

UNIVERSO - UNIVERSIDADE SALGADO DE OLIVEIRA – Campus – Goiânia  
CURSO DE SISTEMAS DE INFORMAÇÃO

UM ESTUDO DE MÉTRICAS DE SOFTWARE ORIENTADA A PONTOS POR  
FUNÇÃO  
NA ENGENHARIA DE SOFTWARE

Charles Fernandes Cardoso Júnior

Guilherme Mendes Magalhães

Paulo Alves da Silva

Goiânia  
Maio – 2006/1

UNIVERSO - UNIVERSIDADE SALGADO DE OLIVEIRA – Campus – Goiânia  
CURSO DE SISTEMAS DE INFORMAÇÃO

UM ESTUDO DE MÉTRICAS DE SOFTWARE ORIENTADA A PONTOS POR  
FUNÇÃO  
NA ENGENHARIA DE SOFTWARE

Trabalho apresentado para avaliação do rendimento acadêmico na disciplina de Estágio Supervisionado, do curso de Sistema de Informação da Universidade Salgado de Oliveira, orientado pela prof. Ana Carolina do Prado.

Charles Fernandes Cardoso Júnior  
Guilherme Mendes Magalhães  
Paulo Alves da Silva

Goiânia  
Maio – 2006/1

*Dedico este trabalho ao Todo Poderoso,  
fonte de toda sabedoria e conhecimento e à  
minha mãe, pelo apoio à minha formação  
acadêmica.*

Charles Fernandes Cardoso Júnior

*Dedico este trabalho ao único e infinito  
Deus, minha família e a minha noiva.*

Guilherme Mendes Magalhães

*Dedico este trabalho à minha esposa,  
Nilvanete, pelo seu constante incentivo e  
às minhas filhas, Jéssica, Andressa e  
Rhandrya, minha fonte de inspiração.*

Paulo Alves da Silva

## **Agradecimentos**

*De Charles Fernandes Cardoso Júnior:*

Ao professor Edigar Antônio Diniz, o qual ministrou a disciplina de engenharia de software com excelência, despertando em mim interesse por este assunto. Agradeço também a empresa Seta Sistemas e o seu diretor Ralph Waldo Rangel, por ter aberto as portas da empresa a fim de serem utilizados os recursos que fossem necessários. Agradeço aos colegas que se empenharam com dedicação para a elaboração deste trabalho e a todos aqueles que tiveram uma participação direta e indireta.

E por fim, agradeço a minha mãe que tem possibilitado a realização de um sonho.

## **Agradecimentos**

*De Guilherme Mendes Magalhães:*

Ao professor Edigar Antônio Diniz, o qual abriu as portas do conhecimento em relação à engenharia de software e a todo conhecimento utilizado para moldar a concretização deste trabalho. Agradeço também a minha família pela paciência, compreensão e apoio e ao Ralph Waldo Rangel por ter facilitado nossa permanência em sua empresa.

E por fim agradeço a todos aqueles que contribuíram direta ou indiretamente para a formação deste trabalho.

## **Agradecimentos**

*De Paulo Alves da Silva:*

À Prof. Mestre Ana Carolina do Prado, minha orientadora neste trabalho, pelo seu apoio e paciência com que sempre nos recebeu.

Aos amigos, Charles Fernandes Cardoso Junior e Guilherme Mendes Magalhães, companheiros de estudos, pela força, amizade e enorme ajuda neste trabalho.

E a Deus pela força de superar os obstáculos e vencer os desafios.

*“O desejo do coração do preguiçoso é: depois de dormir... descansar”*

(Autor desconhecido)

## Sumário

Introdução .....	19
1.1. Apresentação .....	21
1.2. Definição do problema .....	22
1.3. Definição da proposta .....	23
1.4. Justificativa .....	24
1.5. Objetivos da proposta .....	25
2. Software .....	26
3. Engenharia de Software .....	32
3.1. Ciclo de vida clássico .....	36
3.2. Prototipação .....	38
3.3. Modelo Espiral .....	40
3.4. Técnicas da Quarta Geração .....	41
4. Métricas de Software e Pontos por Função .....	43
4.1. Métricas orientadas à pontos de função .....	46
4.1.1. Histórico sobre pontos por função .....	48
4.1.2. Processo de Contagem de Pontos de Função .....	50
4.1.3. Descrição de como avaliar os critérios de pontuação .....	52
5. Estudo de Caso .....	63
5.1. Contagem de função de dados .....	63
5.2. Contagem de funções transacionais .....	65
5.3. Contagem parcial do Sistema .....	66
5.3.1. Contagem de função de Dados .....	70
5.3.2. Contagem de funções transacionais .....	87
6. Treinamento .....	99
6.1 – Resultados do workshop .....	113
Considerações Finais .....	117
Glossário .....	119
Fontes Consultadas .....	121

## Lista de Figuras

Figura 3.01 – Modelo Clássico .....	37
Figura 3.02 – Protótipo .....	39
Figura 3.03 – Modelo Espiral .....	41
Figura 3.04 – Modelo 4GL .....	42
Figura 5.01 – Entidade RotaMapa .....	64
Figura 5.02 – Tela Cadastro de município .....	65
Figura 5.03 – Tela Procura .....	66
Figura 5.04 – Gerar relatório .....	67
Figura 5.05 – Relatório Gerado .....	68
Figura 6.01 – Slide Introdução .....	101
Figura 6.02 – Slide Objetivos .....	101
Figura 6.03 – Slide o que são métricas de software? .....	102
Figura 6.04 – Slide por que medir software? .....	102
Figura 6.05 – Slide por que medir software? (2) .....	103
Figura 6.06 – Slide em resumo .....	103
Figura 6.07 – Slide categorização de métricas .....	104
Figura 6.08 – Slide categorização de métricas (2) .....	104
Figura 6.09 – Slide os quatros papéis de medição .....	105
Figura 6.10 – Slide principais barreiras .....	105
Figura 6.11 – Slide mas podemos contornar .....	106
Figura 6.12 – Slide mas podemos contornar (2) .....	106
Figura 6.13 – Slide o que é o processo de contagem .....	107
Figura 6.14 – Slide funções de dados .....	107
Figura 6.15 – Slide tabela de identificação complexidade FD .....	108
Figura 6.16 – Slide funções de transações .....	108
Figura 6.17 – Slide tabela de identificação complexidade EE .....	109
Figura 6.18 – Slide identificação da complexidade .....	109
Figura 6.19 – Slide contribuição das funções .....	110

Figura 6.20 – Slide determinação do fator de ajuste .....	110
Figura 6.21 – Slide características gerais do sistema .....	111
Figura 6.22 – Slide características gerais do sistema (2) .....	111
Figura 6.23 – Slide referências .....	112
Figura 6.24 – Slide contatos .....	112

## Lista de Tabelas

Tabela 4.01 – Tabela de complexidade .....	51
Tabela 5.01 – Entidade AnaliseQua .....	70
Tabela 5.02 – Entidade Banco .....	70
Tabela 5.03 – Entidade BonificacaoFaz .....	70
Tabela 5.04 – Entidade BonificacaoMun .....	70
Tabela 5.05 – Entidade BonificacaoReg .....	70
Tabela 5.06 – Entidade BTLC .....	70
Tabela 5.07 – Entidade BTLCMov .....	71
Tabela 5.08 – Entidade BTLCPrep .....	71
Tabela 5.09 – Entidade BTLCPrepFazenda .....	71
Tabela 5.10 – Entidade CompradorReg .....	71
Tabela 5.11 – Entidade Convenio .....	71
Tabela 5.12 – Entidade ConvenioDesfaz .....	71
Tabela 5.13 – Entidade ConvenioFaz .....	72
Tabela 5.14 – Entidade ConvenioFaz2 .....	72
Tabela 5.15 – Entidade ConvenioFazPar .....	72
Tabela 5.16 – Entidade ConvenioVei .....	72
Tabela 5.17 – Entidade ConvenioVeiPar .....	72
Tabela 5.18 – Entidade Cota .....	72
Tabela 5.19 – Entidade Crioscopia .....	73
Tabela 5.20 – Entidade CustoLeiteFaz .....	73
Tabela 5.21 – Entidade DefBancaria .....	73
Tabela 5.22 – Entidade DigitoVerificador .....	73
Tabela 5.23 – Entidade Evento .....	73
Tabela 5.24 – Entidade EventoPar .....	73
Tabela 5.25 – Entidade EventoVerba .....	74
Tabela 5.26 – Entidade FaixaQual .....	74

Tabela 5.27 – Entidade Fazenda .....	74
Tabela 5.28 – Entidade FazendaFre .....	74
Tabela 5.29 – EntidadeFazendaPer .....	74
Tabela 5.30 – Entidade FazendaPer .....	74
Tabela 5.31 – Entidade FazendaTan .....	75
Tabela 5.32 – Entidade FornecedorLeite .....	75
Tabela 5.33 – Entidade Funcionario .....	75
Tabela 5.34 – Entidade IndicadorFaz .....	75
Tabela 5.35 – Entidade laboratorio .....	75
Tabela 5.36 – Entidade LimiteLit .....	75
Tabela 5.37 – Entidade ListaLeite .....	76
Tabela 5.38 – Entidade Litragem .....	76
Tabela 5.39 – Entidade LitragemExc .....	76
Tabela 5.40 – Entidade LoCodigoBarras .....	76
Tabela 5.41 – Entidade LoRemessa .....	76
Tabela 5.42 – Entidade LoRemessaCab .....	76
Tabela 5.43 – Entidade LoRemessaDet .....	77
Tabela 5.44 – Entidade LoRemessaGru .....	77
Tabela 5.45 – Entidade LoRemessaMsgCar .....	77
Tabela 5.46 – Entidade LoRemessaMsgEspCar .....	77
Tabela 5.47 – Entidade LoRemessaRod .....	77
Tabela 5.48 – Entidade LoRemessaRodGru .....	77
Tabela 5.49 – Entidade LoRemessaSubDet .....	78
Tabela 5.50 – Entidade LoRetorno .....	78
Tabela 5.51 – Entidade MarcaTanque .....	78
Tabela 5.52 – Entidade Memorando .....	78
Tabela 5.53 – Entidade MemorandoFaz .....	78
Tabela 5.54 – Entidade MemorandoRot .....	78
Tabela 5.55 – Entidade MemorandoUni .....	79
Tabela 5.56 – Entidade MemorandoVei .....	79

Tabela 5.57 – Entidade MicroRegiao .....	79
Tabela 5.58 – Entidade MotivoMem .....	79
Tabela 5.59 – Municipio .....	79
Tabela 5.60 – Entidade OcorrenciaConV .....	79
Tabela 5.61 – Entidade PagamentoFazTot .....	80
Tabela 5.62 – Entidade PagamentoFazTot .....	80
Tabela 5.63 – Entidade PagamentoVei .....	80
Tabela 5.64 – Entidade PagamentoVeiTot .....	80
Tabela 5.65 – Entidade Paramentro .....	80
Tabela 5.66 – Entidade ParametroAdc .....	80
Tabela 5.67 – Entidade ParametroCBT .....	81
Tabela 5.68 – Entidade ParametroCCS .....	81
Tabela 5.69 – Entidade ParametroGor .....	81
Tabela 5.70 – Entidade ParametroPro .....	81
Tabela 5.71 – Entidade ParametrosSis .....	81
Tabela 5.72 – Entidade Percurso .....	81
Tabela 5.73 – Entidade Periodo .....	82
Tabela 5.74 – Entidade PeriodoPar .....	82
Tabela 5.75 – Entidade Ponto .....	82
Tabela 5.76 – Entidade Preco .....	82
Tabela 5.77 – Entidade Preco2 .....	82
Tabela 5.78 – Entidade Preco3 .....	82
Tabela 5.79 – Entidade PreparaDesConF .....	83
Tabela 5.80 – Entidade PreparaDesConV .....	83
Tabela 5.81 – Entidade Produtor .....	83
Tabela 5.82 – Entidade QualidadeRec .....	83
Tabela 5.83 – Entidade RecepcaoBan .....	83
Tabela 5.84 – Entidade Regiao .....	83
Tabela 5.85 – Entidade Relatorio .....	84
Tabela 5.86 – Entidade Rota .....	84

Tabela 5.87 – Entidade RotaMapa .....	84
Tabela 5.88 – Entidade RotaPer .....	84
Tabela 5.89 – Entidade SituacaoPagFaz .....	84
Tabela 5.90 – Entidade SituacaoPagVei .....	84
Tabela 5.91 – Entidade SupervisorReg .....	85
Tabela 5.92 – Entidade TabelaQua .....	85
Tabela 5.93 – Entidade TicketFazenda .....	85
Tabela 5.94 – Entidade Transportador .....	85
Tabela 5.95 – Entidade Unidade .....	85
Tabela 5.96 – Entidade Usuario .....	85
Tabela 5.97 – Entidade UsuarioAcesso .....	86
Tabela 5.98 – Entidade UsuarioSistema .....	86
Tabela 5.99 – Entidade UsuarioUnidade .....	86
Tabela 5.100 – Entidade Veiculo .....	86
Tabela 5.101 – Entidade VeiculoPer .....	86
Tabela 5.102 – Entidade VeiculoPer .....	86
Tabela 5.103 – Tela Cadastro Rota .....	87
Tabela 5.104 – Tela Funcionario .....	87
Tabela 5.105 – Tela Cadastro de Municipio .....	87
Tabela 5.106 – Tela Parâmetros .....	88
Tabela 5.107 – Tela Cadastro de Região .....	88
Tabela 5.108 – Tela Cadastro de Micro-Região .....	88
Tabela 5.109 – Tela Cadastro de Unidade .....	89
Tabela 5.110 – Tela Cadastro de Banco .....	89
Tabela 5.111 – Tela Preço por Unidade .....	89
Tabela 5.112 – Tela Preço por veículo .....	90
Tabela 5.113 – Tela Preço por nota .....	90
Tabela 5.114 – Tela Cadastro de Evento .....	90
Tabela 5.115 – Tela Cadastro de Percurso .....	91
Tabela 5.116 – Tela Cadastro de Tanque de Resfriamento .....	91

Tabela 5.117 – Tela Cadastro de Fornecedor .....	91
Tabela 5.118 – Tela Cadastro de Crioscopia .....	92
Tabela 5.119 – Tela Unidades liberadas para o usuário .....	92
Tabela 5.120 – Tela Cadastro de Ponto .....	92
Tabela 5.121 – Tela Cadastro de ocorrência de litragem .....	93
Tabela 5.122 – Tela Preparação de BTLC .....	93
Tabela 5.123 – Tela BTLC Fazenda .....	93
Tabela 5.124 – Tela BTLC Transferência .....	94
Tabela 5.125 – Tela BTLC Compra .....	94
Tabela 5.126 – Tela BTLC Socorro .....	94
Tabela 5.127 – Tela Cadastro de motivo de memorando .....	95
Tabela 5.128 – Tela Memorando .....	95
Tabela 5.129 – Tela Cadastro de período de pagamento .....	95
Tabela 5.130 – Tela Cadastro de produtor .....	96
Tabela 5.131 – Tela Altera Frete .....	96
Tabela 5.132 – Tela Fecha Lista Leite .....	96
Tabela 5.133 – Tela Transfere dados da Fazenda entre Rotas .....	96
Tabela 5.134 – Tela Integração .....	96
Tabela 5.135 – Tela Transfere dados da fazenda entre rotas .....	97
Tabela 5.136 – Tela Relatório de município sem banco .....	97
Tabela 5.137 – Tela Relatório de banco por município .....	97
Tabela 5.138 – Tela Relatório de fazenda por produtor .....	97
Tabela 5.139 – Tela Relatório fazenda por região .....	97
Tabela 5.140 – Tela Relatório de mapa da rota .....	97

## **Lista de Siglas**

### **A**

**AIE** - Arquivo de Interface Externa

**ALI** - Arquivo Lógico Interno

**APF** - Análise de Pontos por Função

**AR** - Arquivo Referenciado

### **C**

**CE** - Consulta Externa

**CFPS** - Certified Function Point Specialist – Certificado Especializado em Pontos de função

**CPM** - Counting Practices Manual (Manual Prático de Contagem)

**CPU** - Central Processing Unit (Unidade Central de Processamento)

### **E**

**EE** - Entrada Externa

### **F**

**FAV** - Fator de Ajuste de Valor

## **I**

**ID** - Item de Dado

**IFPUG** - International Function Point Users Group (Grupo Internacional de Usuários de Pontos por Função)

**ISO** - International Organization for Standardization (Organização Internacional de Padronização)

**ISO IEC/JTCL/SC7/WG12** - Padrão ISO para Medida Funcional de Tamanho - Grupo de Trabalho 12

## **P**

**PF** - Pontos Por Função

## **R**

**RL** - Registro Lógico

## **S**

**SE** – Saida Externa

## **INTRODUÇÃO**

Na atualidade, quando observados os diversos acontecimentos profissionais e gerenciais que permeiam a existência humana, esta depara-se com um ponto que se tornou fundamental para a sobrevivência da humanidade dentro do contexto tecnológico e empresarial, ou seja, softwares. Tal sobrevivência deixou de ter apenas um mero sentido de existência da tecnologia em suas raízes e avançou para o estágio de dependência. Essa dependência moldada pela tecnologia da informação leva as empresas a se empenharem em obter cada vez mais recursos que as auxiliem nas tomadas de decisões de forma rápida, coerente e eficaz, para assim, manter-se em uma posição elevada dentro do cenário comercial.

Em meio a junção de tecnologia da informação com a criação de ferramentas que venha a prover uma maior iteração entre o homem e o computador, surge a essencialidade de se manter uma posição de domínio das demandas crescentes de softwares com seus recursos, requisitos e diretrizes, que são regidos pelas diversas áreas existentes que necessitam obter, moldar, gerenciar e disponibilizar informações pertinentes às regras de negócios solicitadas. Dentro desse cenário, existe a Engenharia de Software que permite manter um certo domínio nas etapas de criação e manutenção de um software, de forma a prover uma maior qualidade e aceitação por parte daqueles que usarão softwares para suprir suas necessidades.

Este trabalho pertencente à disciplina de estágio supervisionado, tem como proposta abordar os assuntos relacionados à Engenharia de Software e

métricas de softwares por pontos de função para assim, entender seus conceitos e aplicabilidades e a partir de então, conceder soluções e práticas para uma melhor obtenção de resultados referentes ao desenvolvimento e qualidade dos softwares.

