNCIA DA FASE DE IMPLANTAÇÃO DE UM SOFTWARE	19
3.1.1	A NOVA REALIDADE DOS CLIENTES E USUÁRIOS DE SOFTWARE	19
3.1.2	AS CIRCUNSTÂNCIAS QUE ENVOLVEM UMA IMPLANTAÇÃO DE SOFTWARE	19
3.1.3	ESFORÇOS NA DIREÇÃO DA MELHORIA DA QUALIDADE DA IMPLANTAÇÃO.....	20
3.1.4	O PROBLEMA DO CONTROLE.....	21
4	PMBOK.....	24
4.1	O QUE É GERÊNCIA DE PROJETOS?.....	24
4.2	CONTEXTO DE GERÊNCIA DE PROJETO.....	24
4.3	FASES E CICLO DE VIDA	25

4.4	PROCESSOS DE GERÊNCIA DE PROJETOS	25
4.4.1	GERÊNCIA DE TEMPO.....	26
4.4.2	GERÊNCIA DE CUSTO	27
5	MÉTRICA PARA IMPLANTAÇÃO.....	28
5.1	ETAPAS DA IMPLANTAÇÃO	28
5.2	QUESTIONÁRIO PARA CHECK LIST	29
5.2.1	LEVANTAMENTO INICIAL.....	29
5.2.2	INSTALAÇÃO	31
5.2.3	TREINAMENTO	37
5.2.4	VALIDAÇÃO E ACEITE	37
6	IMPLEMENTAÇÃO.....	38
6.1	ESPECIFICAÇÃO DO SISTEMA	38
6.1.1	CASOS DE USO	38
6.1.2	DIAGRAMA DE CLASSES	38
6.1.3	DIAGRAMAS DE SEQUÊNCIA	40
6.2	IMPLEMENTAÇÃO	41
6.2.1	TESTE DO SISTEMA.....	45
7	CONCLUSÕES	47
7.1	SUGESTÕES	47
	ANEXO I.....	49
	REFERÊNCIAS BIBLIOGRÁFICAS	54

RESUMO

Este trabalho tem por objetivo elaborar uma métrica para auxiliar na estimativa de custo e tempo para a implantação de um software de Recursos Humanos. Foi utilizado como base casos reais e técnicas de FPA e PMBOK. Foi criado um questionário que é adotado para o levantamento e obtenção do resultado, que serão os pontos de implantação por fases do projeto de implantação.

Palavras-chave: implantação, qualidade de software, projeto, engenharia de software, métrica.

ABSTRACT

This work has as objective to make produce and work out a metric to assist the cost's and time's estimate for the implant of a Human Resources' software. There were used as base the cases of real use and the assistance use of FPA and PMBOK's technics . It is intend to create a check-list and a questionnaire that will be used for the survey and to obtain the result, witch will be the implant points for this project's phase.

Key-words: implant, software quality, project, software engineering., metric.

1 INTRODUÇÃO

A implantação de sistemas é uma das etapas da engenharia de software que vem crescendo nos últimos tempos. Muitas empresas especializadas e novos métodos para a realização das atividades estão surgindo.

É comum encontrar referências sobre engenharia de software e suas etapas, mas há poucos casos onde seja destacada a fase de implantação. Existem empresas desenvolvedoras de softwares, que direcionam esforços na busca pela organização do processo de implantação de seus produtos, visando o aumento da produtividade e da qualidade dos trabalhos realizados por quem os implanta (FREIRE, 2000).

Esta organização implica na padronização dos procedimentos, ferramentas, documentos e técnicas utilizadas no processo de implantação e na administração do uso do software (FREIRE, 2000).

A abordagem aqui defendida é decorrente da preocupação com os problemas ocorridos durante o período de implantação de um software, e também com os problemas que surgem após sua implantação. Estes problemas são provenientes de falhas, causadas principalmente por execução de trabalhos não padronizados e mal controlados. Em decorrência da evolução da informática, tais problemas tornam-se ainda mais críticos quando a realidade atual mostra que os usuários de software estão cada vez mais exigentes e conhecedores das circunstâncias que envolvem uma implantação. Por causa dos reflexos causados àqueles que estão ligados ao ciclo de engenharia de um software, acentua-se a necessidade de uma metodologia para que se possa elaborar o planejamento da fase de implantação, e isto demanda organização para desenvolver e implementar esse planejamento. Contudo, para que se possa eficientemente gerenciar este processo é necessário que seja estabelecida alguma forma de mensurá-lo, pois, como afirma Pressman (1995), “não se pode gerenciar o que não se pode medir”.

Deste modo, pretende-se criar uma unidade para mensurar uma estimativa de projeto de implantação de software genérico, similar ao *Function Point Analysis* (FPA) ou Análise por ponto de Função, que é utilizado para a medição por pontos de função de um projeto de desenvolvimento de software. A unidade criada será chamada de ponto de implantação e pode ser adotada por empresas de software para auxiliar no processo de implantação de seus sistemas. Sua forma de cálculo será inspirada no FPA, pois se pretende que a partir de

informações de entrada seja possível determinar a quantidade de pontos de implantação de um sistema.

Para colocar em prática esta métrica, cada ponto de implantação deve ser convertido para a quantidade relativa de horas ou valor após um estudo e comparação em cada empresa ou software a adotar este padrão.

O *Project Management Body of Knowledge* (PMBOK) traduzido como universo ou corpo de conhecimento em gerência de projetos, é uma metodologia desenvolvida pelo *Project Management Institute* que é um órgão certificador situado na Filadélfia, Estados Unidos, sendo uma contribuição muito útil para o desenvolvimento da Gerência de Projetos. Deste modo, pretende-se utilizar o PMBOK para a elaboração de um *check-list* e auxiliar na definição das fases do projeto regido pela métrica que será criada.

Para demonstrar a métrica criada com a aplicação dos conceitos de FPA e PMBOK, será desenvolvido um software que deve receber as informações dos clientes que pretendem utilizar um determinado aplicativo e realizar a estimativa do custo e da quantidade de pontos de implantação para o mesmo. A primeira validação do software será realizada em uma implantação do módulo Ronda de controle de ponto e acesso do sistema de Recursos Humanos Vetorh da Senior Sistemas Ltda (SENIOR SISTEMAS, 2003).

1.1 OBJETIVOS

O objetivo deste trabalho é criar uma métrica para estimar a complexidade da implantação de um software de recursos humanos, além de elaborar um software que faça esta estimativa.

Os objetivos específicos do trabalho são:

- a) elaborar um *check-list* e questionário de informações e necessidades do cliente para definir as fases de um projeto;
- b) elaborar um relatório contendo o detalhamento das fases de um projeto e as unidades de pontos de implantação previstos para cada fase;
- c) desenvolver um software para realizar a estimativa de implantação de um Software de Recursos Humanos.

1.2 ESTRUTURA

O presente trabalho está estruturado em seis capítulos, sendo que no capítulo 1 são apresentados a introdução, os objetivos e a estrutura do trabalho.

No capítulo 2 são abordados temas sobre FPA e conseqüentemente sobre Engenharia de Software.

O capítulo 3 aborda a fase de implantação de software com ênfase para o software de Recursos Humanos que é o foco deste trabalho.

O capítulo 4 apresenta o PMBOK e os motivos que levaram a utilização desta técnica.

No capítulo 5 será apresentada a métrica criada, com os objetivos, fases e planejamentos, bem como as explicações de como foi possível chegar a ela.

Finalmente, o capítulo 6 apresenta as conclusões e sugestões para aprimoramento do trabalho.

2 ENGENHARIA DE SOFTWARE E FPA

Este capítulo tem por objetivo uma breve definição sobre Engenharia de Software, e mostrar como é realizado o cálculo de pontos por função usando FPA. O processo de contagem de Pontos por Função deve ser simples, para minimizar o trabalho adicional do processo de mensuração, e conciso para permitir consistência ao longo do tempo dos projetos e entre os usuários da técnica.

2.1 ENGENHARIA DE SOFTWARE

A crise no desenvolvimento de software é praticamente uma constante, pois o mercado desenvolvedor está sempre buscando algo novo. Porém, e hoje mais do que nunca, é importante desenvolver e implantar software com qualidade, para evitar futuros incômodos e correções e evidentemente ter um bom mercado consumidor. Os primeiros passos a serem dados nesse sentido são reconhecer os problemas, suas causas, e demarcar ou provar que todas as técnicas que se tem à disposição para desenvolver software de qualidade, não são apenas mitos do software, mas deveriam ser o padrão no desenvolvimento do mesmo. (FREIRE, 2000)

Sommerville (1992), em *Software Engineering*, define a engenharia de software como uma disciplina da engenharia, preocupada com os problemas práticos do desenvolvimento dos grandes sistemas de software. Não é somente programação nem ciência da computação. Engenheiros de software devem ser profissionais que utilizam a teoria de outras disciplinas eficazmente na solução de problemas difíceis.

A Engenharia de Software propõe uma disciplina no desenvolvimento do software. Ela utiliza-se de um conjunto de técnicas e ferramentas que proporcionam o desenvolvimento de um software confiável e que funcione eficientemente em máquinas reais. Tais técnicas da Engenharia de Software abrangem todas as fases do desenvolvimento do software.

Os métodos da Engenharia de Software proporcionam os detalhes de como fazer um software. São um amplo conjunto de tarefas que incluem, planejamento e estimativa do projeto, análise e requisitos de software e de sistema, projeto da estrutura de dados,

arquitetura de programa e algoritmo de processamento, codificação de teste e manutenção. Todos esses métodos, levarão à qualidade do software.

Entre as técnicas conhecidas da Engenharia de Software, destaca-se o *Function Point Analysis* (FPA) ou análise por ponto de função que serviu de inspiração para o presente trabalho.

2.2 FPA

De acordo com Braga (1996), FPA é uma técnica de dimensionamento de projetos de software, que considera como unidade de medida os aspectos externos do software, requisitados e visíveis ao usuário.

