

CENTRO UNIVERSITÁRIO FEEVALE

CARLOS MARCELO FABIAN

SICLA-APF FERRAMENTA DE ANÁLISE DE PONTOS POR
FUNÇÃO

Novo Hamburgo, Junho de 2008.

CARLOS MARCELO FABIAN

SICLA-APF FERRAMENTA DE ANÁLISE DE PONTOS POR
FUNÇÃO

Centro Universitário Feevale
Instituto de Ciências Exatas e Tecnológicas
Curso de Ciência da Computação
Trabalho de Conclusão de Curso

Professor Orientador: Sandra Teresinha Miorelli

Novo Hamburgo, Junho de 2008.

AGRADECIMENTOS

Gostaria de agradecer a todos os que, de alguma maneira, contribuíram para a realização desse trabalho de conclusão, em especial:

A DEUS que sempre esteve comigo, aos meus pais que desde cedo vislumbraram que o meu futuro seria a área de ciência da computação, investiram em mim. A minha esposa que sempre me apoiou incondicionalmente para chegar até onde cheguei.

Aos amigos e às pessoas que convivem comigo diariamente, minha gratidão.

RESUMO

A necessidade constante de aumento da qualidade e melhoria dos processos de desenvolvimento de software, implicando assim na melhora da produtividade e por consequência na redução de custos têm se percebido uma tônica das empresas que fabricam programas de computador, conhecidas como “*software houses*”. A qualidade do software depende de um considerável nível de processo de desenvolvimento bem estruturado (Pressman, 2002). Até os dias atuais, percebemos que as empresas assumem riscos em projetos estipulando prazos e custos baseando-se no “*feeling*” e/ou experiências em projetos anteriores. Os resultados obtidos na maioria das vezes nem sempre são satisfatórios, prazos estourados, sem falar no alto custo de desenvolvimento (Pressman, 2002). Em virtude disto, podemos ver algumas empresas adotando modelos de qualidade de software, visando melhorar os seus processos. Este trabalho de conclusão teve como objetivo estudar os processos da empresa Rech Informática Ltda. Com crescimento constante, a Rech Informática tem uma considerável demanda de implementações, e ainda tem necessidade de criar novos recursos para manter e conquistar novos clientes. Partindo deste pressuposto, a direção da empresa e colaboradores da área de planejamento de projetos aponta a complexidade de estimar prazos, custos e gerenciar a concorrência da demanda. A proposta deste trabalho foi analisar ferramentas existentes no mercado e desenvolver um protótipo de software que utilize a técnica da Análise de Pontos por Função (APF). O objetivo final deste protótipo é ser incorporado ao CRM da Rech para auxiliar a gerência de projetos, por indicação da direção da empresa. Para validação do protótipo proposto, foi utilizado como estudo de caso um projeto desenvolvido recentemente. Este projeto compreende novas funcionalidades do módulo de Contabilidade SIGER®. O protótipo foi desenvolvido pela mesma ferramenta que a empresa desenvolveu seu CRM, baseado na metodologia estruturada.

Palavras-chave: Prazos, Custos, Produtividade, Pontos por função.

ABSTRACT

The constant efforts to increase quality and improve the software development process, in order to improve production and lower costs have been the companies which made programs focus, known as “software houses”. The software quality depends of a considerable structured development process level (Pressman, 2002). Nowadays, we can see the companies taking risks on projects getting dates and costs based on their feelings and/or experiences from the last project. The results don’t satisfy many times, late projects and a higher development costs (Pressman, 2002). That’s why we can see companies adopting software quality models, to avoid their processes. This graduate theory studied Rech Informática Ltda processes. The company is always growing, it has a considerable implementation demand, and need to create new features to keep and get new clients. From this subject, the company directors and the project planning department employees says that it’s hard to get dates, costs and manage a high level of software demand. This study suggested an analysis of existing softwares and develop a software prototype based on function point analysis (FPA). The final goal of this prototype is to be incorporated to the CRM’s Rech to help on manage projects, asked by the company directors. To validate the prototype, were used one recently developed project. This project has new functions in the accounting software SIGER. The prototype were developed using the same developing tool that company developed their CRM, based on their structured software methodology.

Key words: Dates, costs, production, function points.

LISTA DE FIGURAS

Figura 1 – Fluxo do processo de contagem de pontos por função. _____	24
Figura 2 – Fronteira de aplicação _____	26
Figura 3 – Processo de contagem de funções do tipo transação _____	32
Figura 4 – Processos da Rech Informática Ltda _____	44
Figura 5 – Estrutura de dados do protótipo SICLA_APF _____	58
Figura 6 – Menu de cadastros do protótipo SICLA_APF _____	60
Figura 7 – Menu contagem do protótipo SICLA_APF _____	60
Figura 8 – Janela de manutenção do tipo de contagem do protótipo SICLA_APF _____	61
Figura 9 – Janela de manutenção das categorias do protótipo SICLA_APF _____	61
Figura 10 – Janela de manutenção dos tipos de complexidade do protótipo SICLA_APF _____	62
Figura 11 – Janela de manutenção da contagem do protótipo SICLA_APF _____	62
Figura 12 – Janela de manutenção das funções da contagem do protótipo SICLA_APF _____	63

LISTA DE TABELAS

Tabela 1 - Tabela de complexidade funcional de ALI e AIE _____	30
Tabela 2 - Tabela de contribuição do ponto por função das funções do tipo dado. _____	30
Tabela 3 – Tabela de complexidade para entradas externas _____	34
Tabela 4 – Tabela de complexidade para saídas externas e consultas externas _____	34
Tabela 5 – Tabela de contribuição dos pontos por função das funções de transação _____	35
Tabela 6 – Tabela de níveis de influência das características gerais de sistema _____	38
Tabela 7 – Tabela de funções do tipo dado da contagem de teste _____	50
Tabela 8 – Tabela de complexidade das funções do tipo transação da contagem de teste _____	51
Tabela 9 – Resultado dos PF das funções do tipo dado da contagem teste _____	52
Tabela 10 – Resultado dos PF das funções do tipo transação da contagem teste _____	53

LISTA DE ABREVIATURAS E SIGLAS

APF	Análise de Pontos por Função
AIE	Arquivo de Interface Externa
ALI	Arquivo Lógico Interno
BFPUG	<i>Brazilian Function Point Users Group</i>
CGS	Características Gerais de Sistema
COCOMO	<i>Constructive cost model</i>
COCOMO	<i>Constructive cost model II</i>
II	
ERP	<i>Enterprise Resource Planning</i>
FPA	<i>Function Point Analysis</i>
FSM	<i>Functional Size Measurement</i>
IEC	<i>International Engineering Consortium</i>
IFPUG	<i>International Function Point Users Group</i>
ISBSG	<i>International Software Benchmarking Standards Group</i>
JTC1	<i>Joint Technical Committee One</i>
ISO	<i>International Organization for Standardization</i>
LOC	<i>Lines of code</i>
PMI	<i>Project Management Institute</i>
SC7	<i>Sub-Committee Seven</i>
TI	Tecnologia da Informação
VAF	Valor do Fator de Ajuste
WG12	<i>Working Group 12</i>

SUMÁRIO

INTRODUÇÃO	11
1 A NECESSIDADE DE MEDIR SOFTWARE	14
1.1 Por que medir software	14
1.2 Qual medida utilizar?	15
1.3 Metodologia que será estudada	16
1.4 Empresa utilizada como objeto de estudo	17
2 METODOLOGIAS DE MEDIÇÃO	18
2.1 Análise de Pontos por Função	18
2.2 <i>Constructive Cost Model</i>	19
2.3 MARK II	20
2.4 <i>Common Software Measurement International Consortium</i>	20
2.5 <i>Netherlands Software Metrics Users Association</i>	20
2.6 Análise de Pontos por Função <i>Fuzzy</i>	21
2.7 Instituições voltadas à APF	21
2.7.1 <i>International Function Point Users Group</i>	21
2.7.2 <i>Brazilian Function Point Users Group</i>	22
2.7.3 <i>International Software Benchmarking Standards Group</i>	22
2.8 Por que utilizar APF?	22
2.9 Processo da análise de pontos por função	23
2.9.1 <i>Determinação</i> do tipo de contagem	24
2.9.2 Fronteira de aplicação e escopo da contagem	25
2.9.3 Funções do tipo dado e transação	26
2.9.4 Pontos de função não-ajustados	35
2.9.5 Fator de ajuste e pontos de função ajustados	35
2.10 Ferramentas de medição	42
3 ESTUDO DE CASO	43
3.1 Apresentando a empresa estudada	43
3.2 Processos da Rech Informática	43
3.2.1 Processo principal – Relacionamento com mercado	44
3.2.2 Processo de apoio – Pesquisa e desenvolvimento	44
3.2.3 Processo de apoio – Controladoria e finanças	45
3.3 Problema a ser solucionado	45
3.4 Projeto utilizado como <i>case</i>	46
4 ANALISANDO A FERRAMENTA APF	48

4.1	Características da ferramenta APF	48
4.2	Contagem de teste	49
4.2.1	Contagem manual	49
4.2.2	Contagem utilizando a ferramenta APF	53
4.2.3	Contagem utilizando o protótipo SICLA-APF	54
4.3	Por que desenvolver a própria ferramenta?	54
5	PROTÓTIPO SICLA-APF	56
5.1	Metodologia de análise	56
5.2	Estrutura de dados	56
5.3	Funcionamento do protótipo	59
	CONCLUSÃO	65
	REFERÊNCIAS BIBLIOGRÁFICAS	66

INTRODUÇÃO

Com o crescimento econômico e aumento da concorrência, nos mais variados segmentos de mercado, podemos observar que as organizações buscam diferenciais nos seus processos que possam fazer com que o seu negócio esteja sempre à frente da concorrência. Entre estes diferenciais podemos destacar os custos do processo produtivo, a incansável busca por custos mais baixos, sem deixar de lado a qualidade do produto produzido, em meio a um mercado cada vez mais exigente.