Para uma melhor abordagem, contamos com o apoio e presteza da empresa Seta Sistemas a qual prontamente aceitou a nossa proposta e assim possibilitou a evolução das idéias que estão sendo brotadas pelas pessoas que se prontificaram a levar adiante essa obra. Será percebido então, no decorrer das etapas, que o foco principal está em métricas de softwares orientados à pontos de função o qual será aplicado através de um estudo de caso, ou seja, uma simulação de contagem, que será aplicado em um dos softwares desenvolvido pela empresa participante deste estágio. Este é um ponto de partida para uma grande evolução e conquista, que serão alcançadas por aqueles que se envolveram na viabilização deste trabalho.

## **1.1 – Apresentação**

A Empresa Seta Sistemas, a qual tem participação fundamental neste trabalho, surgiu em meados de 1995, com atuação em desenvolvimento de softwares, junto a empresas que necessitam utilizar os recursos de informática para um melhor desempenho de suas atividades. A Seta Sistemas propõe soluções aos seus clientes com softwares acadêmicos, de laticínios, restaurantes e está em fase de desenvolvimento o software de gestão empresarial.

Buscando melhorar a qualidade em seus produtos, a empresa obteve a certificação ISO 9001 a qual é uma certificação voltada para a qualidade. A Seta Sistemas, começará nos próximos dias o processo de certificação em MPS-BR que significa: certificação de melhoria do processo do software brasileiro.

## **1.2 – Definição do problema**

Como aconteceram com muitas empresas que trabalham com desenvolvimento, a Seta Sistemas começou suas atividades para suprir uma necessidade de um cliente específico. No entanto, com o passar do tempo e com o empenho da equipe de desenvolvimento a empresa cresceu e conquistou uma posição no cenário comercial, no entanto, sem aplicar a Engenharia de Software.

O fato de não ter aplicado a Engenharia de Software é o ponto fundamental que é caracterizado como problema, pois, segundo Pressman [PRE05], uma empresa de desenvolvimento, que não aplica a Engenharia de Software na construção/manutenção de seu produto, poderá entrar em uma certa desordem, ineficiência e ineficácia, mediante ao seu crescimento e expansão do produto. As métricas de software, uma ramificação da Engenharia de Software, permitem obter o tamanho do software através de medidas diretas e indiretas, e a não aplicabilidade das mesmas, poderá prover uma diminuição evolutiva nos processos de desenvolvimento do software, tais como: qualidade do produto, qualidade das pessoas que produzem o produto e outros fatores inerentes as etapas de desenvolvimento. A empresa Seta Sistemas não faz uso das métricas de software para medir o tamanho de seus softwares.

### **1.3 – Definição da Proposta**

É proposto à Seta Sistemas, um trabalho voltado para a Engenharia de Software com ênfase na aplicação de métricas, usando como medida de contagem a técnica baseada em pontos por função, para que a empresa venha a conhecer a existência e aplicabilidade de métricas de software. Dentro deste contexto será feito uma contagem parcial de um software que será fornecido pela empresa para uma aplicação formal da técnica de pontos de função, bem como um treinamento para que a empresa venha a ter um conhecimento mais detalhando sobre o tema que está sendo abordado. É Desejável que depois que a empresa obtver o conhecimento dos benefícios de tal recurso ela venha a analisar se de fato é viável começar a aplicar métricas de software em seu produto e com isso aderir por completo a todos os recursos que são oferecidos pela Engenharia de Software.

#### **1.4 - Justificativa**

Diante da Lacuna que permeia o desenvolvimento de software, aonde não se tem o hábito das empresas em fazer medição do seu produto, levando-as a não usufruírem dos benefícios obtidos com as métricas de software, criou-se então o desejo de obter mais informações sobre o assunto. A real motivação em desenvolver um trabalho abordando métricas de software é devido ao interesse em aprofundar os conhecimentos em Engenharia de Software, podendo assim, através de pesquisas, obter a informação e o conhecimento sobre o assunto e assim, poder praticar essa técnica nas empresas de desenvolvimento de software, levando as mesmas a cultivarem essa abordagem, provendo assim uma maior qualidade no processo de desenvolvimento do software brasileiro.

## **1.5 – Objetivos da Proposta**

Conforme foi abordado anteriormente, existe um desejo em levar à empresa Seta Sistemas o conhecimento, aplicabilidade e uso das métricas de software para que a mesma possa avaliar a viabilidade de uma possível aplicação. Para conseguir isso, é objetivado o seguinte:

- Apresentar métricas de software à empresa
- Abordar os benefícios de métricas de software
- Mostrar os tipos de métricas de software
- Apresentar métricas de pontos por função
- Fazer uma contagem parcial em um dos softwares da empresa, usando a técnica de pontos por função
- Prover um rápido treinamento de contagem usando pontos por função.

## 2.SOFTWARE

### Uma descrição de Software:

Muitos têm uma visão parcial e limitada sobre o que é um software. Há quem pense que o software é apenas um programa de computador que foi codificado por alguma pessoa usando uma linguagem de programação qualquer gerando assim um produto que é usado pelo usuário e pronto. Ainda que o produto desenvolvido interaja com o usuário para que o mesmo venha a utilizar os recursos da tecnologia da informação agregados as regras de negócio da empresa, não quer dizer que tal produto em sua “individualidade” satisfaça por completo uma descrição concisa sobre o que de fato vem a ser um software. A descrição completa de um software vai além de simplesmente o programa que foi desenvolvido (a ferramenta que é concebida para atender às necessidades do usuário), segundo Pressman [PRE05] a sua completude é efetivada mediante a composição de três fatores: programa de computador, estrutura de dados e documentação.

\* **Programa de computador** – Instruções que são executadas para realizar a função e o desempenho desejado, mediante requisições que foram levantadas pelo usuário e identificadas pelo analista de sistemas. O programa de computador é o produto que foi gerado a partir de uma linguagem de programação para ser uma ferramenta que irá ajudar o profissional de um segmento qualquer a proceder em seu trabalho com uma maior eficiência e eficácia.

\* **Estruturas de dados** – Trata-se da estrutura de dados que permitirá ao programa manipular adequadamente a informação. Uma vez que uma determinada informação é usada e/ou manipulada pelo programa, torna-se necessário uma forma de preservar essa informação de forma a proporcionar um armazenamento e uma futura recuperação quando se tornar necessário.

\* **Documentação** - Documentos que descrevem como operar e usar os programas são necessários para que o usuário possa tirar maior proveito dos recursos que foram disponibilizados para o uso do software. Neste caso, o material de apoio serve para auxiliar o usuário, levando-o a saber como operar o produto.

Por mais que existam muitas definições sobre o que vem a ser um software, uma mera descrição formal ou informal não é capaz de estabelecer uma compreensão completa sobre o mesmo, pois existem várias definições com conceitos distintos, cada uma com suas argumentações e divisões. Para uma melhor visualização do software, serão explanadas suas características e aplicações, com o intuito de formalizar uma conclusão mais concisa sobre o mesmo.

#### Características do Software:

Em uma comparação com o hardware, o qual compõe uma outra grandeza do mundo computacional, o software não é manufaturado no sentido clássico, ou seja, ele não é desenvolvido a partir de peças que são unidas para a formação do produto, como acontece, por exemplo, na produção de uma placa mãe, a qual é composta de vários elementos distintos para completar a sua formação.

Ainda que exista o conceito de programação orientada a objetos - o que significa que você cria um molde e a partir de então concentra o desenvolvimento em utilizar algo que já está pronto ou parcialmente pronto mediante os diversos recursos oferecidos por este segmento - tal fato não quer dizer que o software seja composto de uma padronização de componentes, como acontecem com os diversos elementos físicos que compõem a estrutura de uma placa mãe aonde os mesmos são produzidos em séries e encaixados nos seus devidos lugares e interagindo de forma harmônica. Os custos que envolvem o desenvolvimento do hardware estão diretamente relacionados a manufatura dos elementos que o compõem, sendo diferente do custo de produção do software o qual está diretamente relacionado com o trabalho de engenharia.

O software provê vida e funcionalidade a uma estrutura física e não é algo palpável, ou seja, não é possível pegar um software com a mão e mudá-lo de lugar, ou até mesmo sentir o seu peso. O processo de duplicação do software, por exemplo, não requer que seja usado em si algum recurso proveniente da natureza como o silício, devido ao fato de ser uma estrutura lógica e não física.

Outra característica bastante peculiar e interessante do software é que o mesmo não se desgasta com os fenômenos da natureza, o que não é o caso do hardware, pois a queda e elevação de temperatura podem fazer com que o mesmo sinta o impacto levando a uma possível alteração de sua composição e funcionalidade. O efeito do tempo também causa uma influência direta na integridade de um componente físico e devido ao fato do software ser um produto lógico, “o fator tempo”, não o influenciará a ter um certo desgaste, pois o software jamais se desgasta. O software pode deteriorar mediante ao avanço tecnológico ou regras de negócios existentes, e também pelas necessidades de mudanças que devem ser aplicadas para que ele possa se evoluir.

## Aplicações do Software:

Um software é desenvolvido para satisfazer uma necessidade que possa ser originada de diversas formas possíveis. Pode ser que tal necessidade surja em uma área industrial, como por exemplo, a necessidade de se controlar uma máquina ou pode ser que seja de um fator comercial mediante uma requisição de melhor controlar a gestão de negócios. Dentre as aplicabilidades que pode ter um software, serão citadas algumas, sem se deter em maiores detalhes.

\* **Software Básico** - São programas escritos para dar apoio à outros programas ajudando-os em sua funcionalidade. Geralmente os softwares básicos funcionam como uma camada entre o hardware e um programa que necessita usar os recursos daquele hardware.

\* **Software de Tempo Real** - São softwares que provêm uma resposta imediata, geralmente variando de um milissegundo a um minuto. Esses softwares têm a característica de proverem uma resposta quase instantânea à sua solicitação.

\* **Software Comercial** - É voltado para facilitar as operações comerciais ajudando nas tomadas de decisões administrativas.

\* **Software Científico e de Engenharia** - São softwares utilizados no auxílio à projetos científicos e de engenharia, tais como, astronomia, vulcanologia. Os algoritmos desenvolvidos para este segmento abordam o processamento de números.

\* **Software Embutido** - São softwares usados para produtos inteligentes sendo armazenados em memória somente de leitura. Um exemplo típico deste tipo de software são programas de controle remoto.

\* **Software de Computador Pessoal** - São softwares desenvolvidos para atender necessidades pessoais e individuais de usuários. Um exemplo deste tipo de software é uma planilha eletrônica.

\* **Software de Inteligência Artificial** - Software que faz uso de algoritmos não numéricos para resolver problemas complexos.

Independente da área de atuação para a qual um software é criado, ele pode se deparar com diversos fatores que poderão implicar em uma insatisfação por parte do usuário e um resultado apático de seu funcionamento. Isso pode acontecer pelo fato do software não ter uma qualidade que venha a integrar uma forma harmônica com os recursos que ele deve utilizar, bem como resultados contrários aos requisitos que previamente foram identificados para que o software viesse a satisfazê-los.

Muitas empresas não criaram projetos para a construção de seus softwares, ou seja, os softwares foram desenvolvidos sem gerência de projetos, sem elaboração de estimativas e\ou sem análises de riscos, em suma, sem ser aplicado os moldes da engenharia de software. Devido a isso, percebe-se então que existe uma aflição que contaminou o desenvolvimento do software e tal aflição não desaparecerá da noite para o dia. Um ponto de partida é reconhecer os problemas existentes e suas causas, para assim poder identificar e tratar o que necessita ser resolvido.

A boa notícia é que essa lacuna está começando a ser preenchida devido a uma mudança nos paradigmas de desenvolvimento, aonde a concorrência está cada vez maior pela busca da qualidade e pela aceitação

plena do usuário em relação ao produto que foi desenvolvido para satisfazer as suas necessidades.

### 3. ENGENHARIA DE SOFTWARE

Existem muitas definições que tentam expressar o que é engenharia de software, no entanto, a primeira delas foi proposta por Fritz Bauer [PRE05] na primeira grande conferência que se realizou destinada ao assunto. Tal definição nos diz que:

*“O estabelecimento e uso de sólidos princípios para que se possa obter economicamente um software que seja confiável e que funcione eficientemente em máquinas reais.”*

Dentro dessa definição, pode-se retirar alguns fatores chaves que permeiam o desenvolvimento de software, seja de forma satisfatória ou insatisfatória: “uso de sólidos princípios”, “obter economicamente um software”, “que seja confiável”, “que funcione eficientemente”, “em máquinas reais”.

#### Uso de sólidos princípios:

Uma vez que é detectado uma não conformidade em relação a um certo procedimento qualquer, bem como se tratando de algum processo ou a forma de como algo é feito, surge a partir de então, princípios que possam moldar os novos produtos semelhantes àqueles que foram anteriormente acometido de erros, acertos ou qualquer outro fator que seja usável naquilo que está sendo produzido no momento. Essa abordagem evita perdas ou transtornos mediante a frustração de se ter uma etapa que não seja qualificada

a prover o acabamento do produto. Usar sólidos princípios levam a um caminho mais positivo em relação ao que se está produzindo e no caso do software proporciona uma melhor adequação das etapas que constituem a construção do mesmo, independente do paradigma da engenharia de software que for utilizado.

#### Obter economicamente um software:

Um dos fatores cruciais envolvidos no desenvolvimento do software é o seu custo. É necessário que se tenha um bom planejamento e gerência de projeto para assim, poder evitar gastos com recursos desnecessários. Obter economicamente um software possibilitará também que o mesmo seja disponibilizado com um valor que se encaixe adequadamente ao mercado provendo satisfação para ambos os lados que estejam envolvidos com o produto.

#### Que seja confiável:

A confiança é um dos fatores decisivos na aquisição de um software. Atualmente existe um grande poder de fluxo de informações bem como uma gama de dados armazenados que contém o que uma corporação tem de mais estratégico. Um software confiável é permeado por dois pontos fundamentais: O primeiro trata da integridade das informações pertencente às empresas que requerem o uso do software, pois, é necessário que essas informações estejam disponíveis em tempo hábil e que as mesmas estejam íntegras, sem perdas. Mediante a isso percebe-se então que é necessário ter uma boa estratégia de armazenamento, manipulação e recuperação por parte do software. O segundo fator objetiva a necessidade do software cumprir o que realmente foi requerido pelo usuário. Um algoritmo pode apresentar algum problema quando estiver entrando com informações que não sejam previamente tratadas e é necessário

assegurar que as rotinas do programa farão uso correto do ambiente, para assim, executar os procedimentos desejados de forma competente.

#### Que funcione eficientemente:

Um software é desenvolvido com um objetivo de executar uma tarefa de forma eficiente e eficaz. Isso nos diz que as regras de negócios devem ser estabelecidas e o conjunto de operações relacionadas a elas devem ser sanadas provendo a justificção do porque o software foi desenvolvido.

#### Em máquinas reais:

Máquinas reais provêm dois sentidos. O primeiro deles é que se torna necessário estabelecer uma configuração mínima de um equipamento para que o software venha a ser rodado. De nada adianta desenvolver um software adornado aos moldes da engenharia de software e colocá-lo para rodar em um computador que seja pobre em termos de recursos.

O Segundo sentido é que uma vez que o software é planejado, paradigmas são estabelecidos, estimativas são elaboradas, análise de riscos são abordadas e uma série de elementos são previamente levantados. É necessário que o projeto se transforme num produto que saia do papel e dos moldes da análise e possa de fato se tornar uma ferramenta sendo instalada em uma máquina real e assim poder ser usada para satisfazer as necessidades do usuário.

A Engenharia de Software tem como finalidade possibilitar o gerenciamento e controle dos processos de desenvolvimento de software, e para isso, conta com técnicas e regras que devem ser seguidas a risca para assim fazer com que o profissional possa ter uma base para a construção do software com alta qualidade e produtividade [PRE05]. A engenharia de

software abrange três elementos fundamentais: métodos, ferramentas e procedimentos.

### Métodos

Estão relacionados com o “como fazer” para construir o software e dentro desses métodos existem um conjunto de tarefas para moldar o desenvolvimento do produto como planejamento, estimativa, análise de requisitos entre outros.

### Ferramentas

Fornecem apoio automatizado ou semi-automatizado aos métodos. Essas ferramentas de apoio são chamadas de CASE (engenharia de software auxiliada por computador). O CASE combina software, hardware e um banco de dados de engenharia de software para tratar de informações importantes sobre a análise, projeto, codificação e teste, os quais são etapas do processo de desenvolvimento do software.

### Procedimentos

Estabelecem um elo entre métodos e ferramentas possibilitando o desenvolvimento racional e oportuno do software de computador. As finalidades dos procedimentos são estabelecer a seqüência em que os métodos serão aplicados e prover controles que ajudam a assegurar a qualidade entre outros.

Durante as etapas de desenvolvimento de um software é oferecido pela Engenharia de Software - assim como acontece em qualquer outra área que atua em projetos das mais variadas formas existentes - caminhos a serem

percorridos que envolvem os métodos, ferramentas e os procedimentos que foram discutidos anteriormente.

Existe uma linha a ser seguida mediante os esforços despendidos que permeiam as fases de criação e manutenção do produto que é efetivado pelos moldes da Engenharia de software. Essas linhas a serem seguidas são conhecidas como paradigmas da engenharia de software. Tais paradigmas são: ciclo de vida clássico, prototipação, modelo espiral, técnicas da quarta geração. Um paradigma é escolhido tendo como base a natureza do projeto e da aplicação, os métodos e as ferramentas a serem usados, os controles e os produtos que precisam ser entregues. Os paradigmas da Engenharia de Software permitem manter um maior controle e qualidade durante toda vida útil do software.

É considerado como vida útil todos os períodos de elaboração do mesmo, tendo como partida o momento em que surgiu a necessidade de se ter um software para uma especialidade qualquer até o ponto em que o mesmo se torna obsoleto, inadequado e é aposentado.