A FPA é utilizada para estimar o nível de produtividade da equipe, o esforço de desenvolvimento de software, o custo de software, a taxa de produção e de manutenção de software.

2.2.1 HISTÓRICO

No início da década de 70, pesquisadores do Serviço de Processamento de Dados da IBM, a pedido do grupo de usuários, começaram a analisar centenas de programas para isolar as variáveis críticas, que determinam a produtividade da programação (INTERNATIONAL FUNCTION POINT USER GROUP, 1991).

Descobriram que poderiam basear a avaliação de um sistema, medindo-se o valor das funções executadas pelos programas, em vez de utilizar como base o volume ou a complexidade do código dos programas. Em 1979 Allan J. Albrecht (IBM White Plains), prosseguindo essas pesquisas, introduziu uma técnica de avaliação conhecida como FPA.

Em 1986, foi formado o Grupo Internacional de Usuários de FPA ou *International Function Point Users Group* (IFPUG) destinado a divulgar informações e novas implementações da técnica a todos os seus associados.

2.2.2 OBJETIVOS

FPA dimensiona software quantificando a funcionalidade que ele proporciona aos usuários (Azevedo, 1979).

Segundo Azevedo (1997), os objetivos da análise da técnica são: medir o que foi requisitado e recebido pelo usuário, medir independente da tecnologia utilizada para a implementação, prover uma métrica de medição para apoiar a análise de produtividade e qualidade, prover uma forma de estimar o tamanho do software e prover um fator de normalização para comparação de software.

2.2.3 TIPOS DE CÁLCULO DA TÉCNICA FPA

A técnica pode ser aplicada tanto no dimensionamento de projetos de aplicações já implantadas quanto no dimensionamento de projetos de desenvolvimento ou manutenção de aplicações.

2.2.3.1 Dimensionamento de um projeto de desenvolvimento

Este cálculo é utilizado para dimensionar em Pontos de Função o tamanho de um projeto de desenvolvimento de uma nova aplicação. O objetivo deste cálculo é quantificar as funções solicitadas e entregues ao usuário pela nova aplicação, incluindo neste caso, as funções referentes ao processo de conversão de dados. Este valor, menos os Pontos por Função associados às atividades de conversão de dados, torna-se o tamanho da aplicação, após sua implantação.

2.2.3.2 Dimensionamento de um projeto de manutenção

Este cálculo é utilizado para dimensionar o tamanho de um projeto de manutenção em uma aplicação já existente. O objetivo deste cálculo é medir todas as modificações de funções do usuário, incluindo, também, as funções providas pelo processo de conversão de dados. Ao final de um projeto de manutenção é necessário recalcular o tamanho da aplicação, para refletir as mudanças nas funções da aplicação.

2.2.3.3 Dimensionamento de uma aplicação

Este cálculo é utilizado para dimensionar o tamanho real de uma aplicação em Pontos por Função. O valor encontrado representa a funcionalidade da aplicação do ponto de vista do usuário. Esse valor pode diferir da dimensão do projeto de desenvolvimento desta aplicação, uma vez que não inclui as funções do processo de conversão de dados.

2.2.4 VISÃO GERAL DE UMA APLICAÇÃO

Segundo Longstreet (2001?) “Uma aplicação, vista sob a ótica do usuário, é um conjunto de funções ou atividades do negócio que o beneficiam na realização de suas tarefas”.

Essas funções podem ser divididas em 5 agrupamentos:

- a) Arquivo Lógico Interno: grupo lógico de dados do ponto de vista do usuário cuja manutenção é feita internamente pela aplicação.
- b) Arquivo de Interface Externa: grupo lógico de dados utilizados na aplicação cuja manutenção pertence à outra aplicação.
- c) Entrada Externa: transações vindas diretamente do usuário que referenciam arquivos internos.
- d) Saída Externa: são dados extraídos da aplicação, tais como relatórios e mensagens do terminal de vídeo.
- e) Consulta Externa: combinação de uma entrada e uma saída de dados, isto é, uma requisição de dados que gera uma aquisição e exibição imediata de dados.

2.2.5 RESUMO DO CÁLCULO DE PONTOS POR FUNÇÃO

Segundo o IFPUG (1991), determina-se os Pontos por Função de uma aplicação em três etapas de avaliação. A primeira resulta na contagem de pontos de funções não-ajustados, que refletem as funções específicas e mensuráveis do negócio, provida ao usuário pela aplicação. A segunda etapa da avaliação gera o fator de ajuste, que representa a funcionalidade geral provida ao usuário pela aplicação. A terceira etapa resulta na contagem de Pontos por Função ajustados, que reflete o fator de ajuste aplicado ao resultado apurado na primeira etapa.

Para fazer o cálculo de pontos de função não-ajustados, uma função específica do usuário em uma aplicação é avaliada em termos do que é fornecido pela aplicação e não como

é fornecido. Somente componentes solicitados pelo usuário e visíveis são contados. As funções específicas da aplicação vistas pelo usuário devem ser identificadas e agrupadas por tipo de função, conforme dito anteriormente. Cada tipo de função deve ser classificado de acordo com sua complexidade funcional relativa, como: simples, média ou complexa. O valor dos pontos varia de 3 a 15, dependendo do seu tipo e grau de complexidade.

O cálculo do fator de ajuste é realizado a partir de 14 características gerais dos sistemas, que permitem uma avaliação geral da funcionalidade da aplicação. As características gerais de um sistema são: comunicação de dados; processamento distribuído; performance; utilização de equipamento; volume de transações; entrada de dados "on-line"; processamento complexo; reutilização de código; facilidade de implantação; facilidade operacional; múltiplos locais; facilidade de mudanças.

Atribui-se um peso de 0 a 5 para cada característica, de acordo com o seu nível de influência na aplicação: 0 - Nenhuma influência; 1 - Influência mínima; 2 - Influência moderada; 3 - Influência média; 4 - influência significativa; 5 - Grande influência.

Finalmente o total de Pontos por Função da aplicação (pontos ajustados) pode ser encontrado multiplicando-se o fator de ajuste pela quantidade de Pontos por Função não ajustados.

3 IMPLANTAÇÃO DE SOFTWARE

Destaca-se neste capítulo a importância da fase de implantação dos softwares, citando circunstâncias que envolvem esta fase e fatores que direcionam para o desenvolvimento da métrica proposta.

3.1 A IMPORTÂNCIA DA FASE DE IMPLANTAÇÃO DE UM SOFTWARE

Como dito na introdução deste trabalho, sua intenção é a de apresentar uma métrica para a fase de implantação de um software (especificamente de RH). Desta forma, cabe apresentar alguns fatores que justificam esta importância, como se segue nos itens abaixo. Ou seja, se as circunstâncias não forem bem avaliadas e trabalhadas, o processo de implantação não será realizado corretamente fazendo com que os usuários não utilizem de maneira coerente o software, levando à ineficiência e prejudicando todo o seu ciclo de vida. Por isto a fase de implantação é importante.

3.1.1 A NOVA REALIDADE DOS CLIENTES E USUÁRIOS DE SOFTWARE

No estágio atual dos clientes e usuários de software, pode-se destacar o crescimento da exigência em relação à qualidade do produto final, em virtude do aumento da capacidade de análise destes clientes e usuários (FREIRE, 2000).

Além do fator humano e seu conhecimento ser algo muito relevante durante o processo de implantação, é comum encontrar-se especialistas cada vez mais capacitados a participar de forma ativa do processo de implantação de um software, nas diversas áreas afetadas por um projeto. A base do conhecimento da organização é fator preponderante na implantação de um software (FREIRE, 2000).

3.1.2 AS CIRCUNSTÂNCIAS QUE ENVOLVEM UMA IMPLANTAÇÃO DE SOFTWARE

Algumas das principais circunstâncias que envolvem um processo de implantação de software estão ligadas aos aspectos de: conhecimento do próprio negócio e suas

necessidades, conhecimento do mercado de softwares e seus produtos, acesso à informação, etc. Segundo Freire (2000), pode-se classificar estas circunstâncias em quatro características distintas:

- a) conhecimento da Organização, gerenciamento do conhecimento e busca de informações;
- b) análise do conhecimento adquirido e desenvolvimento do projeto;
- c) implementação do projeto;
- d) início da produção e acompanhamento dos resultados.

3.1.3 ESFORÇOS NA DIREÇÃO DA MELHORIA DA QUALIDADE DA IMPLANTAÇÃO

Para se ter qualidade no software, é preciso aplicar qualidade durante todo o trabalho antes do uso do software (PRESSMAN, 1995).

Algumas questões envolvem a qualidade do software, como por exemplo:

- a) O que é “Garantia da Qualidade de um Software?” A garantia de qualidade de software é o mapeamento dos preceitos administrativos e disciplinas de projeto de garantia de qualidade até o espaço administrativo tecnológico aplicável da engenharia de software (PRESSMAN, 1995).
- b) Quais são as maiores falhas de um software? As falhas de projeto e implantação (FREIRE, 2000).
- c) Quais os 5 problemas mais comuns em um desenvolvimento de software? Pobreza nos requisitos básicos do software, cronograma de implantação irreal, testes inadequados, “remendos” e falta de comunicação (FREIRE, 2000).
- d) Quais as soluções para os 5 problemas mais comuns? Concepção sólida dos requisitos, aproximar-se da realidade para elaboração do cronograma da implantação, realizar testes adequados, seguir o escopo inicial dos requisitos e manter-se alinhado a ele e priorizar a troca de informações (FREIRE, 2000).

Uma métrica desenvolvida especialmente para a implantação pode auxiliar estes esforços para a melhoria da qualidade.