No ramo de prestação de serviços, mais especificamente na área de tecnologia da informação, a necessidade constante de aumento da qualidade e melhoria dos processos de desenvolvimento de software tem sido crucial para sobrevivência no mercado. Tem se observado nas empresas desenvolvedoras de software que estando os processos e papéis muito bem definidos, elas tem conseguido melhorar a produtividade e por conseqüência uma redução nos custos de produção.

A qualidade do software (Pressman, 2002) depende de um considerável nível de processo de desenvolvimento bem estruturado. Atualmente existem empresas que não possuem processos mapeados e bem definidos. Além disto, não adotam metodologias que auxiliem os gerentes de projeto e diretores no processo de planejamento dos seus projetos. Em virtude disto, acabam assumindo riscos estipulando prazos e custos baseando-se no “*feeling*”, no sentimento dos seus gerentes, ou em experiências anteriores. Os resultados nem sempre são satisfatórios, projetos com prazos estourados e com alto custo de desenvolvimento (Pressman, 2002). Esta situação pode complicar se observarmos o dia-a-dia das empresas de desenvolvimento de software, onde percebemos uma concorrência desenfreada de projetos a serem analisados e desenvolvidos, às vezes em um curtíssimo espaço de tempo.

Em virtude disto, podemos ver algumas empresas adotando modelos de qualidade de software (Koscianski, 2006), visando melhorar os seus processos. Visando a otimização de processos e por consequência a produtividade, podemos destacar a necessidade de metodologias para mensuração de prazos e custos de novos projetos, bem como a manutenção dos softwares já existentes. A análise de pontos por função (APF) estudada neste trabalho de conclusão, tem sido utilizada por organizações para atender esta necessidade. A APF é também considerada como uma prática chave para empresas que pretendem atingir o nível 2 de maturidade do CMMI (Côttes, 1998). O CMMI é um modelo de avaliação da maturidade dos processos, produtos e serviços das empresas. Foi desenvolvido pela Universidade de Carnegie Mellon a pedido do departamento de defesa dos Estados Unidos, sendo difundido em organizações ao redor do mundo. Conhecida internacionalmente como *Function Point Analysis* (FPA), esta metodologia foi criada por Allan Albrecht na empresa IBM no final da década de 70. A FPA tem como um dos objetivos medir a taxa de produtividade e o esforço no desenvolvimento de software com base na perspectiva do cliente, ou seja, o usuário final. A partir dos requisitos (Pressman, 2002) é feito o levantamento do que deve ser feito, para então chegar ao número de pontos por função a serem implementados. Em posse desta informação, são aplicadas fórmulas matemáticas para chegar ao resultado próximo do esforço que será necessário para produção do projeto, podendo ainda estimar o prazo e o custo com base nos recursos disponíveis da organização. Uma característica interessante desta metodologia é que ela independe de qualquer ferramenta ou técnica de desenvolvimento, porque a contagem é feita pela perspectiva do usuário. Podemos utilizar esta metodologia como:

- Um método para dimensionar as aplicações;
- Um método para estimar esforços, prazos e custos;
- Um método para quantificar produtividade e qualidade;
- Um fator de normalização para comparar software.

O uso da APF na mensuração de projetos ajuda também no momento de negociar com o cliente o valor a ser cobrado e o prazo para o desenvolvimento. Como a APF é baseada na perspectiva do usuário, a negociação se torna menos complicada para o usuário entender a complexidade e o esforço envolvido. Outro ponto na negociação com o cliente a ser observado é o emprego da palavra “custo”. Esta palavra pode criar uma restrição durante a

negociação, é recomendável empregar então o termo “investimento”, já que os projetos desenvolvidos normalmente têm como objetivo aperfeiçoar os processos do cliente. Podemos ressaltar também que o esforço empregado no desenho do projeto poderá ser ineficaz, se os requisitos não refletirem exatamente os anseios do cliente. Quanto ao prazo, estando o projeto bem desenhado e os pontos por função devidamente desdobrados, somando-se os tempos de produção de cada função, obteremos o prazo total do projeto. Com isso podemos acordar com o cliente a data de entrega do projeto e cronograma de treinamento caso necessário.

Percebe-se nas organizações que a taxa de produtividade no desenvolvimento é uma necessidade pontual apontada por diretores e gerentes. Este trabalho de conclusão tem como objetivo o estudo dos processos da empresa Rech Informática Ltda. Estabelecida em Novo Hamburgo/RS, a Rech Informática tem como processo principal a prestação de serviços na área de tecnologia da informação, utilizando como ferramenta de apoio o SIGER, ERP desenvolvido pela empresa. Com crescimento constante, a Rech Informática tem uma considerável demanda de implementações, e ainda tem necessidade de criar novos recursos para manter e conquistar novos clientes. Partindo deste pressuposto, a direção da empresa e colaboradores da área de planejamento de projetos aponta a complexidade de estimar prazos, custos e gerenciar a concorrência da demanda.

A proposta deste trabalho era analisar ferramentas de mensuração existentes e a construção de um protótipo que utiliza a metodologia APF. Este protótipo tem o objetivo de ser futuramente incorporado ao CRM da Rech, o SICLA. Para validação do protótipo proposto, foi utilizado como estudo de caso um projeto recentemente concluído na empresa. O projeto escolhido compreende novas funcionalidades do módulo de Contabilidade SIGER. O protótipo foi desenvolvido pela mesma ferramenta de desenvolvimento que a empresa utiliza no seu CRM, baseado na metodologia estruturada.

1 A NECESSIDADE DE MEDIR SOFTWARE

1.1 Por que medir software

As empresas fabricantes de software cada vez mais precisam aperfeiçoar seus processos para ganhar competitividade no mercado. O diferencial na hora de ganhar da concorrência pode estar em pequenos detalhes, como estimar prazos e custos de projetos. Estes pequenos detalhes podem tornar-se um problema se a empresa não tiver nenhuma técnica para estas questões (HAZAN, 2003).

Além de vencer a concorrência, resultado de conjunto de esforços da empresa, devemos observar que antes disso a empresa precisa que todos seus colaboradores tenham ciência dos processos da empresa e seus papéis dentro do processo. Segundo o *Project Management Institute* (PMI), um projeto é um empreendimento temporário posto em execução para criar um único produto ou serviço. Dentro dos projetos encontramos 3 passos comuns que são o planejamento, execução e controle. O planejamento visa traçar o os objetivos do projeto e o caminho a ser seguido, a execução trata do gerenciamento de pessoas e recursos para dar vida ao projeto. Por fim o controle, garantindo que o planejamento foi seguido, monitorando e medindo, identificando as variações durante a execução e quando necessário, tomar ações corretivas para que o planejamento seja seguido.

No planejamento, segundo Vasquez, Simões e Albert (2005, p.19):

[...] em sua fase inicial que compreende o levantamento de requisitos, ainda não há o conhecimento completo das características do produto que permita a apuração de sua futura dimensão. Nesse caso é necessário estimar.

No controle, segundo Vasquez, Simões e Albert (2005, p.21):

[...] é de suma importância que a definição de meios para a comparação do progresso real com o planejado seja parte do planejamento do projeto. O controle é uma das principais atividades envolvidas na gerência de projetos. Trazendo estes conceitos para o contexto de um projeto de desenvolvimento de sistemas, é possível ter algumas idéias interessantes na busca da resposta de porque medir software. Afinal, não se consegue controlar o que não se consegue medir.

Segundo HAZAN (2007), as razões para se medir software são:

- Indicar a qualidade do produto ao usuário;
- Avaliar a produtividade do processo;
- Melhorar a gerência de projetos e relacionamento com clientes;
- Formar uma *baseline* para estimativas e gerenciar contratos de software.

A partir desta afirmativa dos autores, torna-se claro a necessidade de medir software, tanto para mapeamento do tamanho da aplicação como o controle de sua execução. A mensuração deve ser aplicada no início do projeto e aplicada no decorrer da sua execução até a contagem final. Isto porque podem ocorrer problemas durante a execução como: troca de pessoal da equipe ou até alguns itens de garantia da qualidade propositalmente esquecidos para se atingir marcos de projetos. Nestes casos é que a equipe de controle de projetos entra em ação, fazendo com que tudo que foi previsto no projeto será cumprido, uma nova medição do projeto deve ser feita e com base nos resultados, tomar medidas, avaliar prazos e custos.

Outro motivo para adotar uma técnica de mensuração é a engenharia de processos que passou a estar presente nas grandes empresas, e a necessidade de processos de garantia de qualidade. Com o surgimento de certificações de qualidade ISO e a popularização do CMMI, as empresas que buscam implementá-los precisam entender processos e criar metodologias para desenvolvimento destes. Quando o assunto é planejar projetos na área de software, isto não poderia ser diferente afinal precisamos adotar uma metodologia para mensuração e não estipular através da experiência da equipe.

1.2 Qual medida utilizar?

Uma vez que concordamos da necessidade de medir software, nos deparamos com a pergunta: Qual medida utilizar? Considerando-se que o tamanho é uma característica a ser medida, a unidade mais lógica para medição seria o número de linhas de código. Estudantes

da área utilizam esta unidade ao comparar softwares desenvolvidos. Segundo Vasquez, Simões e Albert (2005), o uso de *lines of code* (LOC) para mensuração pode ser perigoso caso não for levado em consideração que:

- Não podem ser incluídos na contagem da quantidade de linhas de código os comentários, linhas em branco ou comandos nulos;
- Inclusão na contagem de diretrizes de compilação;
- Contagem de múltiplos comandos ou declarações em uma única linha como várias linhas, uma para cada comando ou declaração;
- Contagem de uma única linha nos casos em que um único comando ou declaração é expresso em múltiplas linhas;
- Inclusão na contagem de delimitadores de blocos de comandos, nos casos em que de fato haja mais de um comando, e a desconsideração de delimitadores de blocos de comandos nos casos em que sua utilização seja opcional.