### **3.1 Ciclo de vida clássico**

Também conhecido como modelo cascata, este paradigma requer uma abordagem sistemática aonde existe uma execução seqüencial das etapas inerentes a ele. A execução através deste paradigma se dá a partir da engenharia de sistemas e seguindo seqüencialmente as demais etapas: análise – projeto – codificação – teste – manutenção. O ciclo de vida clássico é o paradigma mais antigo da engenharia de software. Existem problemas relacionados a este tipo de paradigmas, por exemplo:

- Dificilmente um projeto real segue o fluxo que o modelo propõe levando à uma aplicação do mesmo de forma não íntegra.

- Exige que o cliente declare todas as suas exigências explicitamente no começo do projeto.
- Se houver um erro crasso, que não seja detectado até que o programa de trabalho seja revisto, irá ocasionar resultados desastrosos.

Apesar dos problemas existentes neste modelo de paradigma, os quais foram citados acima, ele continua sendo o modelo procedimental mais amplamente usado pela Engenharia de Software.

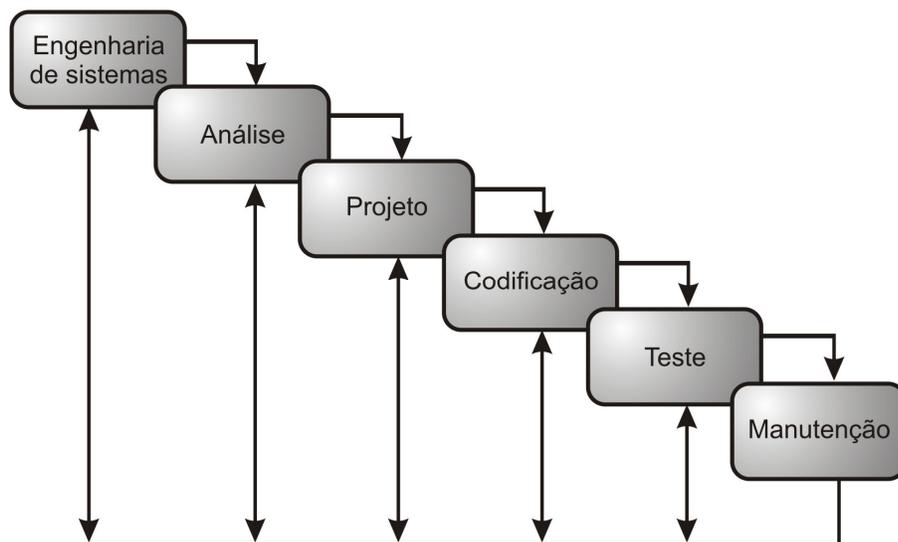


Figura 3.1 – Modelo Clássico

Engenharia de sistemas: Faz-se o levantamento dos requisitos de todos os elementos que comporão o sistema. Convém ressaltar que sistema trata-se dos componentes que farão agregação com o software, podendo ser: equipamentos, pessoas, ambiente físico e assim por diante.

Análise de requisitos de software: Faz o levantamento dos requisitos que formarão o software. É nesta etapa que se estabelecerá esforços para entender quais são as necessidades do usuário.

Projeto: Provê uma tradução das exigências relacionadas à análise dos requisitos para assim avaliar a qualidade antes que a codificação se inicie.

Codificação: Nesta etapa é gerado um conjunto de instruções que seja legível por máquina, para assim conceber o produto desejado.

Testes: Após o código ser gerado, será realizada uma série de testes para verificar e garantir que todas as operações que serão executadas pelo software estejam sendo efetivadas de forma correta.

Manutenção: Após ser entregue ao cliente, o software provavelmente sofrerá mudanças devidos a diversos fatores como erros ocasionais mediante a várias situações possíveis, ou por uma implantação de um novo recurso solicitado pelo usuário.

### **3.2 Prototipação**

O cliente na fase de análise de requisitos irá especificar suas necessidades para que a solução seja criada atendendo assim seu desejo. Mas este processo de especificação de suas necessidades poderá deixar de ser completo mediante o esquecimento de algum recurso que não foi especificado ou até mesmo pelo fato do usuário não saber como será feita a entrada de dados. Neste caso, uma pré-visualização de como ficará a interface homem-máquina poderá ajudar a identificar uma necessidade não abordada, ou até mesmo perceber que a aplicabilidade de um recurso qualquer que tenha sido abordado não é viável de ser continuado.

A proposta do modelo de prototipação é justamente prover uma pré-visualização de como ficará o software, em relação a interação do usuário com o mesmo, pois, através de protótipos é possível ao cliente saber como ficará a interface com suas entradas e saídas de informações. Tal procedimento ajudará a ter uma maior completude e certeza sobre o produto que está sendo criado.

Segundo Pressman [PRE05], a prototipação é um processo que capacita o desenvolvedor a criar um modelo do software que será implementado. Tal modelo pode ser fornecido através de três formas, as quais são:

- \* Protótipo em papel ou modelo baseado em PC que retrata a interação homem máquina de uma forma que capacita o usuário a entender quanta interação ocorrerá;

- \* um protótipo de trabalho que implementa algum subconjunto da função exigida do software desejado;

- \* um programa existente que executa a função desejada, ou parte dela, mas que tem outras características que serão melhoradas em um novo esforço de desenvolvimento.

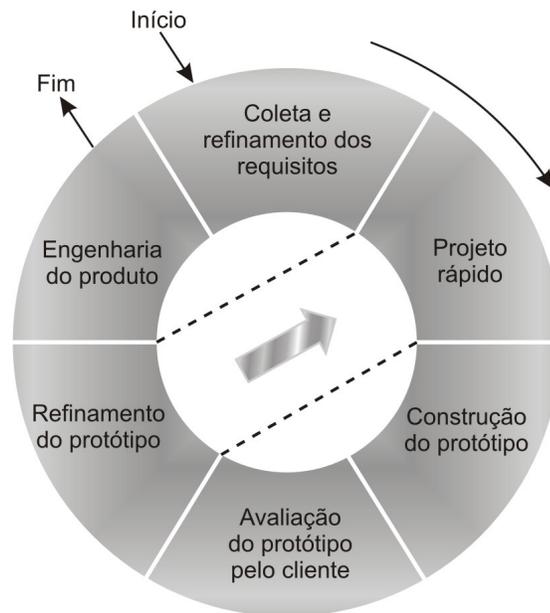


Figura 3.2 – Protótipo

Á idéia do protótipo é servir como um mecanismo que identificará de uma forma mais concisa os recursos do software. A grande questão relacionada aos protótipos é o que fazer com ele após o cumprimento de sua finalidade, ou seja, prover um entendimento visual de como será a aplicação,

no entanto, existem proponentes à utilização do protótipo na continuidade do desenvolvimento de software, aproveitando o que já está pronto para minimizar o esforço e aproveitar o tempo que foi gasto com o mesmo. Há também pessoas que são proponentes a não utilizar o protótipo na continuidade da produção do software, sendo defensores da iniciativa de se começar o projeto a partir do zero sem reutilizar o protótipo para tal. Para reforçar tal polêmica deixaremos uma explanação feita por Brooks [PRE05]:

*“Na maioria dos projetos, o primeiro sistemas construído dificilmente é usável. Ele pode ser muito lento, muito grande, desajeitado em uso, ou todos os três. Não resta outra alternativa senão começar tudo de novo, dolorosamente, mas com mais habilidade, e reconstruir uma versão novamente projetada em que esses problemas sejam resolvidos.. Quando um novo conceito de sistema ou nova tecnologia é usada, alguém tem de construir um sistema para jogar fora, porque até mesmo o melhor planejamento não é tão onisciente a ponto de fazê-lo corretamente logo na primeira vez. A questão administrativa, portanto, não é se se deve construir um sistema-piloto e jogá-lo fora. Isso será feito. A única questão é se se deve planejar antecipadamente a construção de algo que se vai jogar fora ou prometer entregar isso aos clientes...”*

### **3.3 Modelo Espiral**

Veio com uma proposta de agregar as melhores características que envolvem o ciclo de vida clássico e o da prototipação, mas com um adicional

voltado para a análise de riscos. A idéia do modelo espiral é fazer com que todas as suas etapas sejam abordadas de forma progressiva antes que o produto seja finalizado. A cada giro na espiral no sentido de dentro para fora é então contemplado todas as suas fases, as quais são: planejamento, análise de riscos, Engenharia, Avaliação do Cliente.

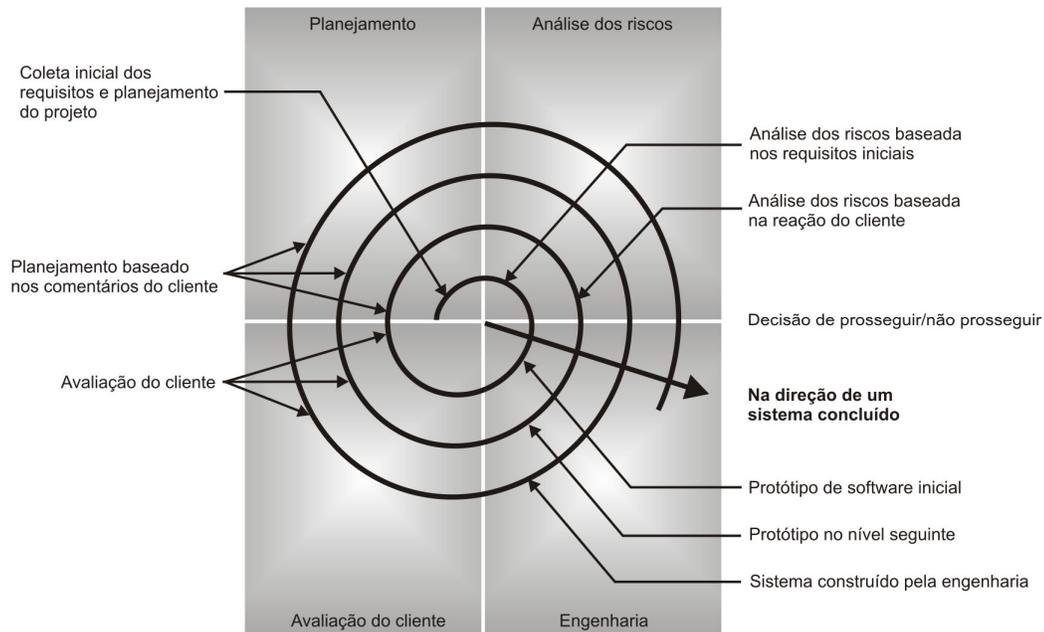


Figura 3.3 – Modelo Espiral

### 3.4 Técnicas da Quarta Geração

Trata-se do uso de ferramentas de software no auxílio do desenvolvimento especificando alguma característica do software em um nível mais elevado. Essas ferramentas que compõem a técnica da quarta geração são capazes de gerar o código fonte, criar o banco de dados e propor ao desenvolvimento uma maneira rápida de se construir um software.

Como muitos dos assuntos abordados pela engenharia de software existem os proponentes e os opositores. Os proponentes afirmam que com o uso da 4GT (técnicas da quarta geração) existe um ganho em relação ao

desenvolvimento, pois o produto é feito de uma forma rápida e existe uma elevação na produção de quem construiu o software. Já os opositores, defendem que o código gerado por tais ferramentas é ineficiente e a manutenibilidade dos mesmos será mais difícil de serem efetuadas.

Apesar dos prós e contras que permeiam este paradigma, a verdade é que atualmente o uso de 4GT já se tornou uma parte importante do desenvolvimento de software no cenário mundial.

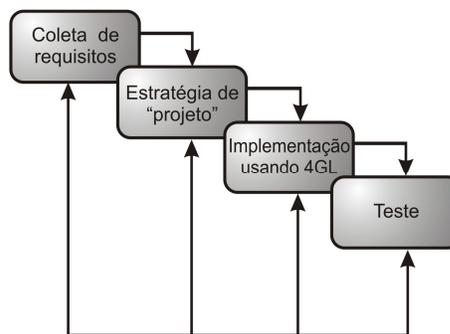


Figura 3.4 – Modelo 4GL

Os paradigmas abordados anteriormente podem ser combinados proporcionando um maior ganho de produtividade e qualidade no desenvolvimento do software, no entanto, é necessária que cada situação ou necessidade seja previamente discutida, para saber qual a melhor forma de paradigma a ser utilizado mediante a construção de um produto específico.

A Engenharia de Software integra métodos, ferramentas e procedimentos para que haja um maior poder de desenvolvimento dos softwares de computadores. Sua abordagem requer o uso de várias técnicas e recursos que quando usados fará com que a criação do produto – software - seja realizada da melhor forma possível.

## 4. MÉTRICAS DE SOFTWARE E PONTOS POR FUNÇÃO

Medir é algo importante e fundamental em qualquer assunto que envolva engenharia, pois quando é feita uma medição é possível ter uma visão mais realística e significativa sobre o tamanho ou abrangência de determinado assunto/produto. Na engenharia de Software medir é fundamental para fazer uma avaliação sobre esforço, custo, estimativas e qualidade, podendo assim, prever, antecipar e avançar em um projeto de uma forma mais aproximada e realística das conseqüências, positivas ou negativas, que são originárias do desenvolvimento do produto.

Certa vez Lorde Kelvin disse [PRE05]:

*“Quando se pode medir aquilo sobre o qual se está falando e expressa-lo em números, sabe-se alguma coisa sobre o mesmo; mas quando não se pode medi-lo, quando não se pode expressa-lo em números, o conhecimento que se tem é de um tipo inadequado e insatisfatório; este pode ser o começo do conhecimento, mas dificilmente alguém terá avançado em suas idéias para o estágio de ciência.”*

As palavras de Lorde Kelvin expressam a importância de encarar um determinado assunto de uma forma mais precisa e com maior exatidão, para assim ter conhecimento e domínio sobre o que está sendo abordado.

Métricas de software é um segmento da Engenharia de Software que permite medir o tamanho do software para que se tenha um maior domínio do processo de desenvolvimento do produto, tendo em vista uma forma de medir o esforço que foi despendido na elaboração do mesmo, podendo assim estabelecer uma linha de valor final devido à complexidade envolvida na criação de um software.

Para uma melhor explanação, é tomada como exemplo a construção de uma casa. Se você perguntar para um profissional desse ramo, quanto tempo ele gastará para fazer uma casa, com certeza a resposta será incerta, pois, você pode estar pensando em uma casa de 5 cômodos e ele em uma de 10 cômodos. No entanto se você quantificar a sua pergunta, ou seja, se você especificar que a casa terá 4 cômodos e fornecer o tamanho de cada cômodo, com certeza a pessoa que foi questionada irá prover uma resposta mais direcionada e precisa para a realidade, pois, dentro de seus conhecimentos ela quantificará grande parte dos fatores que estão envolvidos para a construção daquele tamanho de casa que foi especificado.

Isso não quer dizer que as métricas de software poderão falar quanto tempo se gastará para desenvolver um software com exatidão, ou quanto custará para fazer um software que tenha um tamanho X, até porque existem diversos fatores que possam influenciar no resultado final da elaboração, composição de um software, mas proporcionará o conhecimento dos esforços despendidos para a obtenção do produto e outros fatores tais como o conhecimento da qualidade do mesmo, a partir do estabelecimento de uma linha base para estimativas através de dados históricos obtidos pela funcionalidade. É bom reafirmar que, métricas de software por pontos de função, medem o tamanho da funcionalidade que é proporcionada pelo produto.

No mundo da engenharia muitas coisas são medidas para saber ao certo qual a proporção do tamanho, do volume, da quantidade sobre o que se está medindo. O consumo de energia, por exemplo, é medido para especificar

o valor a ser pago ou para fazer uma análise sobre a demanda dos consumidores. A temperatura é medida para saber se está muito frio ou frio, quente ou muito quente. A lista é interminável, no entanto, seja qual for a medição ou o que se está medindo, a finalidade é sempre a mesma: ter um conhecimento e um domínio sobre o que se tem em mãos.

Em termos de software, qual a razão de medi-lo? Pressman [PRE05] diz que o software é medido por muitas razões:

*“Indicar a qualidade do produto; avaliar a produtividade das pessoas que produzem o produto; avaliar os benefícios (em termos de produtividade e qualidade) derivados de novos métodos e ferramentas de software; formar uma linha básica para estimativas; ajudar a justificar os pedidos de novas ferramentas ou treinamento adicional”.*

Quando se está aplicando métricas de software, existem três possíveis tipos de medição a serem estabelecidos: métricas orientadas ao tamanho. Métricas orientadas à função e métricas de qualidade de software. Existe um quarto tipo de métrica que é voltado para casos de uso.

Métricas orientadas ao tamanho: determina o tamanho do software através da quantidade de linhas de código.

Métricas orientadas à pontos de função: determina o tamanho do software através de entradas e saídas de dados, ou seja, da funcionalidade proporcionada pelo software em relação ao usuário.

Métricas da qualidade de software: mede a qualidade do software a partir do momento que o mesmo é entregue ao usuário.

Existe discordâncias entre os defensores de métricas orientadas ao tamanho e métricas orientadas à pontos de função [PRE05]. Os proponentes de métricas orientadas ao tamanho afirmam que as LOC são de uma certa forma a melhor maneira de se fazer uma contagem e que existe mais recursos literários voltados para o assunto. Os opositores das métricas orientadas ao tamanho, concluem que as LOC são dependentes da linguagem de programação utilizada e que as mesmas prejudicam programas bem projetados, os quais reduzem a quantidade de linhas que serão construídas. Os proponentes das métricas orientadas a pontos de função acreditam que o FP independe da linguagem de programação sendo o ideal para linguagens não procedimentais e devido ao fato de se basear na funcionalidade dos dados coletados logo no começo da evolução do projeto, ela proporcionará uma melhor abordagem para estimativas. Já os opositores da técnica de contagem por pontos de função afirmam que tal técnica não tem nenhum significado físico direto – apenas um número.

Dentre as métricas listadas anteriormente, este trabalho aborda as métricas de pontos por função. No estudo de caso aplicado na empresa Seta Sistemas, usando um de seus softwares, é usado como medida a contagem de pontos por função para a apresentação de métricas de Software.

#### **4.1 – Métricas orientadas à pontos de função**

Para melhor entender sobre o assunto de métricas orientadas a pontos por função, é necessário, primeiramente, que se tenha um conhecimento sobre o que é tamanho funcional.