Atualmente, pode-se encontrar inúmeras organizações e empresas que desenvolvem atividades ligadas à qualidade dos softwares, pode-se citar:

- a) Alliance for International Software Development (www.aisd.com)
- b) British Computer Society (www.bcs.org.uk)
- c) Computer Science Laboratory - SRI International (www.csl.sri.com)
- d) Data & Analysis Center for Software (www.dacs.dtic.mil)
- e) Defense Information Systems Agency (www.disa.mil/disahomejs.html)
- f) Fraunhofer IESE (www.iese.fhg.de)
- g) IEEE Computer Society (www.computer.org)
- h) Int'l Organization for Standards (www.iso.ch/welcome.html)
- i) Laboratory for Computer Science - MIT (www.lcs.mit.edu)
- j) Software Productivity Consortium (www.software.org)
- k) Ukranian Association of Software Developers (American Society for Quality (www.uasd.org.ua))

Não foi encontrada nenhuma referência que tratasse, especificamente, o período de implantação de uma maneira destacada dentro do processo de engenharia de software, assim como se pretende tratar neste trabalho.

A exceção a isto foi o caso de algumas empresas ligadas à implantação de softwares corporativos (Microsiga, Oracle, SAP, Senior Sistemas), que têm suas metodologias próprias para estimar a fase de implantação de seus softwares. Desta forma, justifica-se a criação de uma metodologia para auxiliar o projeto de implantação de software.

3.1.4 O PROBLEMA DO CONTROLE

Não se pode controlar o que não se pode medir (PRESSMAN, 1995). Com base neste princípio, a metodologia para implantação de um software teve originada sua principal característica.

O problema do controle é um assunto delicado. Nenhum gerente de projeto de software deixará de reagir à insinuação de que seu projeto está fora de controle. Embora os gerentes possam ter dificuldades em admitir a falta de controle de seus projetos, vale a pena a apreciação de alguns fatos. Segundo DeMarco, (1989) 15% de todos os projetos de software

nunca proporcionam nada, isto é, são completamente incapazes de atingir as metas propostas e reprocessamentos de 100 a 200% são comuns nos projetos de software.

Muitos projetos de software fracassam devido à falta de um bom gerenciamento. Há casos onde isto é superado com a inclusão de alguns fatores na condução do projeto, por parte de seu gerente:

- a) forte motivação dos membros da equipe do projeto;
- b) compreensão clara dos problemas;
- c) conhecimento adequado das tecnologias relevantes;
- d) capacidade evidente na esfera política da organização.

Um projeto é controlado no sentido de que é administrado para garantir o mínimo de surpresas durante seu desenvolvimento. Um projeto bem controlado não é necessariamente aquele que faz a melhor ou maior parte do trabalho, mas aquele que melhor convive com as previsões iniciais.

Manter o controle significa garantir que os resultados sejam iguais às expectativas. Isto requer dois itens que devem ser utilizados sempre em conjunto:

- a) deve-se gerenciar o projeto de forma que o desempenho seja igual ou superior a algum padrão razoável e aceito;
- b) deve-se garantir que as expectativas não excedam o que é possível ao desempenho de um projeto, dentro de tal padrão.

Qualquer aspecto do projeto que necessite de controle necessita de medição. O controle é impossível sem um *feedback*. Aplica-se aqui, dois corolários de DeMARCO (1989):

- a) a extensão do controle depende da precisão da medição;
- b) qualquer coisa que não seja medida, está fora de controle.

Com um controle adequado do projeto, especialmente na fase de implantação, ter-se-á alguns benefícios como:

- a) menor probabilidade de correções, com indicações precoces de fracassos;
- b) melhores estimativas originais;
- c) melhor gerenciamento destas expectativas;

- d) melhor capacidade de aperfeiçoamento do software;
- e) melhores relações com os clientes do projeto.

Para que estes fatores sejam reforçados, deve-se lembrar da importância da documentação de um projeto. Dentro da filosofia da Qualidade Total, as atividades relacionadas com a geração e controle de documentos, tais como Normas e Registros, têm relevância fundamental, pois são referências para a busca sistêmica da qualidade, especialmente por sua inter-relação com os aspectos humanos da organização (AZAMBUJA, 1996).

4 PMBOK

O PMBOK - *Project Management Body of Knowledge* que pode ser traduzido como "Corpo de conhecimento em gerência de projetos", é um guia onde se descreve a somatória de conhecimento e as melhores práticas dentro da profissão de gerência de projetos. Todo o conhecimento reunido neste guia é comprovado e não se restringe somente a práticas tradicionais, mas também às inovadoras e avançadas. Ele é um material genérico que serve para todas as áreas de conhecimento, ou seja, tanto para construção de edifício ou processo de fabricação industrial como para a produção de software. Um outro objetivo do PMBOK é a padronização de termos utilizados em gerência de projetos (PROJECT MANAGER INSTITUTE, 1996).

4.1 O QUE É GERÊNCIA DE PROJETOS?

De acordo com Oliveira (2002?), gerência de projeto é a aplicação de conhecimento, habilidades, ferramentas e técnicas em atividades do projeto, a fim de satisfazer ou exceder as necessidades e expectativas dos *stakeholders* (interessados e envolvidos). Satisfazer ou exceder as necessidades e expectativas dos *stakeholders* invariavelmente envolve equilibrar demandas concorrentes em relação a:

- a) escopo, prazo, custo e qualidade;
- b) *stakeholders* com necessidades e expectativas diferenciadas;
- c) requisitos identificados (necessidades) e requisitos não identificados (expectativas).

4.2 CONTEXTO DE GERÊNCIA DE PROJETO

Conforme Oliveira (2002?), a gerência de projeto e o projeto operam no ambiente onde está inserido o projeto. Sua equipe de gerenciamento tem que compreender o contexto – gerenciar diariamente as atividades é necessário para o seu sucesso, mas não é suficiente. É importante que sejam avaliados os seguintes tópicos para um amplo entendimento do contexto:

- a) fases e ciclo de vida;
- b) agentes;
- c) influências organizacionais;
- d) habilidades para gerência;

e) influência socioeconômica.

4.3 FASES E CICLO DE VIDA

As fases são a subdivisão do projeto em atividades que têm um objetivo a ser alcançado e geram produto(s) tangível(is) e verificável(is). A conclusão de uma fase normalmente é marcada com uma revisão das atividades e do desempenho do projeto. Esta determina se o mesmo continua em sua fase seguinte, além de detectar e corrigir eficazmente os erros surgidos (OLIVEIRA, 2002?).

Ciclo de vida são as atividades que serão executadas desde o início até o fim do projeto. Genericamente, o ciclo de vida define: qual trabalho técnico deve ser executado em cada fase, e quem deve ser envolvido. A depender da área do conhecimento, existem ciclos de vida de projeto já estudados e definidos. Por exemplo, para software: desenvolvimento rápido, iterativo, evolucionário, cascata, dentre outros (OLIVEIRA, 2002?).

4.4 PROCESSOS DE GERÊNCIA DE PROJETOS

O PMBOK é organizado em áreas de conhecimento e, por sua vez, cada área de conhecimento é descrita através de processos. Cada área de conhecimento se refere a um aspecto a ser considerado dentro da gerência de projetos. A sua não execução irá afetar negativamente o projeto, pois projeto é um esforço integrado.

A Figura 1 demonstra as áreas de conhecimento.

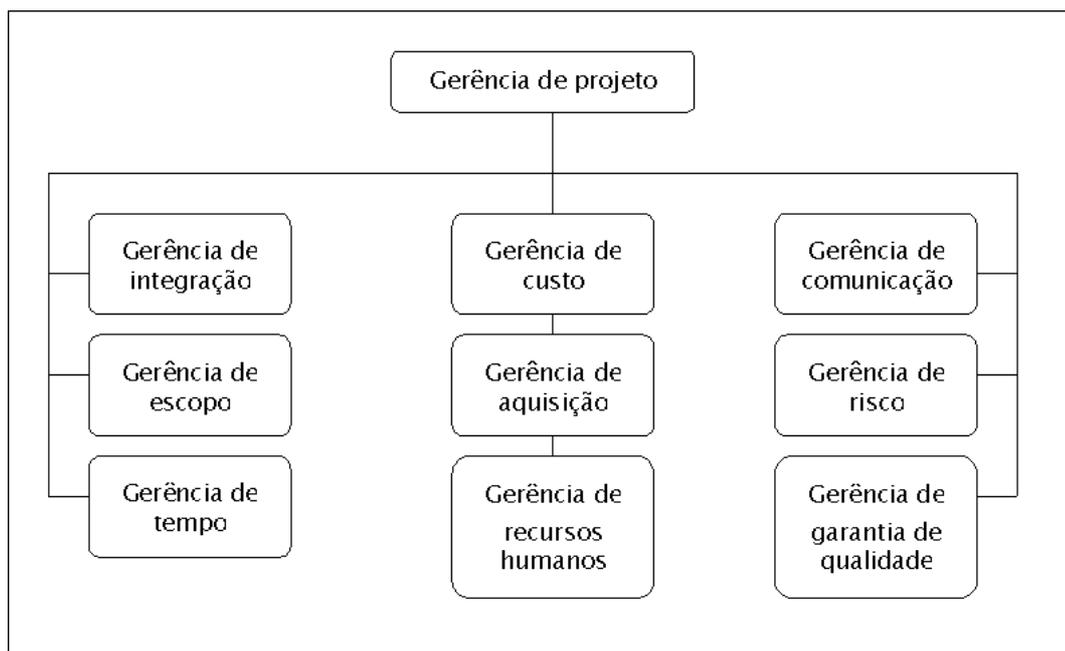


Figura 1 - Áreas de conhecimento na Gerência de Projetos

Este trabalho pretende utilizar apenas as áreas de conhecimento de Gerência de Tempo e de Custo.

4.4.1 GERÊNCIA DE TEMPO

Envolve os processos requeridos para garantir o término do projeto no tempo certo. Os principais processos são:

- a) definição das atividades: Identificação das atividades específicas que devem ser executadas para produzir as várias "entregas" do produto.
- b) seqüencialização das atividades: Identificação e documentação das dependências interativas.
- c) estimativas da duração da atividade: Estimativa do número de períodos de trabalho que serão necessários para completar as atividades individuais.
- d) desenvolvimento do cronograma: Análise das seqüências e durações das atividades e dos recursos necessários para se gerar o cronograma do projeto.
- e) controle do cronograma: Controle das alterações no cronograma do projeto.