A partir destes itens salientados pelos autores, podemos concluir que não há um padrão para contagem de linhas de código, porque a codificação muda de linguagem para linguagem de programação, bem como o emprego de comandos e a lógica utilizada para solução do problema varia de programador para programador. Outra dificuldade encontrada nesta técnica está em argumentar com o cliente esta medida, afinal o cliente não entende o que quer dizer linhas de código. No capítulo seguinte, abordaremos metodologias que surgiram em função desta problemática, no intuito de sanar um dilema enfrentado por quem desenvolve software: como estabelecer um padrão para mensurar o seu esforço e que seus clientes compreendam?

1.3 Metodologia que será estudada

A APF é uma das metodologias de mensuração estudada neste trabalho de conclusão. Além de servir como metodologia para mensurar e controlar projetos, a APF pode ser utilizada para cálculo da remuneração e avaliação da produtividade da equipe de desenvolvimento. Encontramos a APF também em licitações públicas na área da tecnologia

da informação (TI), onde os projetos das empresas candidatas são avaliados através desta metodologia (HAZAN, 2003).

1.4 Empresa utilizada como objeto de estudo

A Rech Informática, participante do Programa Gaúcho de Qualidade (PGQP), possui seus processos já bem definidos e busca melhoria na área de desenvolvimento. A empresa têm ciência das implementações e melhorias a serem feitas, possui uma considerável carteira de projetos já documentada e outra de projetos a documentar. A empresa tem uma lista da prioridade destes projetos, mas os colaboradores da área de desenvolvimento sentem-se carentes de um método eficaz que ajudem a ter noção de quando um projeto deve começar e quando possivelmente terminará. Com os gerentes da área de desenvolvimento podendo ter uma noção mais apurada, eles poderão organizar melhor a equipe de programadores para desenvolvimento dos projetos.

2 METODOLOGIAS DE MEDIÇÃO

2.1 Análise de Pontos por Função

A APF surgiu em 1975 na empresa IBM. Allan Albrecht na época tinha a incumbência de medir os projetos da empresa, na ocasião utilizavam a contagem de linhas de código. Allan percebeu que era complexo medir a partir desta técnica, por motivos já explanados anteriormente. Além de a técnica ser deficiente, a empresa possuía vários projetos desenvolvidos em diferentes linguagens de programação. Isto dificultava a análise da produtividade da empresa. A solução encontrada foi quebrar este paradigma, passaram a medir pela perspectiva do que a aplicação deveria fazer, e não como seria construída. Esta técnica tornou-se assim independente de linguagem de programação. Após a publicação desta técnica e dos sucessivos trabalhos feitos Capers Jones, houve uma crescente legião de adeptos da técnica. Em 1986 culminou com a fundação do *International Function Point Users Group* (IFPUG). Com o passar dos anos começaram a surgir variações da teoria de Albrecht, fazendo com que em 1990 o IFPUG lançasse a primeira versão do manual de práticas de contagem de pontos por função, com objetivo de padronizar a técnica. Esta metodologia do IFPUG é considerada como o padrão mais difundido no mercado de pontos por função. Está atualmente na versão 4.2.1, embora ainda tenham outras técnicas como o Mark II e COSMIC-FFP (Vasquez, Simões e Albert, 2005).

No final de 1992 havia diversos métodos de mensuração do tamanho funcional *Functional Size Measurement* (FSM). Estes métodos surgiram de diferentes entendimentos da metodologia criada originalmente. Com o objetivo de manter uma padronização da metodologia, usuários da Holanda, Inglaterra, Austrália e Estados Unidos formaram o *Working Group 12* (WG12). O WG12 está subordinado ao *Sub-Committee Seven* (SC7) do *Joint Technical Committee One* (JTC1). O JTC1 por sua vez, foi estabelecido pela

International Organization for Standardization (ISO), em conjunto com o *International Engineering Consortium (IEC)*. O resultado do trabalho conjunto culminou com a criação da norma 14.143. Esta norma é composta pelas seguintes partes:

14.143-1: Definição de conceitos;

14.143-2: Avaliação da conformidade de métodos de medição de software com relação ao padrão ISO/IEC 14.143-1;

14.143-3: Verificação de um método de medição de tamanho funcional;

14.143-4: Modelo de referência para medição de tamanho funcional;

14.143-5: Determinação de domínios funcionais para uso com medição de tamanho funcional.

No final de 2002 a APF foi aprovada e recebeu a denominação ISO/IEC 20296:2002. Entretanto, o padrão da APF aprovado pela ISO vai somente até o estágio da determinação dos pontos de função não ajustados. O restante do processo conforme o manual do IFPUG contém itens que são considerados não aderentes ao padrão de medição funcional da ISO (Vasquez, Simões e Albert, 2005).

Podemos concluir que a padronização ISO foi de fundamental importância para que a APF fosse difundida no mundo inteiro, aliado ao trabalho dos membros do IFPUG e a participação de empresas de prestígio como a Unisys aqui no Brasil.

2.2 *Constructive Cost Model*

Em 1981, na tentativa de criar um modelo para medir, Dr. Barry Boehm lançou o *Constructive cost model (COCOMO)*. O COCOMO baseava-se na técnica LOC para estimar esforço, prazo e custo. Este modelo pode ser complicado de utilizar, porque antes de realmente codificar o programa fonte, não sabemos ao certo quantas linhas de código são necessárias para atender determinada funcionalidade, por exemplo. Hoje em dia esta questão fica mais complexa, se observarmos as ferramentas de programação atuais onde fica difícil a abstração.

2.3 MARK II

MARK II ou MK II foi criada por Charles Symons em meados da década de 80. MK II era baseado também na proposta de Albrecht. Ficou conhecida em 1988, após a publicação na revista *Transactions on software engineering* (IEEE). Esta técnica atualmente é de domínio público. A organização responsável pela manutenção do padrão é a UKSMA, associação de métricas do Reino Unido. A UKSMA segue a mesma linha do IFPUG. Esta técnica não se difundiu no mundo devido ao seu método de aplicação ser restrito ao Reino Unido.

2.4 *Common Software Measurement International Consortium*

Na universidade de Quebec, Canadá, os pesquisadores Denis St-Pierre, Marcela Maya, Alain Abran, Jean Marc Desharnais e Pierre Bourque criaram em 1997 um método de medição funcional para sistemas de tempo real. Este método ficou conhecido como FFP – *Full Function Points*. Este grupo de pesquisadores continuou com a pesquisa, na busca da evolução do método. O objetivo era poder aplicar o método em sistemas tradicionais.

Um outro grupo de pesquisadores em 1998, liderados por Alain Abran e Charles Symons, pai do MARK II, criaram o *Common Software Measurement International Consortium* (COSMIC). Este grupo visou desenvolver um novo método de medição, a partir das melhores características dos métodos existentes e que incorporasse novas idéias. Em 2000, foi apresentado o resultado deste estudo. Ficou batizado de COSMIC-FFP, que era considerado um refinamento do FFP.

2.5 *Netherlands Software Metrics Users Association*

A NESMA associação de métricas da Holanda é considerada um dos maiores grupos de usuários de pontos por função da Europa. Foi fundada em Maio de 1989, com nome inicial de NEFPUG – *Netherlands Function Point Users Group*. A entidade inspirou-se nas características e objetivos do IFPUG, inclusive a entidade mantém relacionamento muito próximo. Apesar desta proximidade, a NESMA mantém seu próprio manual de contagem, criado em 1990 baseando-se no manual do IFPUG. Segundo Vasquez, Simões e Albert, os manuais utilizam a mesma filosofia, conceitos, termos e regras. A diferença está nas diretrizes, podemos utilizar as duas técnicas de medição em projetos e chegar a resultados

bem próximos. A diferença maior entre as técnicas está na contagem em projetos de melhoria de software, onde as entidades possuem abordagens distintas para aplicação da análise de pontos por função.

2.6 Análise de Pontos por Função *Fuzzy*

A análise de pontos por função *fuzzy* é uma teoria que surgiu com intuito de fornecer números mais precisos na contagem dos pontos por função. Esta teoria busca construir uma estrutura quantitativa, procurando capturar possíveis imprecisões do conhecimento humano. No entanto, ainda há resistência quanto à utilização desta teoria.

2.7 Instituições voltadas à APF

2.7.1 *International Function Point Users Group*

O IFPUG foi fundado em 1986, após uma crescente legião de adeptos da técnica de Albrecht. O IFPUG é uma entidade sem fins lucrativos, sendo composta por usuários e empresas de diversos países. A entidade visa o correto uso da metodologia de análise de pontos por função, promovido pelo trabalho voluntário dos seus membros. O IFPUG promove várias iniciativas como:

- Conferência anual, seminários e *workshops* educacionais;
- Certificação profissional;
- Comitês e grupos de trabalho com objetivo de manter o manual de práticas entre outras atividades a fim da entidade.

O manual de práticas do IFPUG é gratuito para os seus membros. Para quem não é membro, é possível consegui-lo, porém somente mediante sua compra (Vasquez, Simões e Albert, 2005).

2.7.2 Brazilian Function Point Users Group

Brazilian Function Point Users Group (BFPUG) foi fundado em 1988, é o *chapter* do IFPUG no Brasil. O termo *chapter* é utilizado para as associações locais do IFPUG para difundir a metodologia. O BFPUG conta com centenas de associados, desde estudantes a desenvolvedores e gerentes de sistemas. A entidade promove em seu site diversos eventos, inclusive um fórum de discussão gratuito. A APF começou a prosperar no Brasil somente no início da década de 90 com apoio da empresa Unisys (HAZAN, 2003). A partir disso, foram feitos vários encontros sobre o assunto, inclusive com a presença de palestrantes internacionais. O uso da metodologia se consolidou depois do interesse de entidades envolvidas no desenvolvimento de software por modelos de qualidade e maturidade (ISO e CMMI), aumentando assim o interesse das pessoas no assunto (Vasquez, Simões e Albert, 2005).