Tamanho funcional é uma medida de tamanho de software, baseada em uma avaliação padronizada dos requisitos lógicos dos usuários. Na indústria há atualmente várias maneiras para se medir tamanho funcional, a mais antiga das quais são os pontos de função. Semelhante aos metros quadrados de uma casa, pontos de função são independentes dos métodos

físicos, ferramentas ou linguagem de desenvolvimento utilizados para construir o software. Isso quer dizer que em um programa que faz a validação de um CPF, por exemplo, sua funcionalidade não é medida em termos técnicos da composição de linhas provenientes do código fonte que foi criado, mas sim a funcionalidade que será proporcionada para a satisfação do usuário, ou seja, a entrada e a saída da informação e não a forma como a informação está sendo processada internamente pelo código fonte do programa. Dentro de uma abordagem mais grotesca, tal funcionalidade está diretamente relacionada ao que o usuário possa ver, ou seja, as entradas e saídas que satisfazem as suas necessidades, sem se deter em características internas de programação.

Conforme foi citado anteriormente, essa característica de não avaliar as linhas de códigos levam os opositores desta técnica a manifestarem opiniões contrárias a sua efetivação.

É possível fazer uma combinação dessas abordagens para uma melhor aplicação e um resultado mais aproximado do que seja satisfatório, pois, a relação entre linhas de código e pontos de função depende da linguagem de programação que é usada para implementar o software e também pela qualidade existente no projeto. Atualmente existe uma série de estudos para tentar relacionar as medidas de pontos por função (PF) e as linhas de código (LOC). Podemos observar tal esforço através das palavras de Albrecht e Gaffney [PRE05]:

*“A tese deste trabalho é que a quantidade de funções a serem oferecidas pela aplicação (programas) pode ser estimada a partir da disposição em itens dos principais componentes de dados a serem usados ou oferecidos por ela. Além disso, essa estimativa de função deve estar correlacionada tanto com a quantidade de LOC a ser desenvolvida como com o esforço de desenvolvimento necessário.”*

A aplicação e o potencial das medidas funcionais estão sendo percebidos pela indústria de software como forma de administração do desenvolvimento de software.

Diferente do que muitas pessoas pensam, pontos de função não constituem o próprio programa de medida, pois, o mesmo sozinho sem ter outras medidas funcionais no que diz respeito ao tamanho, não fará com que um programa de medida possa acontecer.

Como medida de tamanho de software (semelhante a metros quadrados na construção civil), Pontos de Função apenas não são suficientes para compor um programa de medição de software. Pontos de Função medem o tamanho funcional do software. Da mesma forma que somente os metros quadrados são insuficientes para um construtor administrar a sua construção, apenas Pontos de Função são insuficientes para um desenvolvedor de software administrar um projeto. O tamanho funcional só é relevante quando utilizado em conjunto com outras medidas fundamentais, a fim de produzir outras métricas normalizadas. O gerenciamento do processo de software é possível com um conjunto coordenado de métricas de software apropriadas, algumas das quais podem estar baseadas em tamanho funcional.

Os pontos por função são semelhantes a medida em metros quadrados do tamanho de uma casa, o qual, não permite deduzir a velocidade com a qual a casa pode ser construída com exatidão, pois não se pode definir uma velocidade. Os pontos de função medem o tamanho do que o software faz e não como ele é desenvolvido e implementado.

#### **4.1.1 – Histórico sobre pontos por função**

O histórico apresentado abaixo foi retirado do site: <http://www.bfpug.com.br/Artigos/Dekkers-PontosDeFuncaoEMedidas.htm> sendo de propriedade intelectual e créditos de seu elaborador.

*“Os conceitos sobre Pontos de Função foram inicialmente introduzidos por Allan Albrecht da IBM, em uma conferência da Guide/Share em 1979. Posteriormente, esses conceitos foram refinados em uma metodologia formal e publicados no domínio público em 1984. Subseqüentemente, uma comunidade de ávidos usuários resolveu efetuar padronizações adicionais nas regras de contagem de Pontos de Função, sendo formado o Grupo Internacional de Usuários de Pontos de Função (IFPUG), como um grupo formalmente constituído e sem finalidades lucrativas, em 1986. Desde então o IFPUG tem sido líder no estabelecimento e publicação de documentos relacionados a Pontos de Função, incluindo o Manual de Práticas de Contagem, o Guia Para Medição de Software e diversos Estudos de Casos detalhados.*

*Atualmente, o IFPUG permanece como uma organização composta de voluntários participando ativamente do grupo de trabalho da ISO sobre Medidas Funcionais de Tamanho (ISO/IEC JTC1 SC7 WG12), administrando certificação de materiais para treinamento em Pontos de Função, Especialistas Certificados em Pontos de Função (CFPS), software sobre Pontos de Função, mantendo seu site na INTERNET, (<http://www.ifpug.org/ifpug>). Atualmente, membros de mais de vários países participam do IFPUG servindo em comitês, ou comparecendo a conferências e treinamentos.*

*A partir de um modesto início como um conceito original de Allan Albrecht da IBM em 1979 até a criação de um grupo de trabalho na ISO sobre Medidas Funcionais de Tamanho em 1994, as medidas funcionais*

*de tamanho (que incluem Pontos de Função) tornaram-se um dos principais assuntos da área. Hoje em dia, outros métodos estão sendo utilizados para a medição funcional de software. Os proponentes de cada método defendem a utilização de suas técnicas de medição para uma classe particular de software e freqüentemente para um tipo específico de utilização.”*

#### **4.1.2 – Processo de Contagem de Pontos de Função**

Para que possa fazer o processo de contagem de pontos de função, é necessário seguir alguns passos básicos os quais são:

Determinar o tipo de contagem (pode ser um projeto de novo desenvolvimento, uma contagem básica de aplicação ou uma contagem de projeto de melhoria)

- Identificar a fronteira da aplicação, ou seja, quais funções o software deve executar?
- Contar os tipos de funções de dados divididos em:
  - Arquivos Lógicos Internos ou ALIs, que são os grupos lógicos de dados mantidos dentro da fronteira da aplicação;
  - Arquivos de Interface Externa ou AIEs, os quais são apenas referenciados pela aplicação.
- Cada ALI vale 7, 10 ou 15 PF, enquanto cada AIE vale 5, 7 ou 10 PF.
- Contar os tipos de funções de transações divididos em:
  - Entradas Externas ou EEs, que são processos de entrada de dados,
  - Saídas Externas ou SEs, por exemplo, relatórios e
  - Consultas Externas ou CEs, por exemplo, *Consultar Detalhes de Empregados*).

- Cada EE ou CE vale 3, 4 ou 6 pontos de função, enquanto cada SE vale 4, 5 ou 7 pontos de função.

Diversas matrizes simples baseadas nos tipos de elementos de dados, juntamente com tipos de registros ou tipos de arquivos referenciados são utilizados para determinar a complexidade de cada função, Baixa, Média ou Alta. A seguinte tabela do IFPUG sintetiza o número de pontos de função atribuídos a cada tipo de função:

<b>Tipo de Função</b>	<b>Baixa</b>	<b>Média</b>	<b>Alta</b>
<b>EE</b>	<b>3</b>	<b>4</b>	<b>6</b>
<b>SE</b>	<b>4</b>	<b>5</b>	<b>7</b>
<b>CE</b>	<b>3</b>	<b>4</b>	<b>6</b>
<b>ALI</b>	<b>7</b>	<b>10</b>	<b>15</b>
<b>AIE</b>	<b>5</b>	<b>7</b>	<b>10</b>

Tabela 4.01 – Tabela de complexidade

Determinar o Fator de Ajuste de Valor (FAV) baseado na equação:  $(FAV = 0,65 + (Soma\ das\ Características\ Gerais\ do\ Sistema \times 0,01))$  e a avaliação, em uma escala de 1 a 5, das seguintes quatorze Características Gerais do Sistema. Instruções específicas para avaliação são fornecidas no CPM do IFPUG:

1. Comunicação de Dados
2. Processamento Distribuído de Dados
3. Desempenho
4. Configuração Intensamente Utilizada
5. Taxa de Transação
6. Entrada de Dados On-Line
7. Eficiência do Usuário Final

8. Atualização On-Line
9. Processamento Complexo
10. Reutilização
11. Facilidade de Instalação
12. Facilidade de Operação
13. Múltiplas Localidades
14. Facilidade de Alteração

Calcular a contagem ajustada final de PF (contagem final de PF = contagem não ajustada \* FAV)

#### **4.1.3 – Descrição de como avaliar os critérios de pontuação.**

1. **Comunicação de dados:** os aspectos relacionados aos recursos utilizados para a comunicação de dados do sistema deverão ser descritos de forma global. Descrever se a aplicação utiliza protocolos diferentes para recebimento/envio das informações do sistema.

0. Aplicação *batch* ou funciona *stand-alone*;
1. Aplicação *batch*, mas utiliza entrada de dados ou impressão remota;
2. Aplicação *batch*, mas utiliza entrada de dados e impressão remota;
3. Aplicação com entrada de dados *on-line* para alimentar processamento *batch* ou sistema de consulta;
4. Aplicação com entrada de dados *on-line*, mas suporta apenas um tipo de protocolo de comunicação;
5. Aplicação com entrada de dados *on-line* e suporta mais de um tipo de protocolo de comunicação.

2. **Processamento de Dados Distribuído:** Esta característica refere-se a sistemas que utilizam dados ou processamento distribuído, valendo-se de diversas CPUs.

0. Aplicação não auxilia na transferência de dados ou funções entre os processadores da empresa;
1. Aplicação prepara dados para o usuário final utilizar em outro processador (do usuário final), tal como planilhas;
2. Aplicação prepara dados para transferência, transfere-os para serem processados em outro equipamento da empresa (não pelo usuário final);
3. Processamento é distribuído e a transferência de dados é *on-line* e apenas em uma direção;
4. Processamento é distribuído e a transferência de dados é *on-line* e em ambas as direções;
5. As funções de processamento são dinamicamente executadas no equipamento (CPU) mais apropriado;

3. **Desempenho:** Trata-se de parâmetros estabelecidos pelo usuário como aceitáveis, relativos a tempo de resposta.

0. Nenhum requisito especial de desempenho foi solicitado pelo usuário;
1. Requisitos de desempenho foram estabelecidos e revistos, mas nenhuma ação especial foi requerida;
2. Tempo de resposta e volume de processamento são itens críticos durante horários de pico de processamento. Nenhuma determinação especial para a utilização do processador foi estabelecida. A data limite para a disponibilidade de processamento é sempre o próximo dia útil;

3. Tempo de resposta e volume de processamento são itens críticos durante todo o horário comercial. Nenhuma determinação especial para a utilização do processador foi estabelecida. A data-limite necessária para a comunicação com outros sistemas é limitante.
4. Os requisitos de desempenho estabelecidos requerem tarefas de análise de desempenho na fase de planejamento e análise da aplicação.
5. Além do descrito no item anterior, ferramentas de análise de desempenho foram usadas nas fases de planejamento, desenvolvimento e/ou implementação para atingir os requisitos de desempenho estabelecidos pelos usuários.

**4. Utilização do Equipamento:** Trata-se de observações quanto ao nível de utilização de equipamentos requerido para a execução do sistema. Este aspecto é observado com vista a planejamento de capacidades e custos.

0. Nenhuma restrição operacional explícita ou mesmo implícita foi incluída.
1. Existem restrições operacionais leves. Não é necessário esforço especial para atender às restrições.
2. Algumas considerações de ajuste de desempenho e segurança são necessárias.
3. São necessárias especificações especiais de processador para um módulo específico da aplicação.
4. Restrições operacionais requerem cuidados especiais no processador central ou no processador dedicado para executar a aplicação.
5. Além das características do item anterior, há considerações especiais que exigem utilização de ferramentas de análise de

desempenho, para a distribuição do sistema e seus componentes, nas unidades processadoras.

5. **Volume de transações:** Consiste na avaliação do nível de influência do volume de transações no projeto, desenvolvimento, implantação e manutenção do sistema.

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

6. **Entrada de dados *on-line*:** A análise desta característica permite quantificar o nível de influência exercida pela utilização de entrada de dados no modo *on-line* no sistema.

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

7. **Usabilidade:** a análise desta característica permite quantificar o grau de influência relativo aos recursos implementados com vista a tornar o

sistema amigável, permitindo incrementos na eficiência e satisfação do usuário final, tais como:

- Auxílio à navegação (teclas de função, acesso direto e menus dinâmicos)
- Menus Documentação e *help on-line*
- Movimento automático do cursor.
- Movimento horizontal e vertical de tela.
- Impressão remota (via transações *on-line*)
- Teclas de função preestabelecidas.
- Processos batch submetidos a partir de transações *on-line*
- Utilização intensa de campos com vídeo reverso, intensificados, sublinhados, coloridos e outros indicadores.
- Impressão da documentação das transações on-line através de *hard copy*
- Utilização de mouse
- Menus *pop-up*
- O menor número possível de telas para executar as funções de negócio.
- Suporte bilingüe (contar como 4 itens)
- Suporte multilíngüe. (contar como 6 itens)

**Pontuação:**

0. Nenhum dos itens descritos.
1. De um a três itens descritos.
2. De quatro a cinco dos itens descritos.
3. Mais de cinco dos itens descritos, mas não há requisitos específicos do usuário quanto a usabilidade do sistema.
4. Mais de cinco dos itens descritos e foram estabelecidos requisitos quanto à usabilidade fortes o suficiente para gerarem atividades específicas envolvendo fatores, tais como minimização da

digitação, para mostrar inicialmente os valores utilizados com mais frequência.

5. Mais de cinco dos itens descritos e foram estabelecidos requisitos quanto à usabilidade fortes o suficiente para requerer ferramentas e processos especiais para demonstrar antecipadamente que os objetivos foram alcançados.

8. **Atualizações *on-line*:** Mede a influência no desenvolvimento do sistema face à utilização de recursos que visem a atualização dos Arquivos Lógicos Internos, no modo *online*.

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

9. **Processamento complexo:** a complexidade de processamento influencia no dimensionamento do sistema, e, portanto, deve ser quantificado o seu grau de influência, com base nas seguintes categorias:

- Processamento especial de auditoria e/ou processamento especial de segurança foram considerados na aplicação;
- Processamento lógico extensivo;
- Processamento matemático extensivo;
- Processamento gerando muitas exceções, resultando em transações incompletas que devem ser processadas novamente. Exemplo: transações de auto-atendimento bancário interrompidas por problemas de comunicação ou com dados incompletos;
- Processamento complexo para manusear múltiplas possibilidades de entrada/saída. Exemplo: multimídia.

## Pontuação

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

10. **Reusabilidade:** a preocupação com o reaproveitamento de parte dos programas de uma aplicação em outras aplicações implica em cuidados com padronização. O grau de influência no dimensionamento do sistema é quantificado observando-se os seguintes aspectos:

0. Nenhuma preocupação com reutilização de código.
1. Código reutilizado foi usado somente dentro da aplicação.
2. Menos de 10% da aplicação foi projetada prevendo utilização posterior do código por outra aplicação.
3. 10% ou mais da aplicação foi projetada prevendo utilização posterior do código por outra aplicação.

4. A aplicação foi especificamente projetada e/ou documentada para ter seu código reutilizado por outra aplicação e a aplicação é customizada pelo usuário em nível de código -fonte.
5. A aplicação foi especificamente projetada e/ou documentada para ter seu código facilmente reutilizado por outra aplicação e a aplicação é customizada para uso através de parâmetros que podem ser alterados pelo usuário.

11. **Facilidade de implantação:** a quantificação do grau de influência desta característica é feita, observando-se o plano de conversão e implantação e/ou ferramentas utilizadas durante a fase de testes do sistema.

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

12. **Facilidade operacional:** a análise desta característica permite quantificar o nível de influência na aplicação, com relação a procedimentos

operacionais automáticos que reduzem os procedimentos manuais, bem como mecanismos de inicialização, salvamento e recuperação, verificados durante os testes do sistema.

0. Nenhuma consideração especial de operação, além do processo normal de salvamento foi estabelecida pelo usuário.

1-4. Verifique quais das seguintes afirmativas podem ser identificadas na aplicação. Selecione as que forem aplicadas. Cada item vale um ponto, exceto se definido explicitamente:

- Foram desenvolvidos processos de inicialização, salvamento e recuperação, mas a intervenção do operador é necessária.
- Foram estabelecidos processos de inicialização, salvamento e recuperação, e nenhuma intervenção do operador é necessária (conte como dois itens)
- A aplicação minimiza a necessidade de montar fitas magnéticas.
- A aplicação minimiza a necessidade de manuseio de papel.

5. A aplicação foi desenhada para trabalhar sem operador, nenhuma intervenção do operador é necessária para operar o sistema além de executar e encerrar a aplicação. A aplicação possui rotinas automáticas para recuperação em caso de erro.

**13. Múltiplos Locais e Organizações do Usuário:** consiste na análise da arquitetura do projeto, observando-se a necessidade de instalação do sistema em diversos lugares.

0. Os requisitos do usuário não consideraram a necessidade de instalação em mais de um local.

1. A necessidade de múltiplos locais foi considerada no projeto e a aplicação foi desenhada para operar apenas em ambientes de software e hardware idênticos.
2. A necessidade de múltiplos locais foi considerada no projeto e a aplicação está preparada para trabalhar apenas em ambientes similares de software e hardware.
3. A necessidade de múltiplos locais foi considerada no projeto e a aplicação está preparada para trabalhar em diferentes ambientes de hardware e/ou software.
4. Plano de documentação e manutenção foram providos e testados para suportar a aplicação em múltiplos locais, além disso, os itens 1 ou 2 caracterizam a aplicação.
5. Plano de documentação e manutenção foram providos e testados para suportar a aplicação em múltiplos locais, além disso, o item 3 caracteriza a aplicação.

14. **Facilidade de mudanças:** focaliza a preocupação com a influencia da manutenção no desenvolvimento do sistema. Esta influência deve ser quantificada baseando na observação de atributos, tais como:

- Disponibilidade de facilidades como consultas e relatórios flexíveis para atender necessidades simples (conte como 1 item);
- Disponibilidade de facilidades como consultas e relatórios flexíveis para atender necessidades de complexidade média (conte como 2 itens);
- Disponibilidade de facilidades como consultas e relatórios flexíveis para atender necessidades complexas (conte 3 itens);
- Se os dados de controle são armazenados em tabelas que são mantidas pelo usuário através de processos on-line, mas mudanças têm efeitos somente no dia seguinte;

- Se os dados de controle são armazenados em tabelas que são mantidas pelo usuário através de processos on-line, as mudanças têm efeito imediatamente (conte como 2 itens).