4.4.2 GERÊNCIA DE CUSTO

Envolve os processos requeridos para garantir o término do projeto dentro do orçamento aprovado. Os principais processos são:

- a) planejamento dos recursos: Identificação de quais recursos (pessoas, equipamentos, materiais) e quais as quantidades de cada um deveriam ser usados para executar as atividades do projeto.
- b) estimativa de custo: Desenvolvimento de uma estimativa de custo dos recursos necessários para completar as atividades do projeto.
- c) orçamento de custo: Alocação das estimativas de custo ao item de trabalho.
- d) controle de custo: Controle das mudanças no orçamento do projeto.

5 MÉTRICA PARA IMPLANTAÇÃO

A métrica sugerida tem como base questionários e estimativas adotadas por uma empresa de Blumenau que desenvolve e implanta softwares e tem um Sistema de Gestão da Qualidade (SGQ) em conformidade com norma ISO 9001:2000. Parte desta experiência será utilizada na definição da métrica.

5.1 ETAPAS DA IMPLANTAÇÃO

Adaptando-se os processos de gerência de tempo e de custo do guia PMBOK (PMBOK, 1996), a fase de implantação de um software pode ser decomposta em 4 etapas:

- a) levantamento inicial: esta etapa se caracteriza por atividades de preparação do projeto de implantação, bem como o seu início no cliente, com reuniões para a formalização e preparação do ambiente para a execução dos trabalhos de implantação;
- b) instalação: nesta etapa são realizadas a instalação do sistema, a carga inicial dos dados e a parametrização do software com as particularidades do cliente. Normalmente esta é a etapa que requer o maior esforço da equipe de implantação;
- c) treinamento: Consiste em realizar os treinamentos direcionados às equipes correspondentes: recursos humanos, gerência, administração de TI. Em alguns momentos é necessária a criação de uma base de dados de treinamento separada da que será utilizada para a validação;
- d) validação: significa, o cliente aceitar como corretas as definições e parametrizações efetuadas, aprovando a liberação para execução do sistema ou módulo. Trata-se de uma ou mais reuniões envolvendo usuários chaves, para aprovar a liberação em produção e efetuar um plano para isto. Normalmente em softwares de Recursos Humanos como folha de pagamento e controle de apuração de ponto, efetua-se um teste em paralelo com o sistema atual da empresa, seja ele manual ou informatizado, e compara-se os resultados para obter a validação.

Estas etapas serão mais detalhadas no decorrer da descrição da métrica, neste capítulo.

5.2 QUESTIONÁRIO PARA CHECK-LIST

Para quantificar e medir a implantação de um software, será criada a medida Pontos de Implantação, ou simplesmente PI. Como todos os procedimentos adotados na empresa que foi pesquisada fazem referência a horas de implantação, para facilitar a elaboração da métrica, foi adotado que um ponto de implantação equivale a uma hora. Posteriormente, o valor em Pontos de Implantação pode ser convertido, ou ajustado para mais ou para menos que uma hora, conforme a realidade da empresa ou software que irá utilizar esta métrica.

Assim como existe um questionário para realizar a estimativa de Pontos de Função na técnica de FPA, foi criado um check-list para obter as informações necessárias para se medir a quantidade de PIs necessários para implantar um sistema.

O objetivo desse questionário é trazer informações sobre o cliente, que auxiliem nas estimativas de tempos de implantação dos sistemas e estimativas de etapas. Se o projeto abrange a implantação em várias empresas que tem operações independentes, cada empresa deve ser tratada com um questionário independente.

5.2.1 LEVANTAMENTO INICIAL

A primeira necessidade é estimar o tamanho da empresa, para isto, saber o número total de colaboradores é fundamental.

O Quadro 1 mostra as perguntas sobre a etapa de Levantamento Inicial.

Pergunta	Motivo	Forma de cálculo
a) Quantidade de colaboradores (funcionários, terceiros ou parceiros):	Estimar o tamanho da Organização / Empresa para medir a complexidade de atividades como carga de dados e definições de procedimento.	Quantidade de colaboradores dividida por 1000. Se resultado menor que 1, atribui-se 1 Ponto de Implantação.
b) Quantidade de empresas	Quanto maior o número de	(1 = 0 pontos) (2 a 5 = 3 pontos)

no cliente:	empresas, maior a quantidade de usuários e procedimentos para a utilização do Sistema.	(5 a 10 = 5 pontos) (mais de 10 = 8 pontos)
-------------	--	--

Quadro 1 – Perguntas sobre a etapa de Levantamento Inicial.

Um sistema de RH é bastante abrangente, tratando desde o pagamento dos colaboradores até mesmo da distribuição e validade dos equipamentos de segurança que cada um deve utilizar em seu trabalho, bem como os treinamentos necessários para exercer as funções. Pretende-se adotar a separação entre módulos do sistema de RH da Senior Sistemas (2002) para facilitar a estimativa:

- a) Folha de pagamento – Calcula a folha e distribui o valor do salário e benefícios dos colaboradores;
- b) Ponto e acesso – efetua o cálculo da apuração de ponto com base na escala de horários e passa as informações para a folha de pagamento. Também faz o controle do acesso dos colaboradores e visitante às dependências da empresa;
- c) Treinamento – gerencia as turmas de treinamento e as habilidades que cada colaborador deve possuir;
- d) Cargos e salários – organiza o plano de cargos e salários da empresa controlando o requisitos de cada cargo;
- e) Recrutamento e Seleção – realiza o acompanhamento das novas contratações com base nas necessidades das vagas e pré-requisitos para o cargo.

Deste modo mais uma pergunta pode ser incluída conforme o Quadro 2 para ser utilizada no software que vai realizar a estimativa, assim sabe-se que outras perguntas realizar.

Pergunta	Motivo	Forma de cálculo
a) Quais são os módulos que estão sendo implantados?	Habilitar demais perguntas conforme a necessidade.	Não existe cálculo para esta questão.

Quadro 2 – Questionamento sobre os módulos do Sistema de RH

5.2.2 INSTALAÇÃO

É importante saber como a equipe de TI do cliente pretende auxiliar a implantação do sistema. A forma de obtenção das informações pode agilizar ou onerar muito o prazo para a conclusão dos serviços. Se o consultor que vai fazer a instalação do sistema tiver que “aprender” como funciona o sistema antigo do cliente para extrair os dados para o sistema em implantação, com certeza deve-se prever um tempo extra para esta execução. As cinco perguntas no Quadro 3 tem relação direta com o exposto neste item e devem ser preenchidas para estimar esta etapa:

Pergunta	Motivo	Forma de cálculo
a) Pode importar dados?	Caso os dados não sejam importados, o cliente cadastra manualmente.	Se não, não acrescentar pontos, se sim acrescentar 15.
b) Disponibilizará informações em formato pré-estabelecido?	Caso os arquivos sejam disponibilizados no formato do cliente, o consultor terá que criar a interface para importar.	Se não, acrescentar 15 pontos, se sim, não se acrescenta pontos.
c) A carga será realizada pelo cliente?	Caso o cliente realize a importação, o consultor não será onerado no trabalho.	Se não, acrescentar 15 pontos, se sim, não se acrescenta pontos.
d) Tem DBA?	A falta de DBA leva o consultor a configurar o Banco de dados	Se não, acrescentar 10 pontos, se sim, zero.
e) Tem equipe TI?	O Consultor terá que configurar acesso à rede e permissões.	Se não, acrescentar 10 pontos, se sim, zero.

Quadro 3 – Perguntas sobre a etapa de Instalação

5.2.2.1 Módulo de Folha de Pagamento

Para o módulo de Folha de pagamento, há necessidade de questionamentos específicos sobre legislação e forma de atuação da empresa. Cada pergunta respondida positivamente, acrescenta pontos no total da estimativa. Para perguntas com mais de uma opção, cada opção é tratada como uma pergunta em separado, isto é, agrega valor para o montante final. O Quadro 4 demonstra as perguntas específicas para o módulo de Folha de Pagamento.

Pergunta	Motivo	Forma de cálculo
a) Efetua pagamento aos Terceiros (Prestadores de Serviços)?	Item que agrega tempo para desenvolvimento de regra.	1 ponto para resposta positiva.
b) Quais os tipos de contratos utilizados na Empresa? CLT, Estatutários, Diretores, Estagiários, Menor Aprendiz, Prazo Determinado.	Item que agrega tempo para desenvolvimento de regra.	1 ponto para resposta positiva.
c) Possui colaboradores com Desconto de Pensão Alimentícia?	Item que agrega tempo para desenvolvimento de regra.	1 ponto para resposta positiva.
d) Utiliza rotina de distribuição de Vale Transporte?	Item que agrega tempo para desenvolvimento de regra.	1 ponto para resposta positiva.
e) Efetua o pagamento de “Anuidades”?	Item que agrega tempo para desenvolvimento de regra.	1 ponto para resposta positiva.
f) Possui “Convênios” que são descontados em Folha de Pagamento?	Item que agrega tempo para desenvolvimento de regra.	1 ponto para resposta positiva.
g) Efetua lançamentos mensalmente ou fixa para	Item que agrega tempo para desenvolvimento de regra.	1 ponto para resposta positiva.

que ocorram todos os meses no cálculo da Folha de Pagamento?		
h) Tem informação que possa ser “Importada” via rotina disponível no sistema (através de arquivo texto entregues por Farmácias, Supermercados, Convênios Médicos, etc.), no movimento mensal?	Elaboração de interface para integração por importação.	5 pontos para resposta positiva.
i) Quais os tipos de processamentos realizados no mês? Mensal, Quinzenal, semanal, adiantamento Salarial, outros.	Item que agrega tempo para desenvolvimento de regra.	1 ponto para resposta positiva.
j) O Sistema contábil irá importar os dados provenientes da folha de pagamento (é necessária integração?).	Elaboração de interface para integração por exportação.	5 pontos para resposta positiva.