2.7.3 International Software Benchmarking Standards Group

O *International Software Benchmarking Standards Group* (ISBSG) é uma entidade sem fins lucrativos, mantido por diversas organizações de métricas de software do mundo. A missão do ISBSG é manter um repositório público de métricas de projetos. O ISBSG auxilia os usuários de pontos por função nas estimativas de projetos, produtividade, análise de riscos e *benchmarking*. Seu repositório de dados contém mais de 3.000 projetos de software de vários países. Regularmente o ISBSG publica uma análise estatística detalhada do seu repositório, chamada de “*The Software Metrics Compendium*”. Seu repositório é constantemente atualizado com novos estudos de caso e qualquer organização pode contribuir. O ISBSG coleta os dados e mantém a confidencialidade dos dados fornecidos e o anonimato da organização (Vasquez, Simões e Albert, 2005).

2.8 Por que utilizar APF?

A APF surgiu para quebrar o paradigma de mensuração do tamanho do produto pela quantidade de linhas de código, bem como a dependência de tecnologia a ser empregada. O foco desta metodologia compreende na mensuração da aplicação pelo que ela faz, pela perspectiva do usuário e não como a aplicação foi construída. Utilizando como unidade de medida o ponto por função, esta metodologia independe de tecnologia empregada. A

contagem dos pontos por função é baseada na avaliação dos requisitos do usuário, sendo que ela representa exclusivamente o tamanho funcional da aplicação. O ponto por função por sua vez serve de base para que com outras variáveis possa ser calculado o esforço, prazo e o custo.

Segundo HAZAN (2007), APF é:

[...] A Análise de Pontos por Função (APF) é um método padronizado para a medição de projetos de desenvolvimento de software, visando estabelecer uma medida de tamanho, em Pontos de Função (PF), considerando a funcionalidade implementada, sob o ponto de vista do usuário. [...]

Na parte de projetos podemos dizer que a APF pode servir de suporte na análise de produtividade e qualidade, em conjunto com métricas de esforço, defeitos e o custo. Podemos concluir ainda outros benefícios como:

- Apoio ao gerenciamento de escopo de projetos, um dos desafios do gerente de projeto;
- Complementa o gerenciamento de requisitos, ao analisar a qualidade do levantamento de requisitos;
- Um meio de estimar custos e recursos nos processos e projetos;
- Uma forma para fundamentar a negociação de contratos, estabelecendo uma unidade tangível para o cliente, o ponto por função, estabelecendo um preço por esta unidade;
- Uma forma de normalização para comparação de software, ou ainda comparação de produtividade na utilização de tecnologias diferentes.

Além destes motivos, a APF possui certificação ISO que é uma garantia consistente de que esta metodologia é eficaz e totalmente madura para ser adotada pelas empresas.

2.9 Processo da análise de pontos por função

O processo da APF segue em linhas gerais o fluxo da figura abaixo.

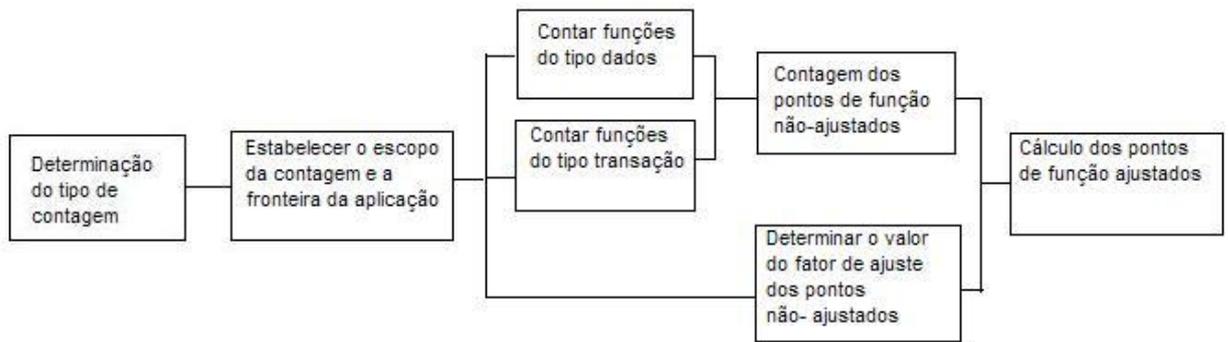


Figura 1 – Fluxo do processo de contagem de pontos por função.

Fonte: Vasquez, Simões e Albert (2005)

Este processo pode ser aplicado tanto em projetos já concluídos como nos projetos ainda em fase de levantamento de requisitos. Para o sucesso da contagem o levantamento de requisitos tem papel fundamental para a APF, afinal o objetivo é medir a partir da perspectiva do usuário. É ele quem define os requisitos que o projeto deve ter (Vasquez, Simões e Albert, 2005). Devemos ressaltar que a palavra usuário na APF tem um conceito mais abrangente. Seu conceito não está apenas associado à pessoa física que utiliza software. Usuário é qualquer pessoa ou coisa que interaja (envia ou recebe dados) com a aplicação ou especifique seus requisitos. Por exemplo, um usuário pode tanto ser um operador (pessoa física), como uma aplicação ou um hardware que interaja com a aplicação a ser medida. Desta forma a APF pode ser aplicada também a softwares em que não tenham interface com o usuário final (pessoa física). (Vasquez, Simões e Albert, 2005). Segundo HAZAN (2003): “Usuário pode ser qualquer pessoa, dispositivo ou sistema que se comunica ou interage com a aplicação”.

2.9.1 Determinação do tipo de contagem

O processo de contagem de pontos por função é iniciado com a determinação do tipo de contagem. A primeira questão que devemos responder é: o que deve ser contado? Ao determinar o tipo de contagem, devemos considerar que a contagem pode ser aplicada tanto a projetos quanto aplicações. Basicamente o tipo de contagem divide-se em três tipos: projeto de desenvolvimento, projeto de melhoria e aplicação. No projeto de desenvolvimento o número de pontos por função mede a funcionalidade do projeto a ser entregue. Isto quer dizer que a contagem não compreende apenas o software em si. Outras ações a serem feitas durante a sua execução também devem ser contadas, como uma migração de base de dados, por

exemplo. Durante a concepção e execução de um projeto, toda medida realizada é apenas uma estimativa do que está sendo feito. É bastante comum durante o desenvolvimento de projetos a equipe encontrar funcionalidades não medidas no planejamento do projeto. Conforme o projeto é produzido e medido durante o processo, chegamos mais perto da sua medida final, permitindo assim uma avaliação final pela equipe do projeto. O projeto de melhoria, diferente do projeto de desenvolvimento, parte de um software já produzido. A contagem de pontos por função em um projeto de melhoria compreende medir as funções novas, alteradas e eliminadas. Caso houver migração de base de dados, as funções para a migração também devem ser medidas. Em uma aplicação o número de pontos por função mede a funcionalidade por uma aplicação instalada. Conhecida por pontos de função instalados ou *baseline*, esse número fornece uma medida atual da funcionalidade pelo usuário de uma aplicação. Ele começa ao final da contagem do projeto de desenvolvimento e atualizado no término de todo projeto de melhoria, o qual altera a funcionalidade da aplicação.

2.9.2 Fronteira de aplicação e escopo da contagem

Estabelecer a fronteira da aplicação é delimitar onde começa e onde termina a medição dos pontos por função. Esta etapa é considerada uma das mais importantes do processo de contagem. Se a fronteira de aplicação não estiver bem definida, há risco do restante da contagem não refletir o real tamanho da aplicação, comprometendo todo o trabalho. Segundo o IFPUG, as regras para determinação da fronteira da aplicação são:

- Identificar a fronteira da aplicação pela perspectiva do usuário, no que ele pode entender e descrever;
- A fronteira entre as aplicações deve ser identificada na separação das funções conforme os processos do negócio e não pelo aspecto tecnológico;
- Em projetos de melhoria, a fronteira estabelecida no início do projeto deve estar em concordância com a fronteira já estabelecida para a aplicação.

Segundo Vasquez, Simões e Albert, 2005, as seguintes dicas podem ajudar a identificar a fronteira de aplicação:

- Obter a documentação do fluxo de dados e desenhar em volta uma fronteira para destacar quais são partes internas e externas a aplicação;

- Observar como os dados são armazenados;
- Identificar áreas funcionais como entidades e processos;
- Observar o gerenciamento da aplicação se é desenvolvida ou mantida na sua totalidade por uma equipe distinta;
- Observar se o software possui ordens de serviço específicas e independentes.