**Pontuação:**

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

O conteúdo apresentado acima, a partir de “processo de contagem de pontos por função”, é uma caracterização técnica inerente a abordagem de métricas de software por pontos de função que veio evoluindo ao longo dos anos e que é proveniente de muitas fontes de pesquisas. O material usado como referência na elaboração do conteúdo anteriormente abordado, foi compilado de:

Site: <http://www.bfpug.com.br/Artigos/Dekkers-PontosDeFuncaoEMedidas.htm>

Pesquisado em 26/03/2006 às 06h30.

Site: <http://www.inf.ufes.br/~falbo/download/aulas/es-g/2005-1/APF.pdf>

Pesquisado em 26/03/2006 às 06h35.

## **5. ESTUDO DE CASO**

Para o estudo de caso proposto foi disponibilizado um software de gestão de leite, o qual faz todo gerenciamento em relação à transação comercial e financeira, manutenção e transporte do produto em questão.

A finalidade deste estudo de caso, conforme descrito na proposta deste documento, é de se fazer à contagem parcial deste sistema a fim de se compreender de forma prática a aplicação de métricas de software em um campo real.

O Software usado para fazer a simulação da contagem de pontos de função é comercializado e por motivos firmados junto à empresa, onde houve um compromisso entre os integrantes deste grupo e o proprietário da empresa, só foram disponibilizadas telas e informações do sistema previamente autorizadas pela mesma, a fim de demonstrarmos as evidências daquilo que está sendo objetivado nesse trabalho. Para isso serão observadas as formas de se fazerem as medidas das entidades do banco de dados, em seguida serão mostradas algumas interfaces exemplificando cada uma das funções transacionais, em seguida aos resultados obtidos no estudo de casos e por fim será feita uma simulação de ajuste final dos pontos de função contados neste sistema.

### **5.1 – Contagem de função de dados**

Para se fazer a contagem de funções de dados devem ser observadas as seguintes características:

- A entidades e atributos multivalorados são consideradas Registros Lógicos (RL), ou seja, um subgrupo de dados reconhecido pelo usuário o que significa que apenas são contados dados que são conhecidos pelo usuário ou que interagem com o mesmo.

- Os atributos da tabela são considerados Itens de Dados (ID), que são campos de dados reconhecidos pelo o usuário, ou seja, são os itens que compõe os registros lógicos. Chaves estrangeiras não devem ser contadas.

Exemplificando o essa situação seja observada a seguinte entidade do sistema:

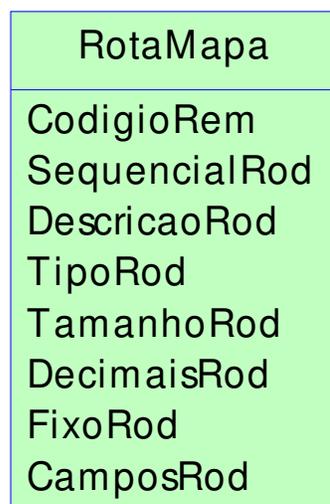


Figura 5.01 – Entidade RotaMapa

A entidade sendo um subgrupo de dados é considerada um Registro Lógico, como neste caso não há atributos multivalorados então será contado apenas um RL, a saber RotaMapa.

Os atributos (campos) que compõe este subgrupo RotaMapa, são considerados Itens de Dados, e na ausência de chaves estrangeiras deve-se contar todos os campos pertencentes a essa entidade, teremos então 8 ID's.

## 5.2 – Contagem de funções transacionais

Para se fazer a contagem das funções transacionais serão observadas três características transacionais: Entradas Externas(EE), Consultas Externas(CE) e Saídas Externas(SE), em cada uma dessas características quantificar o número de Arquivos Referenciados – número e arquivos referenciados lidos ou mantidos durante o processamento, podem ser tanto ALI quanto AIE – e o número de itens de dados – campos, ações e mensagens que envolvam a função transacional.

Entradas Externas [UFE05] é o processo elementar que faz o processamento dos dados que entram pela fronteira da aplicação e seu principal objetivo é manter um ou mais Arquivos Lógicos Internos ou alterar o comportamento do sistema. Geralmente fazemos a contagem de todos os campos da interface envolvidos com esse tipo de transação, o evento ou o componente que o acionou, e as possíveis mensagens que possam ser emitidas pelo sistema nessa transação, sendo que independente da quantidade são contados apenas um Ponto de Função para todas elas. Para exemplificar, considere o seguinte formulário do sistema Gestão de Leite:

A imagem mostra uma janela de software intitulada "Cadastro de município". No topo, há uma barra de menu com "Cadastro" e "Visualização". O formulário principal contém os seguintes campos:

- Código:
- Descrição:
- Código no estado:
- UF:

Na base da janela, há uma barra de ferramentas com os seguintes botões: "Incluir", "Gravar", "Excluir", "Cancelar", "Procurar", "Configurar" e "Sair". Abaixo da barra de ferramentas, o status "Procurando" é exibido.

Figura 5.02 – Tela Cadastro de município

Neste caso acima para Entradas Externas deve-se contar apenas um Arquivo Referenciado, pois apenas um ALI é lido ou mantido pelo sistema – entidade Município. Os itens de dados seriam os campos reconhecidos pelo usuário neste caso quatro, o botão que acionou este tipo de função transacional contaria mais um ID e as possíveis mensagens emitidas pela transação, todas elas somariam mais um ID para a interface da figura 4.2. As Entradas Externas relacionadas a esta interface seriam contados separadamente para a ação incluir e para a ação editar deste formulário.

Consultas Externas (CE) segundo [UFE05] é um processo elementar que envia dados (ou informações de controle) para fora da fronteira da aplicação, mas sem realização de nenhum cálculo nem a criação de dados derivados. Seu objetivo é apresentar informação para o usuário, por meio apenas de uma recuperação das informações. Normalmente na medição desse tipo de transação são considerados as entidades a serem lidas (Arquivos Referenciados) os campos que serão recuperados na consulta (Itens de Dados), os campos que serviram como filtros ou parâmetros de pesquisa, a ação que emitiu o comando de recuperação, um botão por exemplo, e as possíveis mensagens emitidas nessa função transacional. Considere a seguinte interface como um sendo acionada pelo botão Procurar da interface Cadastro de Município (Figura 5.2):

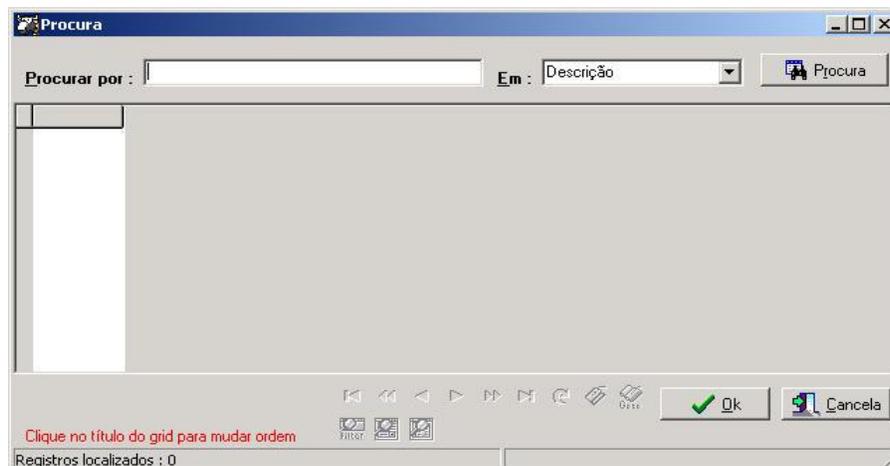


Figura 5.03 – Tela Procura

Neste caso deverá ser contado um arquivo referenciado, uma vez que os dados serão recuperados de uma só entidade (Município), e como Itens de Dados os campos recuperados por essa pesquisa, a saber, todos os campos da tela Cadastro de Município, o campo referenciado por “Pesquisa por”, todos os campos contidos na opção referenciada por “Em:”, um ID para o conjunto de botões que participaram desta transação (Procurar – Tela Cadastro de município, Procura e OK da tela Procura), e um ID para as mensagens que fazem parte desta função transacional.

Por fim Saídas Externas (SE) que assim como as CE's de acordo com [UFE05] são processos elementares que enviam dados (ou informações de controle) para fora da fronteira da aplicação porém seu objetivo é mostrar informações recuperadas através de um processamento lógico (isto é, que envolva cálculos ou criação de dados derivados) e não apenas uma simples recuperação de dados. A forma de se contar uma função transacional do tipo Saída Externa segue o mesmo padrão em relação a Arquivos Referenciados, deve-se contar todos os AR's envolvidos na função, seus Itens de Dados são os filtros, ou parâmetros de pesquisa, cada item do filtro é considerado um ID, contamos o acionamento do botão que disparou o processo desta função transacional, os campos recuperados, derivados e calculados que foram trazidos neste processo. Sejam como exemplo as seguintes interfaces:

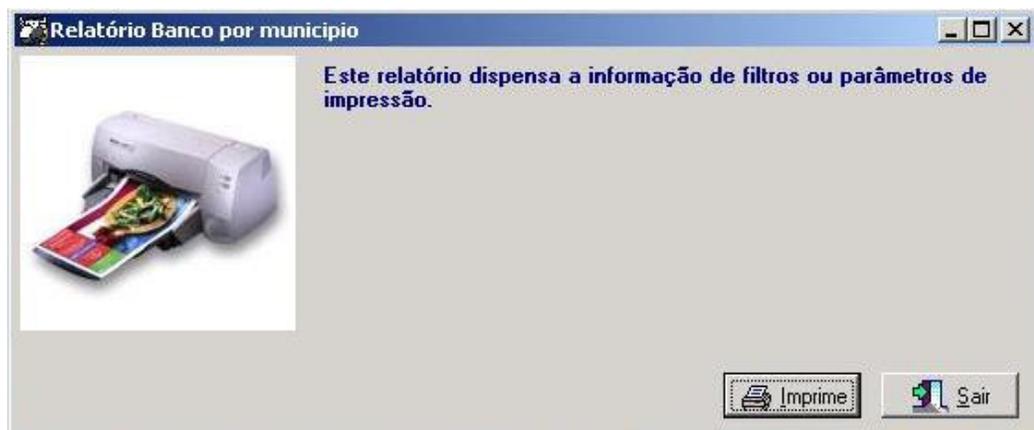


Figura 5.04 - Gerar relatório

Visualizando Impressão

100% Fechar

LATICINIOS MORRINHOS

### Banco por municipio

Banco	Nº	Agência	Fone	Fax
<b>Município : Agro Vila Bradesco-PA</b>				
Banco do Brasil - Agro Vila Bradesco-PA	001	0914-8		
<b>Município : Anápolis-GO</b>				
Banco do Brasil - Anápolis	001	0324-7		
Banco do Brasil - Anápolis-Go	001	3005-8		
Bradesco	237	0240-2		
Itaú - Anápolis	341	4393-0		
Unibanco	409	0096-5		
<b>Município : Anicuns-GO</b>				
Banco do Brasil - Anicuns-Go	001	0557-6		
Banco Do Brasil	001	5557-6		
Bradesco - Anicuns-Go	237	1944-5		
Itaú - Anicuns-Go	341	4315-0		
<b>Município : Aparecida de Goiânia-GO</b>				
Banco do Brasil	001	1452-4		
Bradesco	237	2711-1		
<b>Município : Aragarças-GO</b>				
Itaú - Aragarças-Mt	341	4391-0		
Hsbc -	399	0495-0		
<b>Município : Aurilândia-GO</b>				
Itaú - Aurilândia	341	5099-0		
<b>Município : Barra do Garças-MT</b>				
Banco do Brasil - Barra do Garças-Mt	001	0571-1		
Caixa E. Federal - Barra do Garças	104	1308-0		
Bradesco - Barra do Garças-Mt	237	3292-1		
<b>Município : Barretos-SP</b>				
Banco do Brasil - Barretos	001	0031-0		
Bradesco	237	0144-9		
Bradesco	237	1382-0		

Página 1 de 9

Figura 5.05 – Relatório Gerado

Neste caso será observado que a função transacional de Saída Externa possui dois Arquivos Referenciados, novamente a Entidade município e Banco, contaremos como Itens de Dados na figura 5.4 apenas o acionamento do botão Imprime que irá gerar uma pré-visualização da relação de Bancos por Municípios (Figura 5.5), nessa listagem poderá se observar seis campos resultantes dessa função transacional e que serão contados como Itens de Dados, outro ID será atribuído à alguma mensagem que possa ser emitida nesse processo, não há campos derivados ou calculados.

Depois de contados todas as funções de dados e transacionais temos uma classificação entre Simples, Media e Complexa que deverão ser atribuídas à contagem conforme as tabelas referenciadas no capítulo 3 mediante os atributos contados dentro de suas classificações.

### **5.3 - Contagem parcial do Sistema**

Depois de comentado como são tratados os aspectos comportamentais que envolvem Funções de Dados e Funções Transacionais de um sistema, abordando de forma prática a maneira de como proceder na mensuração envolvendo às mesmas e disponibilizando para isso algumas interfaces do sistema, passaremos aos resultados do estudo de caso onde foram aplicados todos os procedimentos citados neste capítulo.

Seguem se então as tabelas com os valores referentes as contagem de função de dados e função transacional, respectivamente.

### 5.3.1 - Contagem de função de Dados

<b>Entidade – AnaliseQua(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	20	Simple

Tabela 5.01 – Entidade AnaliseQua

<b>Entidade - Banco(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	09	Simple

Tabela 5.02 – Entidade Banco

<b>Entidade - BonificacaoFaz(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	08	Simple

Tabela 5.03 – Entidade BonificacaoFaz

<b>Entidade - BonificacaoMun(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	13	Simple

Tabela 5.04 – Entidade BonificacaoMun

<b>Entidade - BonificacaoReg(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	25	Simple

Tabela 5.05 – Entidade BonificacaoReg

<b>Entidade - BTLC(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	49	Simple

Tabela 5.06 – Entidade BTLC

<b>Entidade – BTLCMov(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	20	Simple

Tabela 5.07 – Entidade BTLCMov

<b>Entidade – BTLCPrep(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	13	Simple

Tabela 5.08 – Entidade BTLCPrep

<b>Entidade - BTLCPrepFazenda(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	05	Simple

Tabela 5.09 – Entidade BTLCPrepFazenda

<b>Entidade - CompradorReg(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	05	Simple

Tabela 5.10 – Entidade CompradorReg

<b>Entidade - Convenio(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	14	Simple

Tabela 5.11 – Entidade Convenio

<b>Entidade - ConvenioDesfaz(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	07	Simple

Tabela 5.12 – Entidade ConvenioDesfaz

<b>Entidade - ConvenioFaz(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	19	Simple

Tabela 5.13 – Entidade ConvenioFaz

<b>Entidade – ConvenioFaz2(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	12	Simple

Tabela 5.14 – Entidade ConvenioFaz2

<b>Entidade - ConvenioFazPar(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	10	Simple

Tabela 5.15 – Entidade ConvenioFazPar

<b>Entidade - ConvenioVei(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	19	Simple

Tabela 5.16 – Entidade ConvenioVei

<b>Entidade - ConvenioVeiPar(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	10	Simple

Tabela 5.17 – Entidade ConvenioVeiPar

<b>Entidade – Cota(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	16	Simple

Tabela 5.18 – Entidade Cota

<b>Entidade – Crioscopia(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	02	Simple

Tabela 5.19 – Entidade Crioscopia

<b>Entidade CustoLeiteFaz(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	16	Simple

Tabela 5.20 – Entidade CustoLeiteFaz

<b>Entidade – DefBancaria(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	36	Simple

Tabela 5.21 – Entidade DefBancaria

<b>Entidade – DigitoVerificador(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	20	Simple

Tabela 5.22 – Entidade DigitoVerificador

<b>Entidade – Evento(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	16	Simple

Tabela 5.23 – Entidade Evento

<b>Entidade – EventoPar(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	03	Simple

Tabela 5.24 – Entidade EventoPar

<b>Entidade – EventoVerba(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	05	Simple

Tabela 5.25 – Entidade EventoVerba

<b>Entidade – FaixaQual(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	02	Simple

Tabela 5.26 – Entidade FaixaQual

<b>Entidade – Fazenda(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	20	Simple

Tabela 5.27 – Entidade Fazenda

<b>Entidade – FazendaFre(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	07	Simple

Tabela 5.28 – Entidade FazendaFre

<b>Entidade – FazendaPer(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	25	Simple

Tabela 5.29 – EntidadeFazendaPer

<b>Entidade – FazendaPer2(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	23	Simple

Tabela 5.30 – Entidade FazendaPer

<b>Entidade – FazendaTan(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	09	Simple

Tabela 5.31 – Entidade FazendaTan

<b>Entidade – FornecedorLeite(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	11	Simple

Tabela 5.32 – Entidade FornecedorLeite

<b>Entidade – Funcionario(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	08	Simple

Tabela 5.33 – Entidade Funcionario

<b>Entidade – IndicadorFaz1(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	14	Simple

Tabela 5.34 – Entidade IndicadorFaz

<b>Entidade – Laboratorio(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	13	Simple

Tabela 5.35 – Entidade laboratorio

<b>Entidade – LimiteLit(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	04	Simple

Tabela 5.36 – Entidade LimiteLit

<b>Entidade – ListaLeite(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	65	Média

Tabela 5.37 – Entidade ListaLeite

<b>Entidade – Litragem(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	29	Simple

Tabela 5.38 – Entidade Litragem

<b>Entidade – LitragemExc(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	10	Simple

Tabela 5.39 – Entidade LitragemExc

<b>Entidade – LoCodigoBarras(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	10	Simple

Tabela 5.40 – Entidade LoCodigoBarras

<b>Entidade – LoRemessa (A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	17	Simple

Tabela 5.41 – Entidade LoRemessa

<b>Entidade - LoRemessaCab(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	08	Simple

Tabela 5.42 – Entidade LoRemessaCab

<b>Entidade - LoRemessaDet(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	08	Simple

Tabela 5.43 – Entidade LoRemessaDet

<b>Entidade - LoRemessaGru(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	08	Simple