Quadro 4 – Perguntas sobre o Módulo Folha de Pagamento

5.2.2.2 Módulo de Controle de Ponto e Acesso

O módulo de Ponto e Acesso requer uma atenção especial para a quantidade de horários e escalas de trabalho. Mesmo que uma empresa tenha 10.000 colaboradores, ela pode implantar um sistema de ponto em menos tempo que uma empresa que possui apenas 1.000 colaboradores. Basta que a primeira utilize apenas um horário fixo como das 8:00 às 18:00 e a segunda adotes vários horários fixos e outros flexíveis (onde os colaboradores têm que cumprir uma quantidade determinada de horas por dia, independente da hora que entram e saem da empresa).

No Quadro 5 cada pergunta respondida positivamente, agrega um ponto no montante final. Para perguntas com valores, o comentário e motivo específico estarão após a pergunta:

Pergunta	Motivo	Forma de cálculo
a) Possui relógio de ponto eletrônico	Prever configuração para leitura dos arquivos texto dos relógios.	5 pontos, se respondida positivamente.
b) Quantos colaboradores marcam ponto?	Quanto mais colaboradores, mais tempo para prever todas as particularidades na regra de apuração de ponto.	Quantidade de Colaboradores / 100 = quantidade de pontos de implantação a ser somado
c) Responsabilidade pelo abono, acerto:	Se abono é pela Chefia, há necessidade de mais instalações e acompanhamentos.	Centralizado (RH) = 1 ponto Descentralizado (chefias) = 5 pontos.
d) Trabalha com escala de revezamento?	Item que agrega tempo para desenvolvimento da regra de apuração.	1 ponto para resposta positiva.
e) Executa atividades em horas extras?	Item que agrega tempo para desenvolvimento da regra de apuração.	1 ponto para resposta positiva.
f) Existe horário noturno?	Item que agrega tempo para desenvolvimento da regra de apuração.	1 ponto para resposta positiva.
g) Horário flexível?	Item que agrega tempo para desenvolvimento da regra de apuração.	1 ponto para resposta positiva.
h) Desconta Descanso Semanal Remunerado (DSR) em caso de faltas ou atrasos?	Item que agrega tempo para desenvolvimento da regra de apuração.	1 ponto para resposta positiva.

i) Quantos turnos ou escalas possui a empresa?	Várias escalas aumentam em muito a complexidade das regras de apuração.	Chega-se ao número de pontos de implantação multiplicando a quantidade de escalas pela quantidade de colaboradores que marcam ponto e dividindo por 100.
j) Possui ou está implantando Banco de horas?	Item que agrega tempo para desenvolvimento da regra de apuração.	1 ponto para resposta positiva.
k) Possui refeitório?	Item que agrega tempo para desenvolvimento da regra de apuração.	1 ponto para resposta positiva.
l) Pretende descontar refeição?	Item que agrega tempo para desenvolvimento da regra de apuração.	1 ponto para resposta positiva.
m) Quantidade de portarias:	Tempo para configuração das estações.	1 ponto por portaria.
n) Áreas de acesso restrito?	Item que agrega tempo para desenvolvimento da regra de apuração.	1 ponto para resposta positiva.
o) Controle de acesso por listas?	Item que agrega tempo para desenvolvimento da regra de apuração.	1 ponto para resposta positiva.

Quadro 5 – Perguntas sobre o módulo de controle de Ponto e Acesso.

Os demais módulos são relativamente mais simples, e pretende-se adotar a seguinte regra relacionando a quantidade de colaboradores com o tempo necessário para a implantação:

5.2.2.3 Módulo de Recrutamento e Seleção - RS

O que vai influenciar a implantação deste módulo é o volume de admissões que a empresa tem num mês.

O Quadro 6 mostra a pergunta necessária para estimar os PIs necessários a este módulo.

Pergunta	Motivo	Forma de cálculo
a) Quantidade de admissões mensais:	Quanto mais admissões mais rotinas o módulo deverá prever.	De 1 a 50 – 20 pontos De 51 a 100 – 40 pontos Mais que 100 – 80 pontos

Quadro 6 – Pergunta sobre o módulo RS

5.2.2.4 Módulo de Treinamento - TR

A quantidade de turmas de treinamentos é o mais importante para este módulo. No Quadro 7 há a pergunta, a justificativa e a forma de cálculo dos PIs para este módulo.

Pergunta	Motivo	Forma de cálculo
a) Quantidade de turmas de treinamento por mês:	Quanto mais turmas a empresa possuir, mais cadastramentos e acompanhamentos deverão ser previstos.	De 1 a 5 – 20 pontos De 6 a 10 – 40 pontos Mais que 10 – 80 pontos.

Quadro 7 – Pergunta sobre o módulo TR.

5.2.2.5 Módulo de Cargos e Salários - CS

O tempo de implantação do módulo de Cargos e Salários vai depender da quantidade de cargos que a Empresa possui. O Quadro 8 mostra a forma de estimar os PIs para este módulo.

Pergunta	Motivo	Forma de cálculo
a) Quantidade de cargos da empresa:	Quanto mais cargos a empresa tiver, maior será o empenho para cadastrar a	De 1 a 50 – 20 pontos De 51 a 100 – 40 pontos Mais de 100 – 80 pontos.

	política de cargos e salários no sistema.	
--	---	--

Quadro 8 – Pergunta sobre o módulo CS.

5.2.3 TREINAMENTO

Para a etapa de treinamento, deve-se levar em consideração a quantidade de pessoas a serem treinadas multiplicando por 8, que é a média de horas de treinamento para a utilização de um software de RH segundo levantamento realizado e dividindo o total por 10 (quantidade de pessoas a serem treinadas ao mesmo tempo). Assim pode-se saber quantos pontos de implantação serão necessários para a etapa de treinamento, conforme o Quadro 9.

Pergunta	Motivo	Forma de cálculo
a) Quantas pessoas devem receber treinamento?	8 horas = média de treinamento por pessoa. 10 = quantidade ideal de pessoas na mesma turma.	Quantidade multiplicado por 8 e dividido por 10.

Quadro 9 – Perguntas sobre treinamento.

5.2.4 VALIDAÇÃO E ACEITE

Aqui a estimativa é relativamente simples, pois a validação normalmente é realizada com uma operação assistida. Esta operação assistida deve ser prevista no momento do preenchimento do *check-list* e acrescida de 10 pontos de implantação para ajustes. Conforme o Quadro 10.

Pergunta	Motivo	Forma de cálculo
a) Realiza Operação Assistida?	Quantidade de horas solicitada pelo cliente.	Quantidade de horas estimadas mais 10 pontos para ajuste.

Quadro 10 – Pergunta sobre Validação e Aceite.

6 IMPLEMENTAÇÃO

6.1 ESPECIFICAÇÃO DO SISTEMA

A especificação do protótipo foi realizada utilizando a linguagem de modelagem *Unified Modeling Language* (UML). Segundo Furlan (1998), a UML é a padronização da linguagem de desenvolvimento orientado a objetos para visualização, especificação, construção e documentação de sistemas.

Para a modelagem foram utilizados os diagramas de casos de uso, diagramas de seqüência e diagrama de classes. Os mesmos foram feitos utilizando a ferramenta *Rational Rose C++ Demo 4.0.3* da empresa *Rational Software Corp.*

6.1.1 CASOS DE USO

Neste software existe um caso de uso primário. Ele é descrito a seguir:

- a) **CalcularPontosImplantação:** Responsável por calcular os pontos de implantação de cada etapa. O cálculo é realizado a cada valor inserido, alterado ou excluído.

A Figura 2 demonstra o caso de uso primário do software.

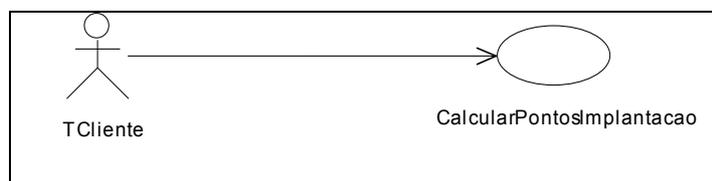


Figura 2 - Caso de uso do sistema

6.1.2 DIAGRAMA DE CLASSES

A Figura 3 demonstra o diagrama de classes do protótipo. Para todos os Atributos foram implementados métodos de leitura e atribuição através de propriedades do Delphi (*properties*).