Abaixo podemos observar uma ilustração de fronteira de aplicação:

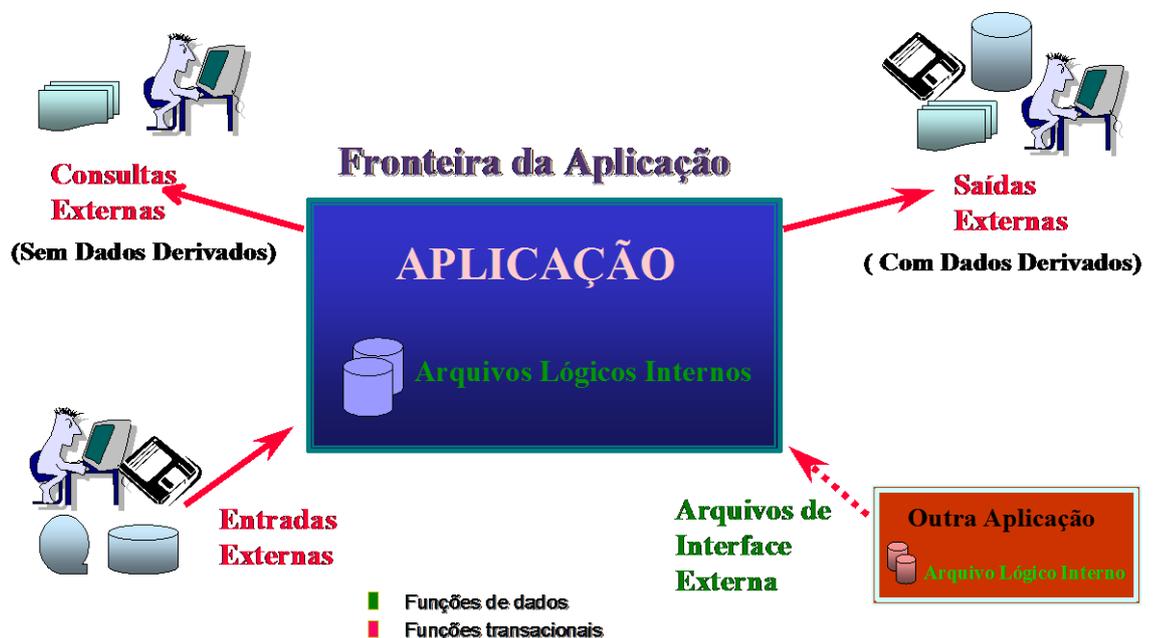


Figura 2 – Fronteira de aplicação

Fonte: Uma aplicação da APF nas estimativas de projetos web (Hazan, 2003)

O objetivo da definição do escopo de contagem consiste em definir quais funções serão incluídas na contagem: se abrange uma ou mais aplicações, podendo compreender todas as funcionalidades; apenas as funcionalidades em uso pelo usuário ou funcionalidades específicas. Segundo HAZAN (2003), o escopo da contagem deve incluir todos os componentes necessários para apresentar as necessidades do negócio.

2.9.3 Funções do tipo dado e transação

Funções do tipo dado são aquelas projetadas para atender ao usuário no que se refere à base de dados da aplicação. As funções estão classificadas em Arquivo Lógico Interno

(ALI) e Arquivo de Interface Externa (AIE). Antes de detalhar estas classificações, precisamos entender o que significa o termo arquivo. Arquivo neste caso não se refere necessariamente a um arquivo do sistema operacional, tem um sentido mais abrangente. Arquivo pode ser um grupo de dados relacionados logicamente e reconhecido pelo usuário. Lembramos que o usuário como definido anteriormente, não se refere necessariamente a uma pessoa física. Pode ocorrer de um ou mais arquivos ou até tabelas de um banco ser tratados como um único arquivo pela APF. A forma como a aplicação é implementada para o armazenamento dos dados não é relevante para determinar as funções do tipo dado.

As funções do tipo transação como o próprio nome sugere, refere-se aos processos da aplicação, como a informação é processada e através de quais funcionalidades. As funções do tipo transação estão classificadas em: Entrada Externa (EE), Saída Externa (SE) e Consulta Externa (CE). Estas classificações serão abordadas a seguir.

2.9.3.1 Funções do tipo dado

Arquivo lógico interno por definição é considerado um grupo de dados, identificável pelo usuário, estando logicamente relacionado e mantido dentro da fronteira da aplicação. A principal função de um ALI é obviamente armazenar dados adicionados, modificados ou excluídos pela aplicação através de funções do tipo transação (Vasquez, Simões e Albert, 2005). Segundo HAZAN (2003), ALI são grupos de dados ou informações de controle especificado pelo usuário logicamente relacionado, cuja manutenção é efetuada dentro da fronteira da aplicação. O objetivo do ALI é armazenar dados mantidos através de um ou mais processos da aplicação contada. Para exemplificar o que são considerados arquivos lógicos internos, podemos considerar uma tabela de usuários da aplicação, com campos conhecidos como usuário, *login* e senha de acesso. Segundo (Vasquez, Simões e Albert, 2005), não são considerados ALI:

- Arquivos temporários de classificação;
- Arquivos que existem apenas durante a execução da aplicação, sem armazenar informações;
- Arquivos de *backup*;
- Arquivos gerados para processamento de outras aplicações;

- Arquivos de índices;
- Visões (*views*).

Arquivo de interface externa por definição é considerado um grupo de dados identificável pelo usuário, logicamente relacionado e mantido fora da fronteira da aplicação. A finalidade de um AIE é utilizar dados para referência no processamento dentro da fronteira de aplicação medida, porém o AIE deve obrigatoriamente ser mantido por outra aplicação. (Vasquez, Simões e Albert, 2005). Segundo HAZAN (2003): AIE são grupos de dados ou informações de controle especificado pelo usuário logicamente relacionado, cuja manutenção é efetuada dentro da fronteira de outra aplicação. O objetivo do AIE é armazenar dados referenciados através de um ou mais processos da aplicação sendo contada. Um exemplo de AIE, conforme o projeto de estudo, observamos na função de inclusão de lançamentos. Nesta função temos o arquivo de plano de contas, cuja manutenção é feita em outra aplicação do sistema de contabilidade. Contudo, é necessário utiliza-lo para associar um lançamento contábil a uma conta contábil existente. Podemos concluir então que a diferença entre ALI e AIE é que o primeiro está dentro da fronteira da aplicação a ser medida já o outro não, logo um AIE é um ALI de outra aplicação. Dentro de cada função do tipo dado precisamos classificar os tipos de dados e os tipos de registro.

Um tipo de dado é um campo único, não repetido e reconhecido pelo usuário. Para contagem de um tipo de dado devem ser observadas as seguintes regras:

- Deverá ser considerado um tipo de dado para cada campo único reconhecido pelo usuário e não repetido, utilizado por um ALI ou AIE por meio de um processo. No caso de um campo de data, mesmo que existam campos separados para dia, mês e ano, devemos conta-los como um tipo de dado;
- Caso o ALI ou AIE possua duas aplicações ou mais que o mantenham ou referenciam, devem ser contados apenas os campos utilizados pela aplicação contada;
- Deverá ser considerado um tipo de dado para cada campo solicitado pela aplicação para estabelecer um relacionamento com outro ALI ou AIE.

No projeto de estudo, na função de inclusão de lançamentos, existe uma verificação do usuário se este está autorizado a incluir lançamentos. Para esta verificação é acessado o arquivo de usuários (AIE) e os campos acessados são: código de usuário e o campo que indica se permite inclusão de lançamentos. Neste caso são contados 2 tipos de dados para o AIE, apesar de existirem outros campos, como nome e outros controles de acesso, mas para esta aplicação somente devemos considerar 2 tipos de dados e 1 tipo de registro. Assim cada campo utilizado pela aplicação é considerado como tipo de dado e deve ser considerado na contagem para obter a complexidade do ALI ou AIE ao qual o campo pertence.

Tipo de registro é um subgrupo de dados de um ALI ou AIE, reconhecido pelo usuário. Para contagem de um tipo de registro devemos observar as seguintes regras:

- Deve ser considerado um tipo de registro para subgrupo de dados de um ALI ou AIE;
- Caso não identificado nenhum subgrupo de dados, deve ser considerado um tipo de registro para cada ALI ou AIE.

No projeto estudado, um lançamento contábil possui tipos de dados como: Conta, data, valor, documento, histórico e complemento do lançamento. Estes tipos de dados estão armazenados logicamente em 2 arquivos distintos, porém são considerados como um grupo de dados do lançamento, logo, devemos considerar apenas 1 tipo de registro.

Na contagem de pontos por função de ALI e AIE devemos apurar os tipos de dados e tipos de registros para então classificarmos conforme a sua complexidade funcional. A complexidade funcional está dividida conforme o IFPUG em três tipos: alta, média e baixa conforme tabela abaixo:

	Tipos de dados			
		< 20	20 – 50	> 50
Tipos de registros	1	Baixa	Baixa	Média
	2 – 5	Baixa	Média	Alta
	> 5	Média	Alta	Alta

Tabela 1 - Tabela de complexidade funcional de ALI e AIE

Fonte: IFPUG

Conforme a tabela acima, um ALI com 60 tipos de dados em 3 tipos de registros é considerado como complexidade alta. Quanto ao projeto de estudo, o arquivo de lançamentos (ALI) possui 45 tipos de dados em 1 tipo de registro, a complexidade neste caso é considerada baixa.

Após a determinação dos tipos de dados, tipos de registros e determinadas a complexidade, o próximo passo é classificarmos a complexidade encontrada dentro da tabela de contribuição, conforme abaixo:

TIPO DE FUNÇÃO	Baixa	Média	Alta
ALI	7 PF	10 PF	15 PF
AIE	5 PF	7 PF	10 PF

Tabela 2 - Tabela de contribuição do ponto por função das funções do tipo dado.

Fonte: IFPUG

A identificação dos arquivos lógicos internos e arquivos de interface externa podem parecer bem próximos da interpretação de um modelo ER, mas devemos estar atentos, porque podem ocorrer interpretações equivocadas em determinadas situações. Na contagem também é necessário conseguirmos distinguir os requisitos de armazenamento funcionais e não funcionais da aplicação, que estão classificados como dados de código ou metadados, dados de referência e dados de código.

Dados de código ou metadados nunca devem ser considerados como arquivos lógicos. Não devem ser considerados na contagem porque são implementações de requisitos técnicos e por isso não devem influenciar no tamanho funcional da aplicação. Um exemplo de dados de código podemos observar em uma tabela de siglas de estado ou qualquer outra tabela que raramente tenha seu conteúdo modificado. Dados de referência são informações no nível de regra de negócio da aplicação, armazenados em arquivos, como por exemplo, regras e cálculos de incidência de impostos em uma nota fiscal. Existem também outras entidades que não são consideradas como arquivos lógicos:

- Arquivos de índices;

- Arquivos com dados consolidados ou visões;
- Arquivos temporários ou de classificação;
- Entidades de ligação.