Tabela 5.44 – Entidade LoRemessaGru

<b>Entidade - LoRemessaMsgCar(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	08	Simple

Tabela 5.45 – Entidade LoRemessaMsgCar

<b>Entidade - LoRemessaMsgEspCar(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	08	Simple

Tabela 5.46 – Entidade LoRemessaMsgEspCar

<b>Entidade - LoRemessaRod(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	08	Simple

Tabela 5.47 – Entidade LoRemessaRod

<b>Entidade - LoRemessaRodGru(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	08	Simple

Tabela 5.48 – Entidade LoRemessaRodGru

<b>Entidade - LoRemessaSubDet(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	08	Simple

Tabela 5.49 – Entidade LoRemessaSubDet

<b>Entidade - LoRetorno(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	35	Simple

Tabela 5.50 – Entidade LoRetorno

<b>Entidade – MarcaTanque(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	02	Simple

Tabela 5.51 – Entidade MarcaTanque

<b>Entidade – Memorando(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	14	Simple

Tabela 5.52 – Entidade Memorando

<b>Entidade - MemorandoFaz(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	15	Simple

Tabela 5.53 – Entidade MemorandoFaz

<b>Entidade - MemorandoRot(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	15	Simple

Tabela 5.54 – Entidade MemorandoRot

<b>Entidade - MemorandoUni(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	15	Simple

Tabela 5.55 – Entidade MemorandoUni

<b>Entidade - MemorandoVei(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	16	Simple

Tabela 5.56 – Entidade MemorandoVei

<b>Entidade - MicroRegiao(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	05	Simple

Tabela 5.57 – Entidade MicroRegiao

<b>Entidade - MotivoMem(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	06	Simple

Tabela 5.58 – Entidade MotivoMem

<b>Entidade – Municipio(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	04	Simple

Tabela 5.59 – Municipio

<b>Entidade – OcorrenciasConV(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	04	Simple

Tabela 5.60 – Entidade OcorrenciaConV

<b>Entidade – PagamentoFazTot(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	22	Simple

Tabela 5.61 – Entidade PagamentoFazTot

<b>Entidade – PagamentoFazTot2(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	17	Simple

Tabela 5.62 – Entidade PagamentoFazTot

<b>Entidade – PagamentoVei(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	07	Simple

Tabela 5.63 – Entidade PagamentoVei

<b>Entidade – PagamentoVeiTot(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	11	Simple

Tabela 5.64 – Entidade PagamentoVeiTot

<b>Entidade - Parametro(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	12	Simple

Tabela 5.65 – Entidade Parametro

<b>Entidade - ParametroAdc(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	05	Simple

Tabela 5.66 – Entidade ParametroAdc

<b>Entidade - ParametroCBT(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	06	Simple

Tabela 5.67 – Entidade ParametroCBT

<b>Entidade - ParametroCCS(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	06	Simple

Tabela 5.68 – Entidade ParametroCCS

<b>Entidade - ParametroGor(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	06	Simple

Tabela 5.69 – Entidade ParametroGor

<b>Entidade - ParametroPro(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	06	Simple

Tabela 5.70 – Entidade ParametroPro

<b>Entidade - ParametrosSis(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	17	Simple

Tabela 5.71 – Entidade ParametrosSis

<b>Entidade – Percurso(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	03	Simple

Tabela 5.72 – Entidade Percurso

<b>Entidade – Periodo(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	04	Simple

Tabela 5.73 – Entidade Periodo

<b>Entidade – PeriodoPar(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	06	Simple

Tabela 5.74 – Entidade PeriodoPar

<b>Entidade Ponto(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	02	Simple

Tabela 5.75 – Entidade Ponto

<b>Entidade – Preço(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	23	Simple

Tabela 5.76 – Entidade Preço

<b>Entidade – Preço2(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	23	Simple

Tabela 5.77 – Entidade Preço2

<b>Entidade – Preço3(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	23	Simple

Tabela 5.78 – Entidade Preço3

<b>Entidade – PreparaDesConF(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	12	Simple

Tabela 5.79 – Entidade PreparaDesConF

<b>Entidade – PreparaDesConV(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	10	Simple

Tabela 5.80 – Entidade PreparaDesConV

<b>Entidade – Produtor(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	20	Simple

Tabela 5.81 – Entidade Produtor

<b>Entidade – QualidadeRec(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	08	Simple

Tabela 5.82 – Entidade QualidadeRec

<b>Entidade – RecepcãoBan(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	13	Simple

Tabela 5.83 – Entidade RecepcãoBan

<b>Entidade - Região(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	03	Simple

Tabela 5.84 – Entidade Região

<b>Entidade – Relatorio(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	07	Simple

Tabela 5.85 – Entidade Relatorio

<b>Entidade – Rota(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	07	Simple

Tabela 5.86 – Entidade Rota

<b>Entidade – Rota Mapa(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	12	Simple

Tabela 5.87 – Entidade RotaMapa

<b>Entidade – RotaPer(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	12	Simple

Tabela 5.88 – Entidade RotaPer

<b>Entidade – SituacaoPagFaz(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	05	Simple

Tabela 5.89 – Entidade SituacaoPagFaz

<b>Entidade – SituacaoPagVei(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	05	Simple

Tabela 5.90 – Entidade SituacaoPagVei

<b>Entidade – SupervisorReg(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	05	Simple

Tabela 5.91 – Entidade SupervisorReg

<b>Entidade – TabelaQua(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	21	Simple

Tabela 5.92 – Entidade TabelaQua

<b>Entidade – TicketFazenda(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	03	Simple

Tabela 5.93 – Entidade TicketFazenda

<b>Entidade – Transportador(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	16	Simple

Tabela 5.94 – Entidade Transportador

<b>Entidade – Unidade(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	15	Simple

Tabela 5.95 – Entidade Unidade

<b>Entidade - Usuario(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	11	Simple

Tabela 5.96 - – Entidade Usuario

<b>Entidade – UsuarioAcesso(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	08	Simple

Tabela 5.97 – Entidade UsuarioAcesso

<b>Entidade – UsuarioSistema(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	04	Simple

Tabela 5.98 – Entidade UsuarioSistema

<b>Entidade – UsuarioUnidade(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	03	Simple

Tabela 5.99 – Entidade UsuarioUnidade

<b>Entidade – Veiculo(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	11	Simple

Tabela 5.100 – Entidade Veiculo

<b>Entidade – VeiculoPer(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	12	Simple

Tabela 5.101 – Entidade VeiculoPer

<b>Entidade – VeiculoPer2(A.L.I.)</b>		
<b>Relacionamentos Lógicos</b>	<b>Itens de Dados</b>	<b>Complexidade</b>
1	10	Simple

Tabela 5.102 – Entidade VeiculoPer

### 5.3.2 - Contagem de funções transacionais

<b>Tela - Cadastro de Rota</b>	
<b>Tipo de função transacional</b>	<b>Complexidade</b>
Incluir (EE)	<b>Complexa</b>
Alterar (EE)	<b>Complexa</b>
Excluir (EE)	<b>Simple</b>
Procurar (CE)	<b>Complexa</b>
Configurar (EE)	<b>Complexa</b>

Tabela 5.103 – Tela Cadastro Rota

<b>Tela - Funcionário</b>	
<b>Tipo de função transacional</b>	<b>Complexidade</b>
Incluir (EE)	<b>Complexa</b>
Alterar (EE)	<b>Complexa</b>
Excluir (EE)	<b>Simple</b>
Procurar (CE)	<b>Complexa</b>
Configurar (EE)	<b>Complexa</b>

Tabela 5.104 – Tela Funcionario

<b>Tela - Cadastro de Município</b>	
<b>Tipo de função transacional</b>	<b>Complexidade</b>
Incluir (EE)	<b>Simple</b>
Alterar (EE)	<b>Simple</b>
Excluir (EE)	<b>Simple</b>
Procurar (CE)	<b>Simple</b>
Visualizar (CE)	<b>Simple</b>
Configurar (EE)	<b>Complexa</b>

Tabela 5.105 – Tela Cadastro de Municipio

<b>Tela - Parâmetros</b>	
<b>Tipo de função transacional</b>	<b>Complexidade</b>
Incluir (EE)	Média
Alterar (EE)	Média
Excluir (EE)	Simple
Procurar (CE)	Média
Visualizar (CE)	Média
Configurar (EE)	Complexa

Tabela 5.106 – Tela Parâmetros

<b>Tela - Cadastro de Região</b>	
<b>Tipo de função transacional</b>	<b>Complexidade</b>
Incluir (EE)	Média
Alterar (EE)	Média
Excluir (EE)	Simple
Procurar (CE)	Média
Visualizar (CE)	Média
Configurar (EE)	Complexa

Tabela 5.107 – Tela Cadastro de Região

<b>Tela - Cadastro de Micro-Região</b>	
<b>Tipo de função transacional</b>	<b>Complexidade</b>
Incluir (EE)	Média
Alterar (EE)	Média
Excluir (EE)	Simple
Procurar (CE)	Média
Visualizar (CE)	Média
Configurar (EE)	Complexa

Tabela 5.108 – Tela Cadastro de Micro-Região

<b>Tela - Cadastro de Unidade</b>	
<b>Tipo de função transacional</b>	<b>Complexidade</b>
Incluir (EE)	<b>Complexa</b>
Alterar (EE)	<b>Complexa</b>
Excluir (EE)	<b>Simples</b>
Procurar (CE)	<b>Complexa</b>
Visualizar (CE)	<b>Complexa</b>
Configurar (EE)	<b>Complexa</b>

Tabela 5.109 – Tela Cadastro de Unidade

<b>Tela - Cadastro de Banco</b>	
<b>Tipo de função transacional</b>	<b>Complexidade</b>
Incluir (EE)	<b>Média</b>
Alterar (EE)	<b>Média</b>
Excluir (EE)	<b>Simples</b>
Procurar (CE)	<b>Média</b>
Visualizar (CE)	<b>Média</b>
Configurar (EE)	<b>Complexa</b>

Tabela 5.110 – Tela Cadastro de Banco

<b>Tela - Preço por unidade</b>	
<b>Tipo de função transacional</b>	<b>Complexidade</b>
Incluir (EE)	<b>Complexa</b>
Alterar (EE)	<b>Complexa</b>
Excluir (EE)	<b>Simples</b>
Procurar (CE)	<b>Complexa</b>
Configurar (EE)	<b>Complexa</b>

Tabela 5.111 – Tela Preço por Unidade

<b>Tela - Preço por veículo</b>	
<b>Tipo de função transacional</b>	<b>Complexidade</b>
Incluir (EE)	<b>Complexa</b>
Alterar (EE)	<b>Complexa</b>
Excluir (EE)	<b>Simple</b>
Procurar (CE)	<b>Complexa</b>
Configurar (EE)	<b>Complexa</b>

Tabela 5.112 – Tela Preço por veículo

<b>Tela - Preço por nota</b>	
<b>Tipo de função transacional</b>	<b>Complexidade</b>
Incluir (EE)	<b>Complexa</b>
Alterar (EE)	<b>Complexa</b>
Excluir (EE)	<b>Simple</b>
Procurar (CE)	<b>Complexa</b>
Configurar (EE)	<b>Complexa</b>

Tabela 5.113 – Tela Preço por nota

<b>Tela Cadastro de Evento</b>	
<b>Tipo de função transacional</b>	<b>Complexidade</b>
Incluir (EE)	<b>Complexa</b>
Alterar (EE)	<b>Complexa</b>
Excluir (EE)	<b>Simple</b>
Procurar (CE)	<b>Complexa</b>
Visualizar (CE)	<b>Complexa</b>
Configurar (EE)	<b>Complexa</b>

Tabela 5.114 – Tela Cadastro de Evento

<b>Tela - Cadastro de Percurso</b>	
<b>Tipo de função transacional</b>	<b>Complexidade</b>
Incluir (EE)	Média
Alterar (EE)	Média
Excluir (EE)	Simple
Procurar (CE)	Média
Visualizar (CE)	Média
Configurar (EE)	Complexa

Tabela 5.115 – Tela Cadastro de Percurso

<b>Tela - Cadastro de Tanque de Resfriamento</b>	
<b>Tipo de função transacional</b>	<b>Complexidade</b>
Incluir (EE)	Simple
Alterar (EE)	Simple
Excluir (EE)	Simple
Procurar (CE)	Simple
Visualizar (CE)	Simple
Configurar (EE)	Complexa

Tabela 5.116 – Tela Cadastro de Tanque de Resfriamento

<b>Tela Cadastro de Fornecedor</b>	
<b>Tipo de função transacional</b>	<b>Complexidade</b>
Incluir (EE)	Média
Alterar (EE)	Média
Excluir (EE)	Simple
Procurar (CE)	Média
Visualizar (CE)	Média
Configurar (EE)	Complexa

Tabela 5.117 – Tela Cadastro de Fornecedor

<b>Tela - Cadastro de Crioscopia</b>	
<b>Tipo de função transacional</b>	<b>Complexidade</b>
Incluir (EE)	<b>Simple</b>
Alterar (EE)	<b>Simple</b>
Excluir (EE)	<b>Simple</b>
Procurar (CE)	<b>Simple</b>
Visualizar (CE)	<b>Simple</b>
Configurar (EE)	<b>Complexa</b>

Tabela 5.118 – Tela Cadastro de Crioscopia

<b>Tela - Unidades liberadas para o Usuário</b>	
<b>Tipo de função transacional</b>	<b>Complexidade</b>
Incluir (EE)	<b>Simple</b>
Alterar (EE)	<b>Simple</b>
Excluir (EE)	<b>Simple</b>
Procurar (CE)	<b>Simple</b>
Configurar (EE)	<b>Complexa</b>

Tabela 5.119 – Tela Unidades liberadas para o usuário

<b>Tela - Cadastro de Ponto</b>	
<b>Tipo de função transacional</b>	<b>Complexidade</b>
Incluir (EE)	<b>Simple</b>
Alterar (EE)	<b>Simple</b>
Excluir (EE)	<b>Simple</b>
Procurar (CE)	<b>Média</b>
Visualizar (CE)	<b>Simple</b>
Configurar (EE)	<b>Complexa</b>

Tabela 5.120 – Tela Cadastro de Ponto

<b>Tela - Cadastro de Ocorrência de Litragem</b>	
<b>Tipo de função transacional</b>	<b>Complexidade</b>
Incluir (EE)	Média
Alterar (EE)	Média
Excluir (EE)	Simple
Procurar (CE)	Média
Visualizar (CE)	Média
Configurar (EE)	Complexa

Tabela 5.121 – Tela Cadastro de ocorrência de litragem

<b>Tela - Preparação de BTLC</b>	
<b>Tipo de função transacional</b>	<b>Complexidade</b>
Incluir (EE)	Média
Alterar (EE)	Média
Excluir (EE)	Simple
Procurar (CE)	Média
Visualizar (CE)	Média
Configurar (EE)	Complexa

Tabela 5.122 – Tela Preparação de BTLC

<b>Tela - BTLC Fazenda</b>	
<b>Tipo de função transacional</b>	<b>Complexidade</b>
Incluir (EE)	Complexa
Alterar (EE)	Complexa
Excluir (EE)	Média
Procurar (CE)	Complexa
Visualizar (CE)	Complexa
Configurar (EE)	Complexa

Tabela 5.123 – Tela BTLC Fazenda

<b>Tela - BTLC Transferência</b>	
<b>Tipo de função transacional</b>	<b>Complexidade</b>
Incluir (EE)	<b>Complexa</b>
Alterar (EE)	<b>Complexa</b>
Excluir (EE)	<b>Simplex</b>
Procurar (CE)	<b>Complexa</b>
Visualizar (CE)	<b>Complexa</b>
Configurar (EE)	<b>Complexa</b>

Tabela 5.124 – Tela BTLC Transferência

<b>Tela - BTLC Compra</b>	
<b>Tipo de função transacional</b>	<b>Complexidade</b>
Incluir (EE)	<b>Complexa</b>
Alterar (EE)	<b>Complexa</b>
Excluir (EE)	<b>Simplex</b>
Procurar (CE)	<b>Complexa</b>
Visualizar (CE)	<b>Complexa</b>
Configurar (EE)	<b>Complexa</b>

Tabela 5.125 – Tela BTLC Compra

<b>Tela - BTLC Socorro</b>	
<b>Tipo de função transacional</b>	<b>Complexidade</b>
Incluir (EE)	<b>Complexa</b>
Alterar (EE)	<b>Complexa</b>
Excluir (EE)	<b>Simplex</b>
Procurar (CE)	<b>Complexa</b>
Visualizar (CE)	<b>Complexa</b>
Configurar (EE)	<b>Complexa</b>

Tabela 5.126 – Tela BTLC Socorro

<b>Tela - Cadastro de Motivo de Memorando</b>	
<b>Tipo de função transacional</b>	<b>Complexidade</b>
Incluir (EE)	<b>Simples</b>
Alterar (EE)	<b>Simples</b>
Excluir (EE)	<b>Simples</b>
Procurar (CE)	<b>Simples</b>
Visualizar (CE)	<b>Simples</b>
Configurar (EE)	<b>Complexa</b>

Tabela 5.127 – Tela Cadastro de motivo de memorando

<b>Tela – Memorando</b>	
<b>Tipo de função transacional</b>	<b>Complexidade</b>
Incluir (EE)	<b>Média</b>
Alterar (EE)	<b>Média</b>
Excluir (EE)	<b>Simples</b>
Procurar (CE)	<b>Média</b>
Visualizar (CE)	<b>Média</b>
Configurar (EE)	<b>Complexa</b>

Tabela 5.128 – Tela Memorando

<b>Tela - Cadastro de Período de Pagamento</b>	
<b>Tipo de função transacional</b>	<b>Complexidade</b>
Incluir (EE)	<b>Média</b>
Alterar (EE)	<b>Média</b>
Excluir (EE)	<b>Simples</b>
Procurar (CE)	<b>Média</b>
Visualizar (CE)	<b>Média</b>
Configurar (EE)	<b>Complexa</b>

Tabela 5.129 – Tela Cadastro de período de pagamento

<b>Tela - Cadastro de Produtor</b>	
<b>Tipo de função transacional</b>	<b>Complexidade</b>
Incluir (EE)	<b>Complexa</b>
Alterar (EE)	<b>Complexa</b>
Excluir (EE)	<b>Simplex</b>
Procurar (CE)	<b>Complexa</b>
Visualizar (CE)	<b>Complexa</b>
Configurar (EE)	<b>Complexa</b>