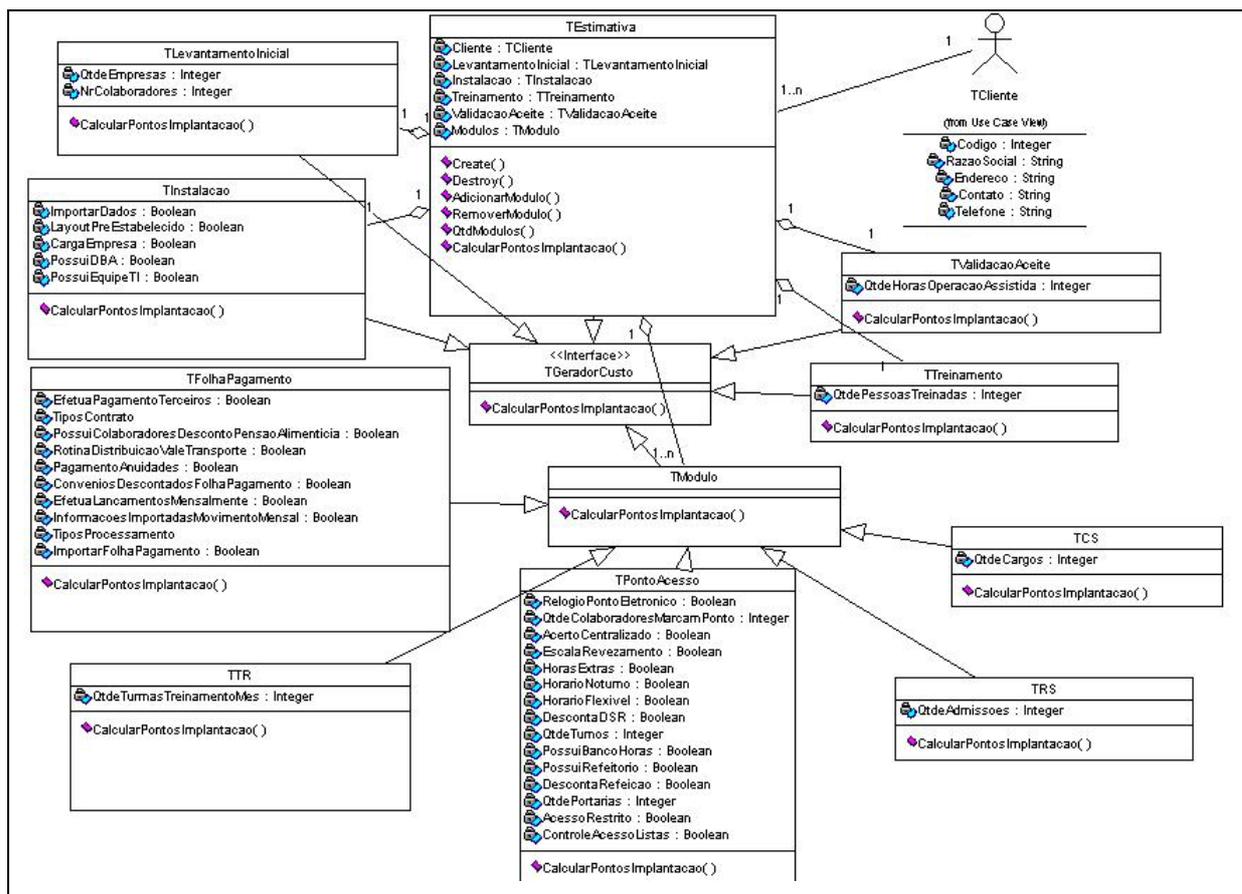


Figura 3 – Diagrama de classes do software

Pode-se verificar a existência das seguintes classes no protótipo:

- TCliente: mantém as informações de um cliente. Cada estimativa possui um cliente. Um cliente pode ter mais de uma estimativa.
- <<interface>>TGeradorCusto: outras classes implementam o método CalcularPontosImplantacao da interface TGeradorCusto que faz o cálculo do PI de uma estimativa (método CalcularPontosImplantacao). Cada estimativa sempre possui um TGeradorCusto.
- TLevantamentoInicial: possui as informações referentes ao levantamento inicial da estimativa. Também Implementa o método CalcularPontosImplantacao.
- TInstalacao: possui as informações referentes à instalação da estimativa.
- TTreinamento: possui as informações referentes a etapa de treinamento da estimativa.
- TValidacaoAceite: trata a etapa de validação e aceite da estimativa.
- TModulo: implementa TGeradorCusto. Classe abstrata criada apenas para

"agrupar" os módulos da estimativa. Através da propriedade herança da orientação a objeto. Porque no futuro talvez todos os módulos tenham algum atributo ou método em comum, por isso é recomendado que eles derivem da mesma classe.

- h) TFolhaPagamento: deriva de TModulo. Cada estimativa pode possuir ou não um TFolhaPagamento. Recebe as informações referentes ao módulo Folha de pagamento a ser implantado na estimativa. Também possui o método CalcularPontosImplantacao que retorna os PI do módulo Folha de pagamento .
- i) TPontoAcesso: idem TFolhaPagamento, só que para o módulo Ponto e Acesso.
- j) TRS: idem TFolhaPagamento, só que para o módulo Recrutamento e Seleção.
- k) TTR: idem TFolhaPagamento, só que para o módulo Treinamento.
- l) TCS: idem TFolhaPagamento, só que para o módulo Cargos e Salários.
- m) TEstimativa = representa uma estimativa. Sempre que o aplicativo cria uma nova estimativa, um objeto da classe TEstimativa é criado. Esse objeto possui um objeto de TLevantamentoInicial (para armazenar as informações do levantamento inicial da estimativa), um objeto de TInstalacao (para armazenar as informações da instalação da estimativa), um objeto de TTreinamento (para armazenar as informações do treinamento da estimativa), um objeto de TValidacaoAceite (para armazenar as informações de validação e aceite da estimativa) e um *array* de objetos de TModulo, porque a estimativa pode ter 1 ou vários módulos a serem instalados.

6.1.3 DIAGRAMAS DE SEQUÊNCIA

Os diagramas de seqüência representam a seqüência em que as ações ocorrem dentro do sistema. Eles demonstram como é feita a troca de mensagens entre as classes. Existe um diagrama de seqüência para o caso de uso primário do software.

A Figura 4 mostra o Diagrama de Seqüência do caso de uso CalcularPontosImplantação.

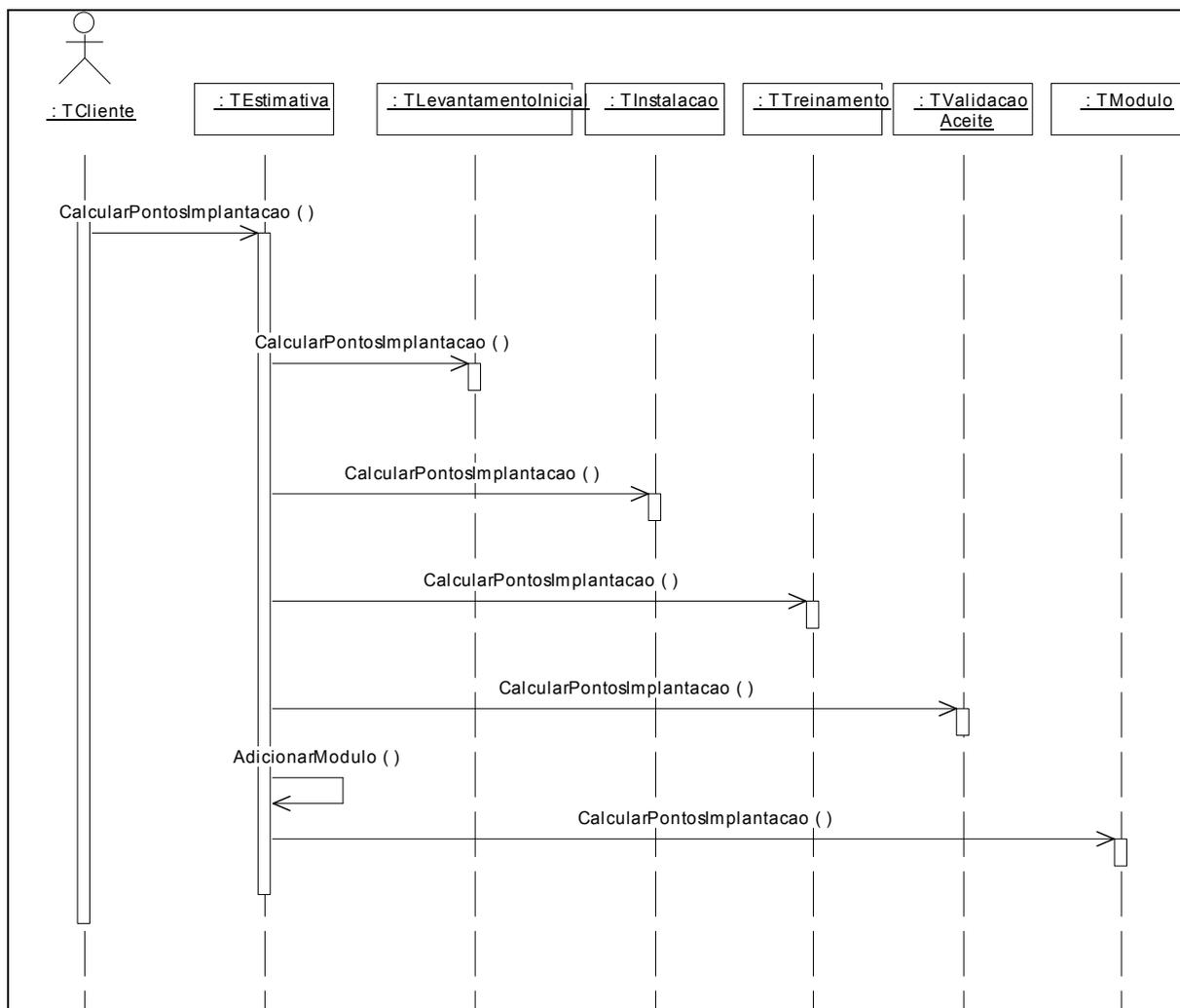


Figura 4 – Diagrama de seqüência CalculadorPontosImplantação

6.2 IMPLEMENTAÇÃO

Para implementação do sistema foi utilizada a ferramenta Borland Delphi 6.0. O Delphi é um ambiente integrado de desenvolvimento que permite a edição, compilação e execução de programas. A linguagem de programação utilizada pelo Delphi é o Object Pascal.

A Figura 5 mostra a tela principal do sistema. Nela o usuário pode visualizar um resumo das estimativas cadastradas, além de emitir relatórios. É nesta tela também que se tem acesso às telas de cadastramento de cliente e estimativas.



Cliente	Levantamento Inicial	Instalação	Treinamento	Validação e Aceite	Pontos de Implantação (Total)
Empresas ABC LTDA	4	50	4	26	149
CG Data Center	1	20	4	18	65
Comércio e Represent	4	5	8	54	395

Buttons: Adicionar, Alterar, Remover, Relatório

Figura 5 – Tela principal do sistema

A Figura 6 mostra a tela de Cadastramento de Clientes



Cliente

Razão Social:
Empresas ABC LTDA

Endereço:
Rua Central, 434

Contato:
Sr. João Almeida

Telefone:
555-5500

Sair

Figura 6 – Tela de Cadastro de Clientes

A Figura 7 mostra a tela de Cadastramento de Estimativas

Cliente:

Quantidade de empresas: Número de colaboradores:

Número de pessoas treinadas:

Quantidade de horas da operação assistida:

Importar dados Possui DBA Possui equipe de TI

Disponibilizará informações em lay-out pré-estabelecido

Carga será realizada pelo cliente

Módulos a serem implantados:

Módulo	Pontos de Implantação
Ponto Acesso	46
Folha de Pagamento	19

Pontos de Implantação:

Levantamento Inicial: 4
 Instalação: 50
 Treinamento: 4
 Validação e Aceite: 26
 Total: 149

OK Cancelar

Figura 7 – Tela para cadastrar estimativas

Para adicionar módulos, deve-se clicar no botão Adicionar para exibir a tela correspondente, conforme a figura 8. Ao inserir um módulo, a tela referente ao mesmo é exibida permitindo o preenchimento do check-list conforme o exemplo da figura 9 que contém a tela do módulo Ponto e Acesso.