2.9.3.2 Função do tipo transação

Funções do tipo transação são processamentos realizados pela aplicação com objetivo de atender requisitos funcionais apontados pelo usuário. As funções do tipo transação estão divididas em três tipos: EE, SE e CE.

EE são processamentos de dados recebidos de fora da fronteira de aplicação. Seu objetivo é a manutenção de um ou mais arquivos lógicos internos e/ou modificar o comportamento da aplicação (Vasquez, Simões e Albert, 2005). Segundo HAZAN (2003): EE é um processo elementar que processam dados ou informações de controle que vem do lado de fora da fronteira de aplicação. Tem como objetivo manter um ou mais ALI e/ou alterar o comportamento do sistema. Podemos exemplificar EE a partir de uma janela de manutenção de uma tabela qualquer onde tenha opção de incluir, alterar ou excluir registros da tabela. Neste exemplo devemos contar três entradas externas. Este exemplo também podemos observar no projeto estudado, onde existem funções de inclusão, alteração e exclusão de lançamentos. Outro exemplo é a atualização de um cadastro de cliente com o seu limite de crédito, após emissão de uma nota fiscal de venda. Por outro lado, telas de *login* de sistema e menus, por exemplo, não são consideradas entradas externas.

Saídas externas são processamentos de dados que são enviados para fora da fronteira de aplicação. Seu objetivo é demonstrar informação através de um processamento lógico, não apenas a exibição de dados. Nesta lógica de processamento, por definição deve conter algum cálculo, ou criar dados derivados (Vasquez, Simões e Albert, 2005). Segundo HAZAN (2003): SE é um processo elementar que enviam dados ou informação de controle para fora da fronteira da aplicação. Tem como objetivo principal apresentar a informação para o usuário através de processamento lógico adicional a recuperação de dados ou informação de controle. O processamento deve conter no mínimo uma fórmula matemática ou criar dados derivados, ou alterar comportamento da aplicação ou manter ALI. Como exemplo comum de uma SE, podemos considerar um relatório de estatística de vendas de um determinado período, com totalização e percentual de vendas de cada representante em relação ao total vendido.

CE são processos de simples recuperação e exibição de dados armazenados para fora da fronteira da aplicação. Seu objetivo é a demonstração de dados através da simples recuperação das informações de ALIs e/ou AIEs e obrigatoriamente não pode conter nenhum tipo de cálculo, do contrário trata-se de uma SE (Vasquez, Simões e Albert, 2005). Segundo HAZAN (2003), CE é um processo elementar que envia dados ou informação de controle para fora da fronteira de aplicação. Tem como objetivo principal apresentar informação para o usuário através da recuperação de dados ou informação de controle de um ALI ou AIE. O processamento lógico não contém fórmulas matemáticas ou cálculos e não cria dados derivados. Além disso, não mantém ALI durante o processamento nem altera o comportamento do sistema. Como exemplos de CE temos telas de ajuda (*help*), gráficos de informações simples, telas simples de *login*; menus gerados dinamicamente baseados na configuração da aplicação.

Para chegarmos à contribuição do ponto por função das funções do tipo transação, precisamos observar primeiro o processo de contagem conforme figura abaixo:

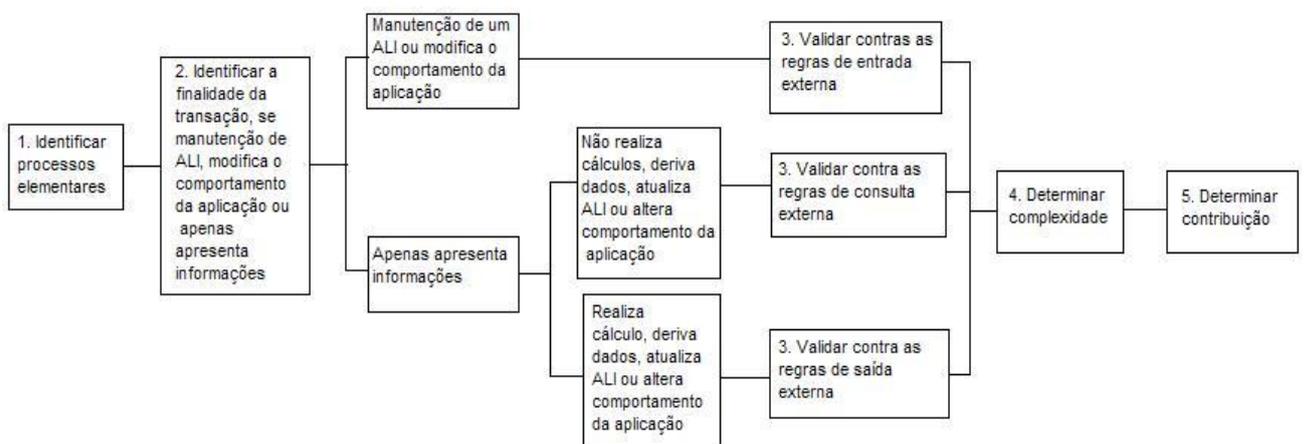


Figura 3 – Processo de contagem de funções do tipo transação

Fonte: Vasquez, Simões e Albert (2005)

No primeiro quadro, mais a esquerda da figura anterior, identificar os processos elementares, compreende-se por processo elementar a menor fração significativa para o usuário final. Esta definição é considerada uma questão chave na contagem de funções do tipo transação. Outra definição importante são as informações de controle, que são dados que influenciam um processo elementar. Este processo especifica o que, quando ou como os dados devem ser processados pela aplicação. Podemos dar o exemplo do *Internet Explorer*, o qual possui informações de controle para o funcionamento do *browser* durante a navegação entre

as páginas da *internet* (Vasquez, Simões e Albert, 2005). Assim como nas funções do tipo dado, devemos também classificar as funções do tipo transação quanto a sua complexidade funcional na aplicação. Cada EE, SE e CE deve ser classificada em alta, média ou baixa, de acordo com o número de arquivos referenciados e quanto ao número de tipo de dado (IFPUG). Um arquivo referenciado é um ALI lido ou atualizado pela função do tipo transação ou ainda pode ser um arquivo de interface externa lido pela mesma função (Vasquez, Simões e Albert, 2005). Tipo de dado como vimos anteriormente, é um campo único e reconhecido pelo usuário. Para contar um arquivo referenciado devemos observar as seguintes regras:

- Contar um arquivo referenciado para cada ALI que é atualizado ou lido pela função;
- Contar um arquivo referenciado para cada AIE lido pela função;
- Embora o ALI e/ou AIE processar mais de um tipo de registro, deverá ser contado apenas uma vez;

Devemos considerar também que:

- Somente devemos contar como arquivo referenciado arquivos lógicos internos e arquivos de interface externa;
- Mesmo que o ALI e/ou AIE possa fazer várias leituras, deve ser contado apenas uma vez.

Para contagem dos tipos de dados, devemos observar as seguintes regras:

- Contar um tipo de dado para cada campo não repetido e reconhecido pelo usuário, que entra ou sai pela fronteira de aplicação e utilizado no processo. Não importa quantas vezes o campo entra ou sai da fronteira da aplicação, mesmo assim deve ser contado apenas uma vez;
- Não devem ser contados campos que durante o processo são derivados da aplicação ou recuperados de um ALI ou AIE, e que não atravessam a fronteira da aplicação;
- Troca de mensagens que sai da fronteira da aplicação, como tratamento de exceções, mensagem de término de processamento, etc.;

Não são considerados tipos de dados:

- Literais;
- Variáveis de paginação ou campos automáticos gerados pela aplicação.

Concluído o levantamento do número de arquivos referenciados e tipos de dados de cada EE, SE e CE, devemos classificar conforme as tabelas abaixo:

Arquivos referenciados	Tipos de dados			
		< 5	5 – 15	> 15
< 2	Baixa	Baixa	Média	
2	Baixa	Média	Alta	
> 2	Média	Alta	Alta	

Tabela 3 – Tabela de complexidade para entradas externas

Fonte: IFPUG

Uma EE que possui 1 arquivo referenciado, sendo 10 tipos de dados, é considerado como complexidade baixa.

Arquivos referenciados	Tipos de dados			
		< 6	6 – 19	> 19
< 2	Baixa	Baixa	Média	
2 – 3	Baixa	Média	Alta	
> 3	Média	Alta	Alta	

Tabela 4 – Tabela de complexidade para saídas externas e consultas externas

Fonte: IFPUG

Uma SE de 3 arquivos referenciados e 12 tipos de dados, é considerado complexidade média. Após o enquadramento de cada função do tipo transação conforme as tabelas anteriores, basta calcularmos a sua contribuição conforme a tabela abaixo:

Tipo de função	Baixa	Média	Alta
EE	3 PF	4 PF	6 PF
SE	4 PF	5 PF	7 PF
CE	3 PF	4 PF	6 PF

Tabela 5 – Tabela de contribuição dos pontos por função das funções de transação

Fonte: IFPUG

Uma EE de complexidade média devemos considerar 4 pontos por função, conforme tabela do IFPUG.

2.9.4 Pontos de função não-ajustados

Uma vez apuradas as funções do tipo dado e transação, estas devem ser analisadas e classificadas quanto a sua complexidade em três níveis: alta, média e baixa. A complexidade de uma função do tipo dado compreende em quantos campos e registros são necessários para armazenar a informação. A função de transação por sua vez compreende no número de tipos de dados e arquivos envolvidos. A escala de complexidade é atribuída conforme uma tabela que trata dos pontos de função não-ajustados. O resultado final deste passo é a soma da contagem da complexidade das funções, chamada de pontos de função não-ajustados.