Tabela 5.130 – Tela Cadastro de produtor

<b>Tela - Altera Frete</b>	
<b>Tipo de função transacional</b>	<b>Complexidade</b>
Alterar (EE)	<b>Complexa</b>

Tabela 5.131 – Tela Altera Frete

<b>Tela - Fecha Lista Leite</b>	
<b>Tipo de função transacional</b>	<b>Complexidade</b>
Alterar (EE)	<b>Complexa</b>

Tabela 5.132 – Tela Fecha Lista Leite

<b>Tela - Transfere dados da Fazenda entre Rotas</b>	
<b>Tipo de função transacional</b>	<b>Complexidade</b>
OK (EE)	<b>Complexa</b>

Tabela 5.133 – Tela Transfere dados da Fazenda entre Rotas

<b>Tela – Integração</b>	
<b>Tipo de função transacional</b>	<b>Complexidade</b>
OK (EE)	<b>Média</b>

Tabela 5.134 – Tela Integração

<b>Tela - Transfere dados da Fazenda entre Rotas</b>	
<b>Tipo de função transacional</b>	<b>Complexidade</b>
OK (CE)	Simple

Tabela 5.135 – Tela Transfere dados da fazenda entre rotas

<b>Tela - Relatório de Município sem Banco</b>	
<b>Tipo de função transacional</b>	<b>Complexidade</b>
Filtro e Relatório (SE)	Simple

Tabela 5.136 – Tela relatório de município sem banco

<b>Tela - Relatório de Banco por Município</b>	
<b>Tipo de função transacional</b>	<b>Complexidade</b>
Filtro e Relatório (SE)	Média

Tabela 5.137 – Tela Relatório de banco por município

<b>Tela - Relatório de Fazenda por Produtor</b>	
<b>Tipo de função transacional</b>	<b>Complexidade</b>
Filtro e Relatório (SE)	Complexa

Tabela 5.138 – Tela relatório de fazenda por produtor

<b>Tela – Relatório Fazenda por Região</b>	
<b>Tipo de função transacional</b>	<b>Complexidade</b>
Filtro e Relatório (SE)	Complexa

Tabela 5.139 – Tela relatório fazenda por região

<b>Tela - Relatório de Mapa da Rota</b>	
<b>Tipo de função transacional</b>	<b>Complexidade</b>
Filtro e Relatório (SE)	Complexa

Tabela 5.140 – Tela Relatório de mapa da rota

Após feita toda a contagem e determinados os graus de complexidade com seus respectivos pesos, valorados conforme as tabelas descritas no capítulo 3, pode-se obter através da soma de todos os valores contados nas funções de dados e nas funções transacionais o tamanho não ajustado do sistema, neste caso foram contados 911 pontos de função não ajustados.

Estando com as PF's não ajustadas, é feito o ajuste dos mesmos calculando-se primeiro o fator de ajuste de valores, que é baseado em valores que variam entre 0 e 5 pontos conforme o comportamento do sistema em relação a cada característica. Para fins de uma simulação foi adotado como quantidade de pontos de característica gerais do sistema o valor 45 – valor estipulado como média para sistemas de pequeno a médio porte – valor este que após atribuído a forma do FAV resultará no valor 1.1.

Uma vez estando com os valores não ajustados e com o FAV já aplicado, os mesmos são multiplicados gerando assim o tamanho final desse sistema: 1002,1 pontos por função.

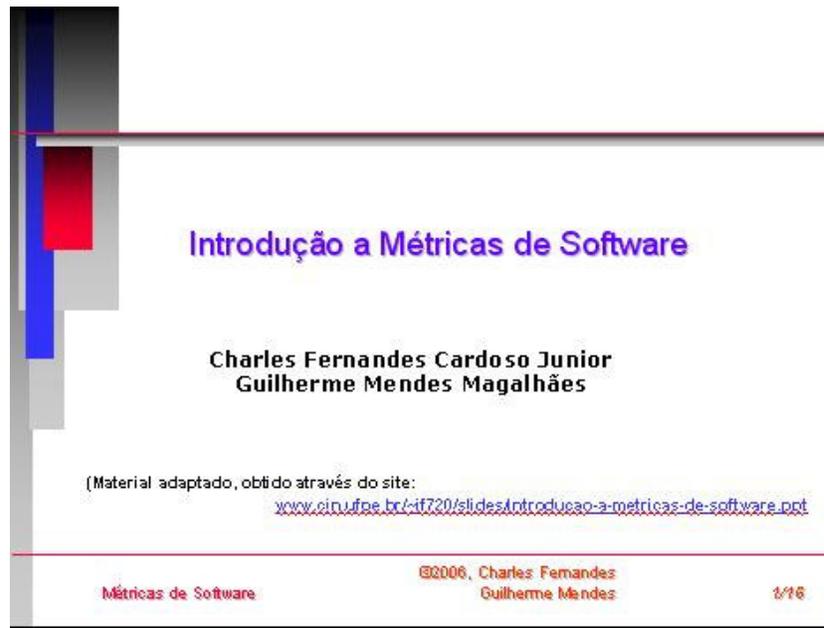
Ao finalizar este estudo de caso pode-se notar que o processo de medição de software é um trabalho árduo que necessita de um plano de ação, tempo, esforço e comprometimento, envolvendo desde a alta direção até os seus colaboradores, pois estes atributos fazem a diferença no resultado final e no objetivo proposto dentro desse processo.

## 6. TREINAMENTO

Na fase de idealização do estágio supervisionado, o qual foi agraciado com o tema abordando Engenharia de Software com ênfase em métricas de software por pontos de função, apresentamos à empresa Seta Sistemas a possibilidade de ser ministrado um treinamento para que a empresa pudesse ter um conhecimento mais profundo e abrangente sobre o que é e como se aplica métricas de software através da funcionalidade do produto. Este treinamento teve como objetivo facilitar o entendimento da empresa sobre o assunto abordado e esclarecer sobre os benefícios que se obtém em usar tal recurso em seu produto e caso a mesma venha a trabalhar posteriormente com métricas de software por pontos de função, ela já tenha uma linha base de conhecimento, podendo assim, ter uma facilidade no processo de implementação.

Para uma melhor explanação de métricas de software por pontos de função, foi elaborado um material de apoio para o treinamento. O material foi desenvolvido usando o MS PowerPoint, uma vez que a abordagem seria auxiliada por slides através de um datashow. Gostaríamos de salientar ainda, que o material foi extraído de uma disponibilização na internet através do endereço [www.cin.ufpe.br/~if720/slides/introducao-a-metricas-de-software.ppt](http://www.cin.ufpe.br/~if720/slides/introducao-a-metricas-de-software.ppt) e adaptado à nossa necessidade.

Segue-se então o respectivo material que auxiliou no treinamento de pontos por função na empresa Seta Sistemas. Ao final deste material será abordado os resultados obtidos com o treinamento ministrado.



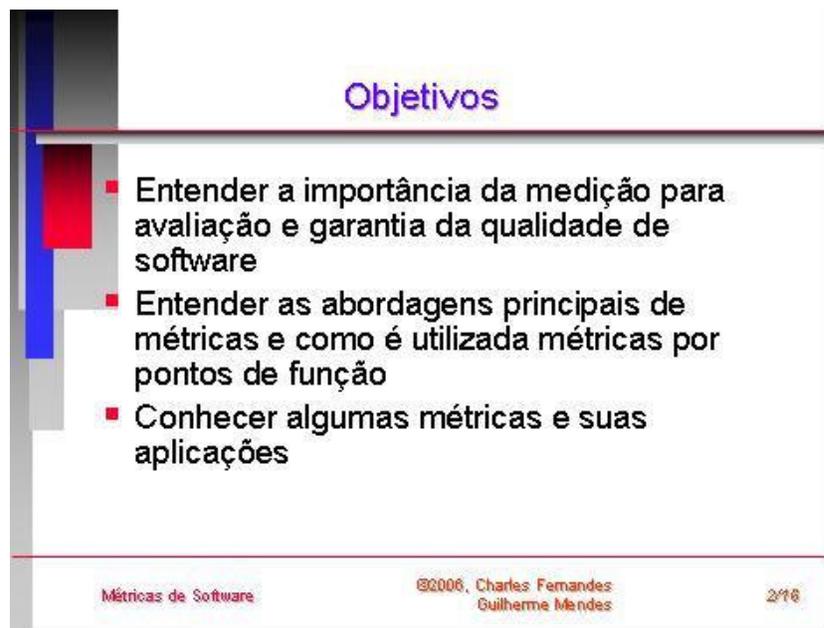
**Introdução a Métricas de Software**

**Charles Fernandes Cardoso Junior**  
**Guilherme Mendes Magalhães**

(Material adaptado, obtido através do site:  
[www.cin.ufpe.br/~if720/slides/introducao-a-metricas-de-software.ppt](http://www.cin.ufpe.br/~if720/slides/introducao-a-metricas-de-software.ppt))

Métricas de Software ©2006, Charles Fernandes  
Guilherme Mendes 1/16

Figura 6.1 – Slide Introdução



**Objetivos**

- Entender a importância da medição para avaliação e garantia da qualidade de software
- Entender as abordagens principais de métricas e como é utilizada métricas por pontos de função
- Conhecer algumas métricas e suas aplicações

Métricas de Software ©2006, Charles Fernandes  
Guilherme Mendes 2/16

Figura 6.2 – Slide Objetivos

## O que são métricas de software?

- Uma métrica é a medição de um atributo (propriedades ou características) de uma determinada entidade (produto, processo ou recursos). Exemplos:
  - ◆ Tamanho do produto de software (ex: Número de Linhas de código)
  - ◆ Número de pessoas necessárias para implementar um caso de uso
  - ◆ Número de defeitos encontrados por fase de desenvolvimento
  - ◆ Esforço para a realização de uma tarefa
  - ◆ Tempo para a realização de uma tarefa
  - ◆ Custo para a realização de uma tarefa
  - ◆ Grau de satisfação do cliente (ex: adequação do produto ao propósito, conformidade do produto com a especificação)

Métricas de Software ©2006, Charles Fernandes  
Guilherme Mendes 3/16

Figura 6.3 – Slide o que são métricas de software?

## Por que medir software?

- Entender e aperfeiçoar o processo de desenvolvimento
- Melhorar a gerência de projetos e o relacionamento com clientes
- Reduzir frustrações e pressões de cronograma
- Gerenciar contratos de software
- Indicar a qualidade de um produto de software
- Avaliar a produtividade do processo
- Avaliar os benefícios (em termos de produtividade e qualidade) de novos métodos e ferramentas de engenharia de software
- Avaliar retorno de investimento

Métricas de Software ©2006, Charles Fernandes  
Guilherme Mendes 4/16

Figura 6.4 – Slide por que medir software?

**Por que medir software?**

- Identificar as melhores práticas de desenvolvimento de software
- Embasar solicitações de novas ferramentas e treinamento
- Avaliar o impacto da variação de um ou mais atributos do produto ou do processo na qualidade e/ou produtividade
- Formar uma *baseline* para estimativas
- Melhorar a exatidão das estimativas
- Oferecer dados qualitativos e quantitativos ao gerenciamento de desenvolvimento de software, de forma a realizar melhorias em todo o processo de desenvolvimento de software

Métricas de Software ©2006, Charles Fernandes  
Guilherme Mendes 5/16

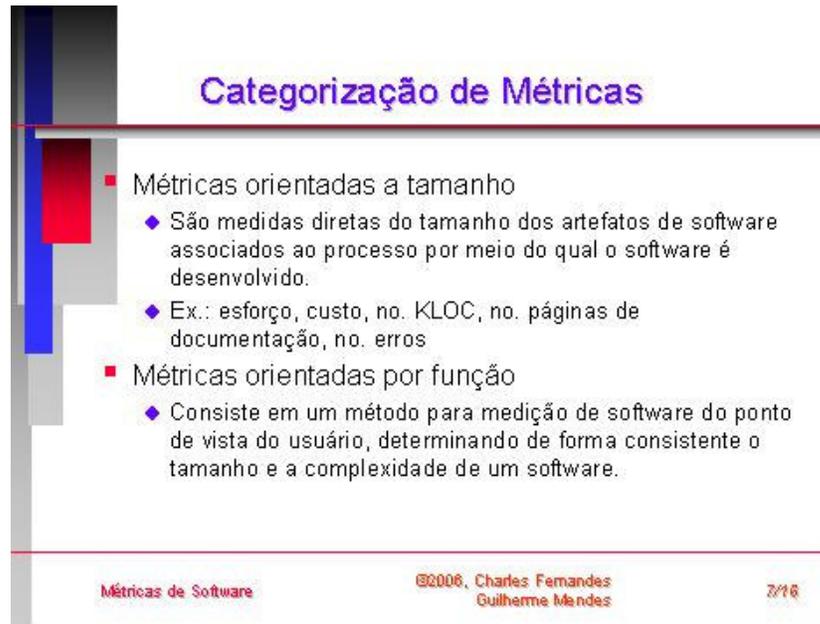
Figura 6.5 – Slide por que medir software? (2)

**Em resumo...**

- Uma métrica deve ser:
  - ◆ Válida: quantifica o que queremos medir
  - ◆ Confiável: produz os mesmos resultados dadas as mesmas condições
  - ◆ Prática: barata, fácil de computar e fácil de interpretar
- Dois contextos para medição de software
  - ◆ Processo: ex. produtividade
  - ◆ Produto: ex. qualidade

Métricas de Software ©2006, Charles Fernandes  
Guilherme Mendes 6/16

Figura 6.6 – Slide em resumo

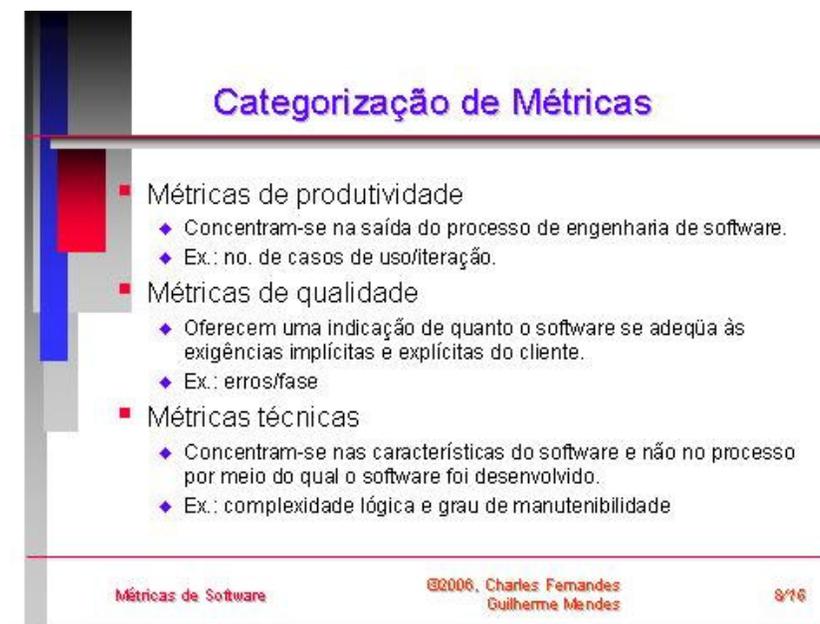


**Categorização de Métricas**

- Métricas orientadas a tamanho
  - ◆ São medidas diretas do tamanho dos artefatos de software associados ao processo por meio do qual o software é desenvolvido.
  - ◆ Ex.: esforço, custo, no. KLOC, no. páginas de documentação, no. erros
- Métricas orientadas por função
  - ◆ Consiste em um método para medição de software do ponto de vista do usuário, determinando de forma consistente o tamanho e a complexidade de um software.

Métricas de Software ©2006, Charles Fernandes  
Guilherme Mendes 7/16

Figura 6.7 – Slide categorização de métricas



**Categorização de Métricas**

- Métricas de produtividade
  - ◆ Concentram-se na saída do processo de engenharia de software.
  - ◆ Ex.: no. de casos de uso/iteração.
- Métricas de qualidade
  - ◆ Oferecem uma indicação de quanto o software se adequa às exigências implícitas e explícitas do cliente.
  - ◆ Ex.: erros/fase
- Métricas técnicas
  - ◆ Concentram-se nas características do software e não no processo por meio do qual o software foi desenvolvido.
  - ◆ Ex.: complexidade lógica e grau de manutenibilidade

Métricas de Software ©2006, Charles Fernandes  
Guilherme Mendes 8/16

Figura 6.8 – Slide categorização de métricas (2)

## Os Quatros papéis de Medição

- Entender
  - ◆ Métricas ajudam a entender o comportamento e funcionamento de processos, produtos e serviços de software
- Avaliar
  - ◆ Métricas podem ser utilizadas para tomar decisões e determinar o estabelecimento de padrões, metas e critérios de aceitação
- Controlar
  - ◆ Métricas podem ser utilizadas para controlar processos, produtos e serviços de software
- Prever
  - ◆ Métricas podem ser utilizadas para prever valores de atributos

Métricas de Software ©2006, Charles Fernandes  
Guilherme Mendes 9/16

Figura 6.9 – Slide os quatros papéis de medição

## Principais Barreiras



- Falta de comprometimento da alta gerência
- Medir custa caro
- Os maiores benefícios vêm a longo prazo
- Má utilização das métricas
- Grande mudança cultural necessária
- Dificuldade de estabelecer medições apropriadas e úteis
- Interpretações dos dados realizadas de forma incorreta
- Obter o comprometimento de todos os envolvidos e impactados
- Estabelecer um programa de medições é fácil, o difícil é manter!!

Métricas de Software ©2006, Charles Fernandes  
Guilherme Mendes 10/16

Figura 6.10 – Slide principais barreiras

**Mas podemos contornar ...**

- Foco desde os estágios iniciais da melhoria de processo
- Começar Pequeno
- Selecionar um conjunto coerente
- É importante definir cada detalhe da métrica
- Descartar o que não estiver sendo útil
- Fornecer as informações corretas, para as pessoas certas
- "Agregar valor", ao invés de gerar apenas dados

Métricas de Software ©2006, Charles Fernandes  
Guilherme Mendes

Figura 6.11 – Slide mas podemos contornar...