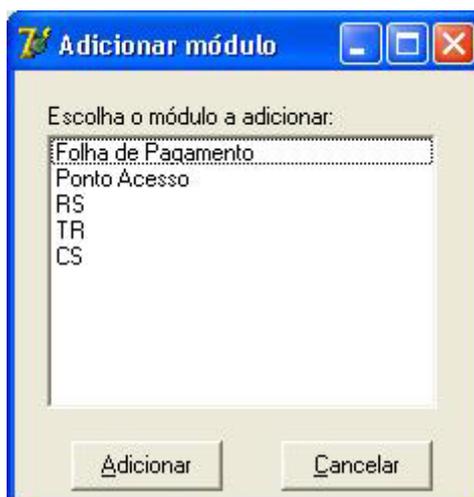


Figura 8 – Tela para adicionar módulos

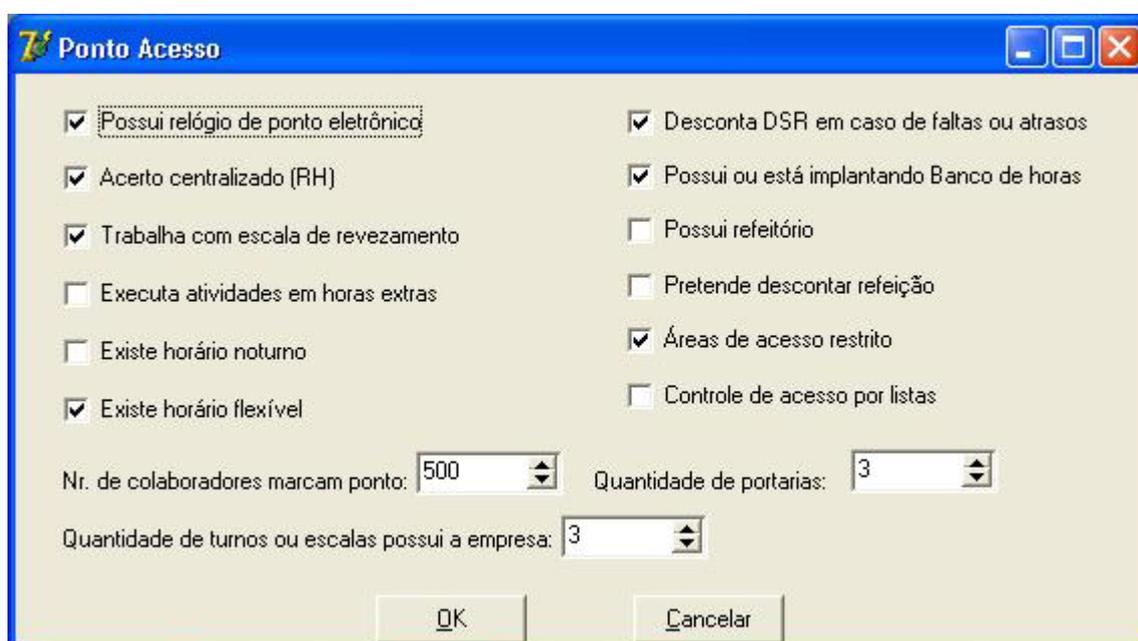


Figura 9 – Tela do módulo Ponto e Acesso

Após o cadastramento da estimativa e dos seus módulos, pode-se emitir um relatório que traz o resumo dos PIs por etapa da implantação e ainda prevê uma estimativa de custo e quantidade de dias para o projeto com base em valores cadastrados nas definições do programa. A figura 10 mostra a tela com o relatório.

perguntas sendo respondidas, o valor dos pontos de implantação pode diminuir ou aumentar, mas no final, sempre se chega ao valor especificado na métrica.

A métrica foi comparada com valores reais de uma implantação de software. Foram inseridas informações no questionário conforme um projeto de implantação e o resultado obtido foi comparado com o realizado, conforme o quadro abaixo:

Etapa	Estimado (PI)	Realizado (H)
Levantamento Inicial	1	1
Instalação	20	18
Validação	4	6
Treinamento	18	16
Ponto e Acesso	22	22
Total	65	63

Quadro 11 – Validação do Software.

No quadro 11 pode-se ver que a diferença entre o estimado e o realizado está dentro de uma margem de 5%. Margem esta, admissível para um projeto de implantação de software.

7 CONCLUSÕES

O tema abordado tem interesse restrito a um determinado segmento do mercado de informática. Além das empresas de consultoria de softwares de gestão, poucas organizações demonstram alocar recursos para tratar o período de implantação do software, como objeto de pesquisa, visando obter vantagens. Analisou-se, também, a necessidade de se utilizar conceitos de diversos segmentos para se desenvolver uma abordagem metodológica para a implantação de softwares. Isto possibilitou um resultado que, provavelmente, dará condições de ampliar o campo de aplicação da métrica proposta.

Colocar em prática a métrica faz com que os profissionais envolvidos na implantação tenham um padrão rigoroso a ser seguido, evitando desvios nas diretrizes do projeto, fazendo com que não se perca tempo e conseqüentemente dinheiro com ações indevidas e estouro de cronogramas.

A métrica foi desenvolvida para auxiliar as empresas ou profissionais que desejem aumentar a confiabilidade nos resultados e diminuir os riscos de prejuízos em uma implantação de software.

Com esses argumentos, acredita-se ter apresentado suficientes fundamentos e subsídios para a viabilização de estudos que aprimorem uma métrica específica para a implantação de softwares, embora conscientes da suas limitações e, portanto, da necessidade de novas pesquisas sobre este tema, ainda recente e carente de conclusões.

As técnicas de FPA e PMBOK e as ferramentas Borland Delphi 6.0 e Rational Rose Demo 4.0 utilizadas na elaboração do trabalho, cumpriram o que se esperava, sendo de grande utilidade para o desenvolvimento da métrica e implementação do software.

7.1 SUGESTÕES

Durante o desenvolvimento deste trabalho foi constatado que praticamente não há literatura sobre este tema. Os documentos estão restritos às empresas que têm negócios ligados à implantação de softwares. Mesmo assim, os esforços para aprimoramento de metodologias voltadas ao tema exposto, não são direcionados na mesma proporção da importância do assunto.

O aprimoramento da métrica poderá vir da diversificação de sua aplicação em projetos que não sejam exclusivamente ligados a softwares de recursos humanos. Seguem abaixo algumas sugestões de assuntos que darão seqüência a este trabalho ou algumas sugestões de futuros trabalhos relacionados ao assunto aqui abordado:

- a) o refinamento da métrica, fortalecendo mais seus embasamentos técnicos, dando uma abordagem mais aprofundada aos seus conceitos e prevendo com mais sensibilidade os custos para o projeto;
- b) elaboração de uma métrica similar para implantação de software corporativo de gestão empresarial;
- c) a criação de uma ferramenta baseada nos conceitos da métrica que auxilie no acompanhamento do projeto de implantação, criando um cronograma mais detalhado.

ANEXO I

Este anexo apresenta as interfaces das classes do software.

a) Interface das classes do software:

```

TCliente = class
  private
    FCodigo: Integer;
    FRazaoSocial: String;
    FEndereco: String;
    FContato: String;
    FTelefone: String;
  public
    property Codigo: Integer read FCodigo write FCodigo;
    property RazaoSocial: String read FRazaoSocial write FRazaoSocial;
    property Endereco: String read FEndereco write FEndereco;
    property Contato: String read FContato write FContato;
    property Telefone: String read FTelefone write FTelefone;
end;

TEstimativa = class;

TGeradorCusto = class
  public
    function CalcularPontosImplantacao( aEstimativa: TEstimativa ): Integer; virtual;
end;

TLevantamentoInicial = class(TGeradorCusto)
  private
    FQtdeEmpresas: Integer;
    FNrColaboradores: Integer;
  public
    property QtdeEmpresas: Integer read FQtdeEmpresas write FQtdeEmpresas;
    property NrColaboradores: Integer read FNrColaboradores write FNrColaboradores;

    function CalcularPontosImplantacao: Integer; reintroduce;
end;

TInstalacao = class(TGeradorCusto)
  private
    FImportarDados: Boolean;
    FLayoutPreEstabelecido: Boolean;
    FCargaEmpresa: Boolean;
    FPossuiDBA: Boolean;
    FPossuiEquipeTI: Boolean;
  public
    property ImportarDados: Boolean read FImportarDados write FImportarDados;
    property LayoutPreEstabelecido: Boolean read FLayoutPreEstabelecido write
FLayoutPreEstabelecido;
    property CargaEmpresa: Boolean read FCargaEmpresa write FCargaEmpresa;
    property PossuiDBA: Boolean read FPossuiDBA write FPossuiDBA;
    property PossuiEquipeTI: Boolean read FPossuiEquipeTI write FPossuiEquipeTI;

    function CalcularPontosImplantacao: Integer; reintroduce;
end;

TTreinamento = class(TGeradorCusto)
  private
    FQtdePessoasTreinadas: Integer;
  public
    property QtdePessoasTreinadas: Integer read FQtdePessoasTreinadas write
FQtdePessoasTreinadas;

    function CalcularPontosImplantacao: Integer; reintroduce;
end;

TValidacaoAceite = class(TGeradorCusto)
  private
    FQtdeHorasOperacaoAssistida: Integer;
  public