2.9.5 Fator de ajuste e pontos de função ajustados

Esta fase tem como objetivo estabelecer um fator de ajuste para a soma dos pontos de função não-ajustados, baseando-se em 14 critérios de influência. O ajuste fica em média 35% de acordo com as 14 características de influência, definidas pelo IFPUG. Esta fase não está compreendida no padrão ISO/IEC de medição funcional, o IFPUG tornou este passo opcional na APF para poder se adequar à norma. A última etapa da APF é calcular os pontos de função ajustados, o qual em posse dos pontos por função não-ajustados e o fator de ajuste definido, é aplicada uma forma matemática, específica para cada tipo de contagem (projeto de desenvolvimento, melhoria e aplicação).

Conforme o IFPUG, para determinação do valor do fator de ajuste (VAF) devemos considerar 14 características gerais de sistema (CGS). Para cada característica devemos determinar um nível de influência na aplicação. As características gerais de sistema para determinação do valor do fator de ajuste dos pontos por função não-ajustados são:

- Comunicação de dados: Refere-se ao nível em que a aplicação comunica-se diretamente com o processador. Os dados são enviados e recebidos por meio de recursos de comunicação, como terminais conectados localmente à unidade de controle e protocolo de comunicação;
- Processamento distribuído: Refere-se ao nível de transferência de dados que a aplicação faz entre seus componentes dentro da fronteira de aplicação;
- Performance: Refere-se ao nível de tempo de resposta e taxa de transações que influenciam o desenvolvimento da aplicação;
- Configuração altamente utilizada: Refere-se ao nível que restrições de recursos computacionais influenciam no desenvolvimento da aplicação;
- Volume de transações: Refere-se ao nível que o alto volume de transações influencia o projeto, desenvolvimento, instalação e suporte da aplicação;
- Entrada de dados *on-line*: Refere-se ao nível de entradas de dados na aplicação através de transações interativas;
- Eficiência do usuário final: Refere-se ao nível de considerações sobre fatores humanos e facilidade de uso pelo usuário final influenciam o desenvolvimento da aplicação;
- Atualização *on-line*: Refere-se ao nível em que os arquivos lógicos internos são atualizados *on-line*;
- Complexidade de processamento: Refere-se ao nível em que o processamento lógico e/ou matemático influencia o desenvolvimento da aplicação;

- Reutilização: Refere-se ao nível em que a aplicação e seu código foram projetados, desenvolvidos e suportados para serem utilizados em outras aplicações;
- Facilidade de instalação: Refere-se ao nível em que a conversão de ambientes preexistentes influencia o desenvolvimento da aplicação. Um plano e/ou ferramentas de conversão e instalação foram fornecidos e testados durante a fase de teste do sistema;
- Facilidade de operação: Refere-se ao nível em que a aplicação atende a alguns aspectos operacionais como inicialização, segurança e recuperação. A aplicação minimiza a necessidade de atividades manuais como montagem de fitas, manipulação de papel e intervenção manual do operador;
- Múltiplos locais: Refere-se ao nível em que a aplicação foi projetada, desenvolvida e suportada para diferentes ambientes de hardware e software;
- Facilidade de mudanças: Refere-se ao nível em que a aplicação foi desenvolvida para facilitar a mudança de sua lógica e/ou estrutura de dados.

As características gerais de sistema conforme definição do IFPUG trata de funções que afetam a aplicação de uma maneira geral, diferente das funções do tipo dado e transação que refletem requisitos de armazenamento e processos respectivamente. Para cada característica devemos atribuir um nível de influência, que varia de 0 a 5 conforme tabela abaixo:

Nível	Influência
0	Nenhuma
1	Mínima
2	Moderada
3	Média
4	Significativa

5	Grande
---	--------

Tabela 6 – Tabela de níveis de influência das características gerais de sistema

Fonte: IFPUG

Assim que os níveis de influência das características gerais de sistema foram determinados, devemos fazer um somatório dos níveis de influência e aplicar na seguinte fórmula:

$$\text{VAF} = (\text{total dos níveis de influência} \times 0,01) + 0,65.$$

Por exemplo, dado um determinado nível de influência para cada uma das 14 CGS:

CGS	Peso
Comunicação de dados	5
Processamento distribuído	2
Performance	2
Configuração altamente utilizada	2
Volume de transações	2
Entrada de dados <i>on-line</i>	5
Eficiência do usuário final	2
Atualização <i>on-line</i>	5
Complexidade de processamento	2
Reusabilidade	0
Facilidade de instalação	1
Facilidade de operação	2
Múltiplos locais	2

Facilidade de mudanças	2
------------------------	---

Temos o total dos níveis de influência igual a 34, aplicando a fórmula:

$$VAF = (34 \times 0,01) + 0,65 = 0,99.$$

Inicialmente quando Allan Albrecht concebeu a metodologia de pontos por função não existiam quaisquer critérios para determinar o fator de ajuste. Sua determinação era de forma muito subjetiva. O fator de ajuste variava em torno de 25% nos pontos de função não-ajustados. Em 1984 houve uma revisão técnica, quando surgiu então os 14 CGS descritos anteriormente. Com esta revisão, o fator de ajuste passou a variar em torno de 35% nos pontos de função não-ajustados. Novamente devemos destacar que o fator de ajuste é opcional devido ao enquadramento da metodologia a norma ISO 14143 em 2002. Mesmo antes do fator de ajuste ser considerado opcional pelo IFPUG, a entidade fez uma pesquisa e constatou que vários usuários da metodologia já não adotavam o fator de ajuste. O assunto é tão polêmico na comunidade dos usuários da APF que o IFPUG criou um grupo de trabalho para se dedicar à questão do fator de ajuste. As críticas são das mais variadas:

- Percentual de variação de 35% insuficiente;
- Falta de requisitos considerados importantes;
- Uso do fator de ajuste não trazer nenhum benefício na estimativa de esforço;
- Outros defendem que os 14 CGS são muitos subjetivos e que dá margem para diferentes interpretações, comprometendo a contagem final.

Por este item ser tão polêmico e subjetivo e o próprio IFPUG inseguro quanto ao assunto, neste trabalho de conclusão não foi implementado o fator de ajuste ao aplicar a APF no projeto estudado. Outra questão que motivou não aplicar o fator de ajuste foi a cultura da empresa Rech Informática Ltda. A empresa sempre buscou padronizar seus processos com base em normas técnicas como a ISO, e neste caso a norma ISO não considera o fator de ajuste.

Para o cálculo dos pontos de função ajustados são aplicadas fórmulas matemáticas para chegar ao resultado final. Nos projetos de desenvolvimento devemos aplicar a seguinte fórmula:

$$DFP = (UFP + CFP) \times VAF$$

Os termos da fórmula são:

- DFP: Número de pontos por função do projeto de desenvolvimento;
- UFP: Número de pontos de função não-ajustados das funções disponíveis após a instalação da aplicação, exceto as funções de conversão (contagem final do projeto, o realizado);
- CFP: Número de pontos de função das funções de conversão;
- VAF: Valor do fator de ajuste.

Como podemos observar a fórmula aplicada para estimativa em projetos não é complexa, o que determinará o sucesso da estimativa está realmente nas fases iniciais do processo de contagem.

Nos projetos de melhoria devemos aplicar a seguinte fórmula:

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

Os termos da fórmula são:

- EFP: Número de pontos de função do projeto de melhoria;
- ADD: Número de pontos de função não-ajustados das novas funções;
- CHGA: Número de pontos de função não-ajustados das funções modificadas, levando em consideração o funcionamento após a alteração;
- CFP: Número de pontos de função não-ajustados de funções de conversão;
- VAFA: Valor do fator de ajuste da aplicação após o projeto de melhoria;
- DEL: Número de pontos de função não-ajustados das funções excluídas da aplicação;
- VAFB: Valor do fator de ajuste da aplicação antes do projeto de melhoria.

A fórmula empregada em projetos de melhoria é mais complexa que a de projetos de desenvolvimento. Isto porque a primeira envolve uma aplicação já em uso, devendo ser calculada a aplicação antes e depois do projeto. Neste tipo de contagem devemos considerar apenas as funções adicionadas, alteradas e excluídas da aplicação. Não devemos considerar na contagem as funções não afetadas no projeto de melhoria. Novamente podemos observar a importância na fase inicial do processo, que envolve a determinação da fronteira. Outra questão na contagem que devemos destacar são as funções modificadas que são utilizadas por mais de uma transação. Neste caso devemos considerar que todas as transações foram alteradas, devendo ser incluídas na contagem dos pontos de função não ajustados.

Para aplicações devemos aplicar duas fórmulas matemáticas. A primeira calcula o tamanho da aplicação a partir apenas das funções solicitadas pelo usuário, sem considerar funções de conversão. A segunda recalcula o seu tamanho após um projeto de melhoria ter alterado suas funcionalidades. A primeira fórmula aplicada lembra a fórmula do projeto de desenvolvimento como podemos ver abaixo:

$$AFP = ADD + VAF$$

Os termos da fórmula são:

- AFP: Número de pontos de função ajustados da aplicação;
- ADD: Número de pontos de função não-ajustados das funções;
- VAF: Valor do fator de ajuste da aplicação.

Após o projeto de melhoria, devemos aplicar a fórmula abaixo:

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

Os termos da fórmula são:

- AFP: Número de pontos de função ajustados da aplicação;
- UFPB: Número de pontos de função não-ajustados antes do projeto de melhoria;

- ADD: Número de pontos de função não-ajustados das funções incluídas no projeto;
- CHGA: Número de pontos de função não-ajustados das funções alteradas depois do término do projeto;
- CHGB: Número de pontos de função não-ajustados das funções alteradas antes do início do projeto;
- DEL: Número de pontos de função não-ajustados das funções excluídas pelo projeto;
- VAFA: Valor do fator de ajuste depois do projeto.