**Mas podemos contornar ...**

- Incentivar a equipe de desenvolvimento a fazer uso das métricas
- Envolvimento de todos os impactados
- Estabelecer as expectativas
- Educação e Treinamento
- Ganhar Confiança
- Adotar uma Abordagem Evolucionária
- Compreender que a Adoção leva Tempo

Métricas de Software ©2006, Charles Fernandes  
Guilherme Mendes

Figura 6.12 – Slide Mas podemos contornar... (2)

## O Que é o Processo de Contagem de Pontos de Função?

- Os passos básicos envolvidos na contagem de pontos de função incluem:
- Determinar o tipo de contagem (pode ser um projeto de novo desenvolvimento, uma contagem básica de aplicação ou uma contagem de projeto de melhoria)

Figura 6.13 – Slide o que é o processo de contagem

## Funções de Dados

- Contar os tipos de funções de dados divididos em:
- **Arquivos Lógicos Internos** ou ALIs, que são os grupos lógicos de dados mantidos dentro da fronteira da aplicação;
- **Arquivos de Interface Externa** ou AIEs, os quais são apenas referenciados pela aplicação.
- Cada ALI vale 7, 10 ou 15 PF, enquanto cada AIE vale 5, 7 ou 10 PF.

Figura 6.14 – Slide funções de dados

## Tabela de Identificação da Complexidade das Funções de Dados

Número de Registros Lógicos	Número de Itens de Dados Referenciados		
	De 1 a 19	De 20 a 50	51 ou mais
Apenas 1	Simples	Simples	Média
De 2 a 5	Simples	Média	Complexa
6 ou mais	Média	Complexa	Complexa

Métricas de Software

©2006, Charles Fernandes  
Guilherme Mendes

15/16

Figura 6.15 – Slide tabela de identificação complexidade FD

## Funções de Transações

- Contar os tipos de funções de transações divididos em:
  - Entradas Externas ou EEs**, que são processos de entrada de dados;
  - Saídas Externas ou SEs**, por exemplo, relatórios;
  - Consultas Externas ou CEs**, por exemplo, *Consultar Detalhes de Empregados*.]
- Cada EE ou CE vale 3, 4 ou 6 pontos de função, enquanto cada SE vale 4, 5 ou 7 pontos de função.

Métricas de Software

©2006, Charles Fernandes  
Guilherme Mendes

16/16

Figura 6.16 – Slide funções de transações

## Tabela de Identificação da Complexidade de Entradas Externas

Número de Arquivos Referenciados	Número de Itens de Dados Referenciados		
	De 1 a 4	De 5 a 15	16 ou mais
0 ou 1	Simple	Simple	Média
2	Simple	Média	Complexa
3 ou mais	Média	Complexa	Complexa

Métricas de Software

©2006, Charles Fernandes  
Guilherme Mendes

17/16

Figura 6.17 – Slide tabela de identificação complexidade EE

## Tabela de Identificação da Complexidade de Saídas e Consultas Externas

Número de Arquivos Referenciados	Número de Itens de Dados Referenciados		
	De 1 a 5	De 6 a 19	20 ou mais
0 ou 1	Simple	Simple	Média
2 ou 3	Simple	Média	Complexa
4 ou mais	Média	Complexa	Complexa

Métricas de Software

©2006, Charles Fernandes  
Guilherme Mendes

18/16

Figura 6.18 – Slide identificação da complexidade

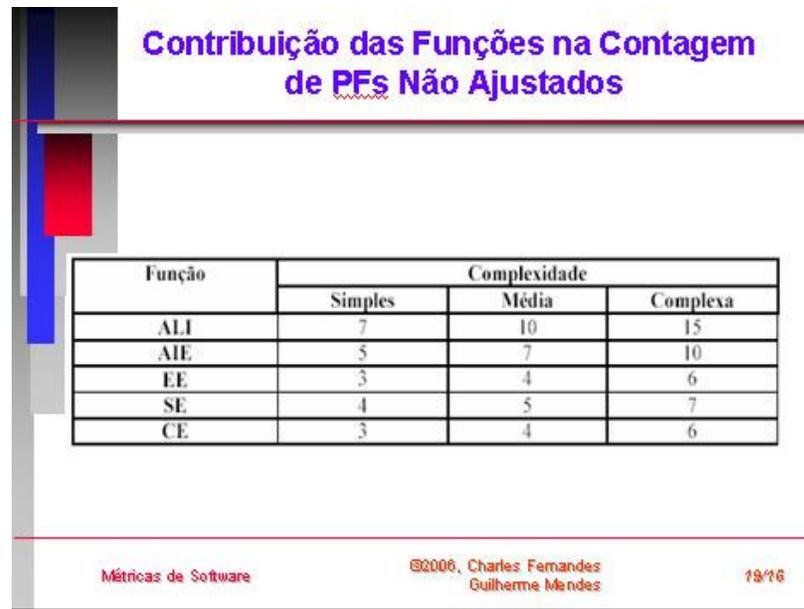


Figura 6.19 – Slide contribuição das funções

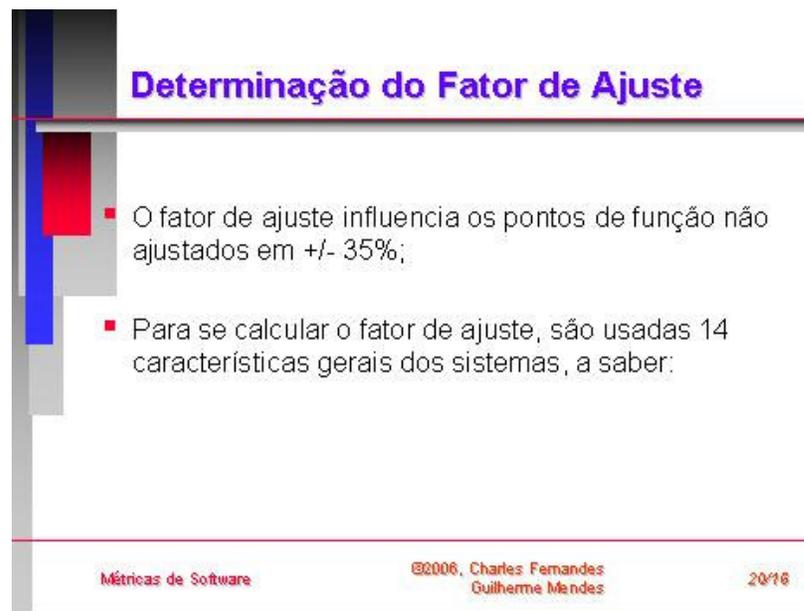


Figura 6.20 – Slide determinação do fator de ajuste

## Características Gerais do Sistema

- 1. Comunicação de Dados
- 2. Processamento Distribuído de Dados
- 3. Desempenho
- 4. Configuração Intensamente Utilizada
- 5. Taxa de Transação
- 6. Entrada de Dados On-Line
- 7. Eficiência do Usuário Final
- 8. Atualização On-Line
- 9. Processamento Complexo
- 10. Reutilização
- 11. Facilidade de Instalação
- 12. Facilidade de Operação
- 13. Múltiplas Localidades
- 14. Facilidade de Alteração

Métricas de Software ©2006, Charles Fernandes  
Guilherme Mendes 21/16

Figura 6.21 – Slide características gerais do sistema

## Características Gerais do Sistema

- Determinar o Fator de Ajuste de Valor (FAV) baseado na equação:  
$$(FAV = 0,65 + (\text{Soma das Características Gerais do Sistema} \times 0,01))$$
- Calcular a contagem ajustada final de PF:  
$$\text{contagem final de PF} = \text{contagem não ajustada} * FAV$$

Métricas de Software ©2006, Charles Fernandes  
Guilherme Mendes 22/16

Figura 6.22 – Slide características gerais do sistema (2)

**Referências**

- Pressman, Roger. S. Engenharia de Software. Makron Books, 1995
- IFPUG International Function Point Users Group: [www.ifpug.org](http://www.ifpug.org)
- Fatto Consultoria e Sistemas: [www.fattocs.com.br](http://www.fattocs.com.br)

Métricas de Software ©2006, Charles Fernandes  
Guilherme Mendes 23/16

Figura 6.23 – Slide referências

**Contatos**

**Guilherme Mendes**  
[guilherme373@yahoo.com.br](mailto:guilherme373@yahoo.com.br)

**Charles Fernandes**  
[charx@ibest.com.br](mailto:charx@ibest.com.br)

Métricas de Software ©2006, Charles Fernandes  
Guilherme Mendes 24/16

Figura 6.24 – Slide contatos

## 6.1 - Resultados do workshop

Na tarde do dia 12 de Maio de 2006, mais precisamente às 15:00 hs, na sala de treinamento da empresa Seta Sistemas, iniciou-se o referido treinamento sobre métricas de software com pontos de função. O treinamento contou com a estrutura local oferecida pela empresa, bem como um datashow para reprodução dos slides e explanação do contexto abordado. Devido ao fato da empresa estar em horário de expediente e tendo que atender as solicitações de seus clientes, não foi possível mediar um treinamento focado para toda a equipe, no entanto, como uma forma estratégica de melhor aproveitar a oportunidade oferecida mediante o workshop que seria aplicado, participou ativamente o Sr. Daniel Fontes Coelho Júnior, vulgo Jhones, o qual além de ser um dos Analistas/Programadores é também responsável pela gerência do setor de desenvolvimento da Seta Sistemas. Devido a essa posição ocupada pelo Sr. Daniel na empresa, ele foi uma peça chave na aplicação e efetivação do treinamento ministrado naquela tarde ensolarada de sexta feira para obter um resultado satisfatório em termos técnicos e de enriquecimento teórico embasado no assunto referente a métricas de softwares usando como medição a técnica de contagem por pontos de função, a qual mede o tamanho do software através da funcionalidade do mesmo em relação ao usuário.

Para expor de uma forma mais aplausível o resultado obtido em relação ao treinamento no sentido de saber se de fato foi aproveitável pelo participante do workshop, avaliando se dentro do contexto apresentado houve uma aquisição de conhecimento por parte do mesmo, elaboramos algumas perguntas a serem feitas ao Sr. Daniel Fontes Coelho Júnior. Essas perguntas tiveram como única finalidade obter uma avaliação focada no resultado do treinamento, ou seja, se o mesmo apresentou um ponto positivo ou negativo. Ao retornar-mos na empresa, na outra semana após o treinamento, dialogamos com o Sr. Daniel dentro dos seguintes questionamentos:

**Pergunta:** Daniel, gostaríamos de saber se você já conhecia métricas de software e pontos de função, caso sim, qual era o seu nível de conhecimento em relação a esse assunto?

**Resposta:** “Apesar de ter conhecimento em relação as técnicas da engenharia de software e métricas por pontos de função, tal conhecimento não é muito profundo em relação a medidas de software. Já tive contato com o assunto quando cursei a faculdade e naquela época a engenharia de software não tinha um foco tão relevante como hoje em dia, pois atualmente na busca da qualidade as empresas de desenvolvimento estão buscando conhecer mais a respeito do assunto para que seus softwares possam ser mais completos em relação a funcionalidade para o qual os mesmos estão sendo desenvolvidos. Tive contato também sobre o assunto em alguns cursos que fiz e palestras que participei e mesmo já tendo ouvido falar e algumas vezes lido a respeito da possibilidade de se medir um software nunca fiz usabilidade e aplicabilidade real de tal técnica”.

**Pergunta:** Como você avalia a sua participação no workshop que foi ministrado? Você acredita ter obtido um ganho em relação ao conhecimento sobre o assunto?

**Resposta:** “Sim, acredito que foi muito produtivo e aproveitável para mim, pois, conforme tinha dito, eu até tinha um pouco de conhecimento sobre a existência de medidas de software, mas não sabia de fato como funcionava a contagem, o que realmente era contado e como as técnicas de medição eram aplicadas. Após o treinamento, consigo identificar melhor o que é métrica de software e como fazer a contagem.”

**Pergunta:** Qual a maior dificuldade que a empresa tem para fazer uso da aplicabilidade de métricas de software?

**Resposta:** “Observo que existem três fatores que podem ser considerados como empecilho para a aplicabilidade de tal técnica:

Primeiro: Existe um problema relacionado a mão de obra, pois, atualmente aqui na empresa não existe uma pessoa que esteja com tempo disponível para ser dispersado para a aplicação de métricas, pois, acredito que seria necessário um profissional específico para essa área.

Segundo: Por necessitar de profissional voltado exclusivamente para tal área, isso irá gerar um certo custo para a empresa. Não estou falando aqui de custo-benefício, pois, por mais que tenha sido apresentado no treinamento alguns benefícios, teríamos que ter uma abordagem real para tal avaliação. Por isso acredito que teríamos um certo custo em relação a essa aplicabilidade.

Terceiro: Devido ao fato de não termos domínio sobre o assunto isso gera em nós uma certa incerteza em relação aos resultados que poderão ser obtidos. Não temos uma linha de base para que possamos usar como referência e por isso deveríamos fazer uma pesquisa e um estudo mais aprofundado para que seja possível fazer uma real aplicação de medição aqui na empresa.

Apesar de haver um certo receio, acreditamos que em se tratando de algo que possa contribuir para a qualidade, melhoria e produtividade do software desenvolvido por nós, teríamos sim interesse em fazer abordagem da técnica, mas devemos conhecer mais sobre o assunto para uma possível abordagem”.

Pode-se observar através desta breve conversa com o Sr. Daniel Fontes Coelho Júnior que o treinamento teve um ponto positivo no sentido de ter mostrado para a empresa a existência e uso de métricas de software como foi objetivado, no entanto, como ele mesmo frisou, a falta de um conhecimento profundo sobre o assunto cria um certo receio sobre a aplicação da técnica, pois não se sabe o quanto de esforço será despendido para a obtenção de resultados e se o resultado será de fato satisfatório, mediante o custo que será despendido para a aquisição do mesmo.

Este é um ponto interessante a ser observado, pois um dos anseios em fazer este trabalho é levar as empresas ao conhecimento de métricas de software e despertar o interesse pela aplicação da engenharia de software na construção/manutenção do seu produto. Estamos apenas no começo, mas iremos avançar para que essa possibilidade venha a ser uma mera concretização de um fato consumado.

## CONSIDERAÇÕES FINAIS

A Engenharia de Software provê técnicas que ajudam a manter um melhor domínio sobre as etapas de criação do software, bem como uma melhor abordagem no que diz respeito à qualidade e evolução do mesmo, provendo assim, mais satisfação por parte do usuário, o qual é uma peça chave dentro desse segmento.

Software é um produto caro, que necessita ser lapidado e moldado pelos nortes da engenharia e ser conhecido quanto ao esforço dispersado para a sua concretização, bem como saber qual é de fato o tamanho de sua funcionalidade através de técnicas de métricas orientadas a pontos por função.

Após obter o conhecimento dos fatos e acontecimentos fornecidos até então, é notável que existe uma certa brecha em meio as empresas de desenvolvimento de software em relação ao produto que as mesmas desenvolvem. Isso acontece devido ao fato de grande parte das empresas, em algumas vezes, negligenciarem ou desconhecerem os recursos e técnicas que são fornecidos pela Engenharia de Software, o que as levam a não usufruírem de fatores que proporcionam um maior poder de controle em relação ao produto que as mesmas desenvolvem.

As métricas orientadas a pontos por função medem não o tamanho do software em relação ao seu comprimento, mas sim em relação à funcionalidade que o mesmo concede aos seus usuários.

Dentro de uma conclusão genérica, é chegado ao término deste trabalho de estágio supervisionado com uma satisfação pelos componentes do estágio e pela empresa participante, pois esse relacionamento entre ambos, gerou oportunidades aceitáveis e permitiu que fosse evoluído o conhecimento

sobre o assunto abordado, alcançando novas possibilidades e descortinando horizontes, os quais, estão em fase de mudança devido ao fato da Engenharia de Software estar cada vez mais madura, evolutiva, apta a participar e auxiliar nas grandes demandas por softwares existente no cenário mundial.

## Glossário

### B

**BATCH** – É um modo de processamento de dados no qual os dados de entrada são coletados em grupos ou lotes.

### C

**CÓDIGO FONTE** - é o conjunto de palavras escritas de forma ordenada, contendo instruções em uma das linguagens de programação existentes no mercado, de maneira lógica.

**CPM** - Counting Practices Manual (Manual Prático de Contagem)

**CPU** - Central Processing Unit (Unidade Central de Processamento)

### H

**HARD COPY** - Cópia impressa, impressão, cópia em papel, impresso menus pop-up - menu instantâneo, que aparece temporariamente em forma de janela, sobreposta informações já presentes na tela

**HELP ON-LINE** - Ajuda disponível em uma rede de comunicação qualquer.

### I

**INTERNET** - A Internet é uma rede de redes em escala mundial de milhões de computadores que permite o acesso a informações e todo tipo de transferência de dados.

### M

**MULTIMIDIA** - é o uso combinado de diferentes tipos de media na comunicação - texto, áudio, imagem estática (fotografia, ilustração) ou em movimento (vídeo, animação).

## O

**ON-LINE** - Termo utilizado para designar quando um computador está conectado à uma rede ou qualquer tipo de comunicação entre computadores

## S

**STAND-ALONE** - São aplicativos completamente auto-suficientes, não necessitando de um segundo software para o seu funcionamento.

**SUPORTE BILINGUE** - Suporte com versão em dois idiomas

**SUPORTE MULTILINGUE** - Suporte com versão em mais de dois idiomas

## FONTES CONSULTADAS

[PRE05] PRESSMAN, Roger S. Engenharia de Software. São Paulo: Makron Books, 2005.

VAZQUEZ, Carlos Eduardo. SIMÕES, Guilherme Siqueira. ALBERT, Renato Machado. Análise de pontos de função. Medição, Estimativas e Gerenciamento de Projetos de Software. 4ª Edição. São Paulo: Érica, 2005.

Site: <http://www.bfpuq.com.br/Artigos/Dekkers-PontosDeFuncaoEMedidas.htm>,  
Pesquisado em 26/03/2006 às 06h30.

[UFE05]Site: <http://www.inf.ufes.br/~falbo/download/aulas/es-q/2005-1/APF.pdf>,  
pesquisado em 26/03/2006 às 06h35.