```

```

    property QtdeHorasOperacaoAssistida: Integer read FQtdeHorasOperacaoAssistida write
FQtdeHorasOperacaoAssistida;

    function CalcularPontosImplantacao: Integer; reintroduce;
end;

TModulo = class(TGeradorCusto)
private
public
    function CalcularPontosImplantacao: Integer; virtual; abstract;
end;

TFolhaPagamentoTipoContrato = (tcCLT, tcEstatutario, tcDiretor, tcEstagiario,
tcMenorAprendiz, tcPrazoDeterminado);

TSetFolhaPagamentoTipoContrato = set of TFolhaPagamentoTipoContrato;

TFolhaPagamentoTipoProcessamento = (tpMensal, tpQuinzenal, tpSemanal,
tpAdiantamentoSemanal);

TSetFolhaPagamentoTipoProcessamento = set of TFolhaPagamentoTipoProcessamento;

TFolhaPagamento = class(TModulo)
private
    EFetuaPagamentoTerceiros: Boolean;
    FTiposContrato: TSetFolhaPagamentoTipoContrato;
    FPossuiColaboradoresDescontoPensaoAlimenticia: Boolean;
    FRotinaDistribuicaoValeTransporte: Boolean;
    FPagamentoAnuidades: Boolean;
    FConveniosDescontadosFolhaPagamento: Boolean;
    EFetuaLancamentosMensalmente: Boolean;
    FInformacoesImportadasMovimentoMensal: Boolean;
    FTiposProcessamento: TSetFolhaPagamentoTipoProcessamento;
    FImportarFolhaPagamento: Boolean;
public
    property EfetuaPagamentoTerceiros: Boolean read
FEfetuaPagamentoTerceiros write EFetuaPagamentoTerceiros;
    property TiposContrato: TSetFolhaPagamentoTipoContrato read FTiposContrato write
FTiposContrato;
    property PossuiColaboradoresDescontoPensaoAlimenticia: Boolean read
FPossuiColaboradoresDescontoPensaoAlimenticia write
FPossuiColaboradoresDescontoPensaoAlimenticia;
    property RotinaDistribuicaoValeTransporte: Boolean read
FRotinaDistribuicaoValeTransporte write FRotinaDistribuicaoValeTransporte;
    property PagamentoAnuidades: Boolean read FPagamentoAnuidades
write FPagamentoAnuidades;
    property ConveniosDescontadosFolhaPagamento: Boolean read
FConveniosDescontadosFolhaPagamento write FConveniosDescontadosFolhaPagamento;
    property EfetuaLancamentosMensalmente: Boolean read
EEfetuaLancamentosMensalmente write EFetuaLancamentosMensalmente;
    property InformacoesImportadasMovimentoMensal: Boolean read
FInformacoesImportadasMovimentoMensal write FInformacoesImportadasMovimentoMensal;
    property TiposProcessamento: TSetFolhaPagamentoTipoProcessamento read FTiposProcessamento
write FTiposProcessamento;
    property ImportarFolhaPagamento: Boolean read
FImportarFolhaPagamento write FImportarFolhaPagamento;

    function CalcularPontosImplantacao: Integer; reintroduce;
end;

TPontoAcesso = class(TModulo)
private
    FRelogioPontoEletronico: Boolean;
    FQtdeColaboradoresMarcamPonto: Integer;
    FAcertoCentralizado: Boolean;
    FEscalaRevezamento: Boolean;
    FHorasExtras: Boolean;
    FHorarioNoturno: Boolean;
    FHorarioFlexivel: Boolean;
    FDescontaDSR: Boolean;
    FQtdeTurnos: Integer;
    FPossuiBancoHoras: Boolean;
    FPossuiRefeitório: Boolean;

```

```

    FDescontaRefeicao: Boolean;
    FQtdePortarias: Integer;
    F AcessoRestrito: Boolean;
    FControleAcessoListas: Boolean;
public
    property RelogioPontoEletronico: Boolean          read FRelogioPontoEletronico write
FRelogioPontoEletronico;
    property QtdeColaboradoresMarcamPonto: Integer read FQtdeColaboradoresMarcamPonto write
FQtdeColaboradoresMarcamPonto;
    property AcertoCentralizado: Boolean            read FAcertoCentralizado write
F AcertoCentralizado;
    property EscalaRevezamento: Boolean           read FEscalaRevezamento write
F EscalaRevezamento;
    property HorasExtras: Boolean                  read FHorasExtras write FHorasExtras;
    property HorarioNoturno: Boolean               read FHorarioNoturno write FHorarioNoturno;
    property HorarioFlexivel: Boolean              read FHorarioFlexivel write
FHorarioFlexivel;
    property DescontaDSR: Boolean                  read FDescontaDSR write FDescontaDSR;
    property QtdeTurnos: Integer                   read FQtdeTurnos write FQtdeTurnos;
    property PossuiBancoHoras: Boolean             read FPossuiBancoHoras write
FPossuiBancoHoras;
    property PossuiRefeitório: Boolean            read FPossuiRefeitório write
FPossuiRefeitório;
    property DescontaRefeicao: Boolean              read FDescontaRefeicao write
FDescontaRefeicao;
    property QtdePortarias: Integer                read FQtdePortarias write FQtdePortarias;
    property AcessoRestrito: Boolean               read F AcessoRestrito write F AcessoRestrito;
    property ControleAcessoListas: Boolean         read FControleAcessoListas write
FControleAcessoListas;

    function CalcularPontosImplantacao: Integer; reintroduce;
end;

TRS = class(TModulo)
private
    FQtdeAdmissoes: Integer;
public
    property QtdeAdmissoes: Integer read FQtdeAdmissoes write FQtdeAdmissoes;

    function CalcularPontosImplantacao: Integer; reintroduce;
end;

TTR = class(TModulo)
private
    FQtdeTurmasTreinamentoMes: Integer;
public
    property QtdeTurmasTreinamentoMes: Integer read FQtdeTurmasTreinamentoMes write
FQtdeTurmasTreinamentoMes;

    function CalcularPontosImplantacao: Integer; reintroduce;
end;

TCS = class(TModulo)
private
    FQtdeCargos: Integer;
public
    property QtdeCargos: Integer read FQtdeCargos write FQtdeCargos;

    function CalcularPontosImplantacao: Integer; reintroduce;
end;

TEstimativa = class
private
    FLevantamentoInicial: TLevantamentoInicial;
    FInstalacao: TInstalacao;
    FTreinamento: TTreinamento;
    FValidacaoAceite: TValidacaoAceite;
    FModulos: TList;
    FCliente: TCliente;
    procedure SetModulo( I: Integer; aModulo: TModulo );
    function GetModulo( I: Integer ): TModulo;
public

```

```
property Cliente: TCliente read FCliente write FCliente;
property LevantamentoInicial: TLevantamentoInicial read FLevantamentoInicial write
FLevantamentoInicial;
property Instalacao: TInstalacao read FInstalacao write FInstalacao;
property Treinamento: TTreinamento read FTreinamento write FTreinamento;
property ValidacaoAceite: TValidacaoAceite read FValidacaoAceite write FValidacaoAceite;

property Modulos[ I: Integer ]: TModulo read GetModulo write SetModulo;

constructor Create;
destructor Destroy; override;
procedure AdicionarModulo( aModulo: TModulo );
procedure RemoverModulo( aIndex: Integer );
function QtdeModulos: Integer;

end;
```

REFERÊNCIAS BIBLIOGRÁFICAS

AZAMBUJA, Telmo Travassos de. **Documentação de sistemas da qualidade: um guia prático para a gestão das organizações**. Rio de Janeiro: Campus, 1996.

AZEVEDO, Douglas José Peixoto de. **FPA-Function Point Analysis–Sistema de Métrica**, [S.l.], 2003. Disponível em: <<http://www.softwaremetrics.com>>. Acesso em: 02 mar. 2003.

BRAGA, Antônio. **Análise por pontos de função**. Rio de Janeiro : Infobook, 1996.

CANTÚ, Marco. **Dominando o Delphi 5: a bíblia**. São Paulo: Makron Books, 2000.

DeMARCO, Tom. **Controle de projetos de software: gerenciamento, avaliação, estimativa**. Trad. Maria E. Comenale, Áurea C. T. Dal Bo, Norma Pinto de Carvalho. Rio de Janeiro: Campus, 1989.

FREIRE, Agnaldo. **Engenharia de implantação – uma abordagem metodológica**. 2000. 122 f. Dissertação (Mestrado em Informática) – Instituto de Informática, Pontifícia

FURLAN, José Davi. **Modelagem de objetos através da UML – The Unified Modeling Language**. São Paulo: Makron Books, 1998.

INTERNATIONAL FUNCTION POINT USER GROUP. **Análise por pontos de função**. [S.l.]:IFPUG, 1991. (Baseado na Release 3.4 do Manual de Práticas de Contagem do IFPUG).

LONGSTREET, David. [S.l.], 2003. Disponível em: <<http://www.celepar.br/batebyte/bb70/fpa.htm>>. Acesso em: 10 jan. 2003.

MICRODOFT CORPORATION. MSDN, [S.l.], 2003. Disponível em: <<http://www.microsoft.com/msdn>>. Acesso em: 28 fev. 2003.

OLIVEIRA, William Lopes de. **Visão geral do PMBOK**, São Paulo, PMI - Project Management Institute, 2002.

PMI – PROJECT MANAGEMENT INSTITUTE. **A guide to the project management body of knowledge**. [S.l.], 1996. Disponível em: <<http://www.pmi.org>>. Acesso em: 18 abr. 2003.

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

QUATRANI, Terry. **Modelagem visual com Rational Rose 2000 e UML**. Rio de Janeiro: Ciência Moderna, 2001.

SENIOR SISTEMAS LTDA. **Manual do Sistema Vetorh**. Blumenau, Senior Sistemas, 2002.

SOMMERVILLE, Ian. **Engenharia de Software – quarta edição**. Campinas: Universidade Católica, 1992.