2.10 Ferramentas de medição

Como parte do estudo deste trabalho de conclusão, pesquisamos na internet ferramentas no Brasil que utilizam a técnica de APF como método de mensuração de projetos de desenvolvimento de software. Encontramos apenas a ferramenta “APF-Análise de Pontos de Função” versão 2.5.0.0, desenvolvida pela MCSOFT de Brasília-DF por Ivan Mecenas. Segundo o manual do aplicativo, a versão desta ferramenta está em conformidade com o manual do IFPUG versão 4.1. A ferramenta possui também desde a versão 2.2.0.0 métodos de contagem indicativa e contagem estimada conforme a NESMA. Logo mais demonstramos a análise feita da ferramenta.

3 ESTUDO DE CASO

3.1 Apresentando a empresa estudada

A Rech Informática Ltda, empresa estudada neste trabalho de conclusão, atua desde 1990 no mercado de software para gestão empresarial. Sediada em Novo Hamburgo, a Rech possui seus clientes predominantemente no estado do Rio Grande do Sul, atuando mais especificamente no Vale dos Sinos e região metropolitana de Porto Alegre. A empresa ainda possui alguns clientes também nos estados de Santa Catarina, Paraná e São Paulo. A Rech foi constituída a partir de um alicerce sólido de valores humanos, conjugando experiência, seriedade e competência nas áreas de informática para gestão empresarial.

Fundada pelos irmãos Carlos Vanderlei Rech e Rovani Marcelo Rech, que desde o início estiveram comprometidos em oferecer ao mercado produtos e serviços de qualidade. Foi idealizado o desenvolvimento de um software completo, que eliminasse completamente os retrabalhos e desperdícios nas empresas, através do conhecimento de gestão empresarial e do uso de tecnologias viáveis e produtivas. O objetivo principal da Rech Informática é a conquista diária da satisfação de seus clientes. Por isto, a empresa constantemente investe em novas tecnologias e no treinamento de pessoal, aprimorando o seu ERP, o SIGER® e a qualidade dos serviços prestados. Estes investimentos são tanto em estrutura física, quanto em recursos de pessoal, sempre buscando melhorar cada vez mais o relacionamento da empresa com seus clientes, fornecedores e colaboradores.

3.2 Processos da Rech Informática

Os processos da Rech podemos observar na figura abaixo:

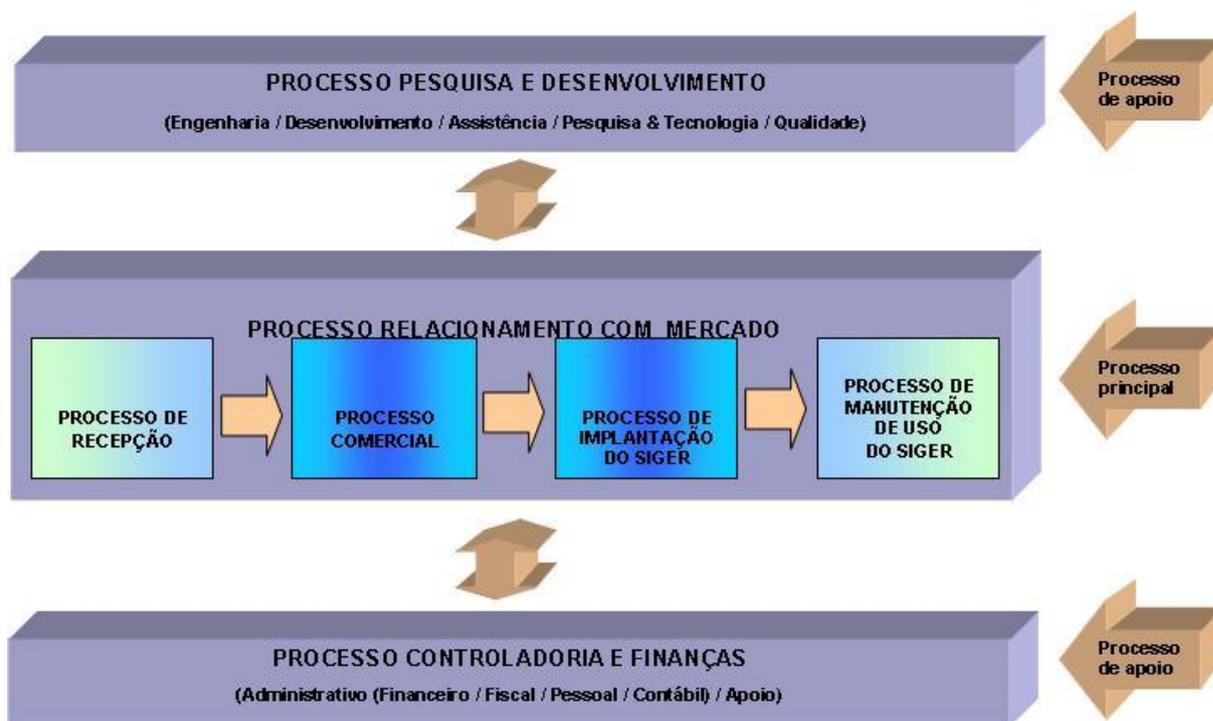


Figura 4 – Processos da Rech Informática Ltda

Fonte: Manual Rech

3.2.1 Processo principal – Relacionamento com mercado

O processo principal da Rech é composto dos subprocessos: recepção, comercial, implantação e manutenção do SIGER®. Os processos de implantação e manutenção são os responsáveis pelo contato direto com o cliente, são por estes processos que a demanda de implementações chega à empresa. Estes processos têm como uma das tarefas fazer o levantamento de requisitos, a partir da necessidade dos clientes. Os colaboradores envolvidos neste processo possuem apenas capacitação do negócio e não de programação ou engenharia de software. Na empresa chamamos de levantamento do “o quê?”.

3.2.2 Processo de apoio – Pesquisa e desenvolvimento

O processo de apoio de pesquisa e desenvolvimento é composto dos subprocessos: engenharia, desenvolvimento, assistência, pesquisa e tecnologia, e qualidade. Este trabalho está direcionado ao subprocesso de engenharia, é nele que a demanda recepcionada do processo principal da empresa chega. Os colaboradores que participam deste processo são programadores mais experientes da empresa, que possuem considerável noção de negócio

para então traduzir o “o quê?” em “como?”. É neste processo que é feita a análise de requisitos, interpretando a necessidade do cliente e traduzindo em instruções aos programadores do processo de desenvolvimento.

3.2.3 Processo de apoio – Controladoria e finanças

O processo de apoio de controladoria e finanças é composto dos subprocessos: administrativo e apoio (limpeza e manutenção). Este processo, de forma sucinta, é responsável pelo funcionamento da empresa, tanto na parte administrativa como na manutenção da infra-estrutura da empresa. Este processo não tem contato direto com o assunto deste trabalho, está sendo apenas apresentado para o completo entendimento dos processos da empresa.

Todos os processos da empresa são registrados no CRM da empresa o SICLA, um software construído pela própria Rech Informática para uso próprio. É a principal ferramenta utilizada para registrar, organizar e sincronizar os movimentos e operações de seus colaboradores, para planejar e acompanhar os processos da empresa e para realizar a análise dos resultados. Este software é utilizado por todos os setores da Rech e está em constante evolução para cada vez mais atender todas as necessidades de registros e controles da gestão, tanto nos processos principais quanto de apoio. A seguir será estudado um aplicativo que trata da APF.

3.3 Problema a ser solucionado

Atualmente o processo de pesquisa e desenvolvimento da Rech possui uma considerável demanda de projetos de implementações e também de melhorias no SIGER. A empresa tem encontrado dificuldade de estimar o tempo de duração de cada projeto e ainda distribuir a carteira de projetos entre seus programadores. Foi pesquisado então metodologias de mensuração de software para auxiliar no processo. Das metodologias pesquisadas foi escolhida a metodologia APF do IFPUG, porque:

- É uma metodologia já difundida no mundo inteiro;
- As demais metodologias partiram da APF do IFPUG;

- Possui certificação ISO o que é uma garantia de qualidade.

A Rech possui uma política de desenvolver suas próprias ferramentas, visto que possui seu próprio CRM o SICLA. Ao desenvolver suas soluções, a empresa também busca por normas técnicas como a ISO para embasamento teórico, garantindo maior qualidade para os seus clientes. Os diretores acreditam que desenvolvendo a sua própria ferramenta a empresa independe de terceiros na manutenção e suporte. Além disso, os desenvolvedores da empresa tem um conhecimento apurado dos processos da empresa, garantindo assim uma ferramenta mais otimizada as suas necessidades. A proposta foi então construir um protótipo baseado na APF para ser exercitado pela empresa e então posteriormente ser incorporado ao SICLA. Outro ponto que pesou para o desenvolvimento deste protótipo, foi que a única ferramenta encontrada de autoria do Ivan Mescenas, faz a contagem individual e não disponibiliza uma opção para gerenciar projetos contados, permitindo montar um plano mestre de produção de projetos. Esta é outra necessidade dos colaboradores, porque não adianta somente saber o tamanho de um projeto, é preciso ainda comparar com outros projetos e montar uma escala de projetos e desenvolvedores para produzi-los.

3.4 Projeto utilizado como *case*

O projeto da Rech que será utilizado como *case* será o projeto de melhoria do módulo de contabilidade. O projeto de melhoria é bastante extenso e já foi produzido pela empresa, porém para estudo deste trabalho de conclusão será considerada apenas uma funcionalidade prevista no projeto. Esta funcionalidade é a mais importante do projeto de melhoria, que compreende a amarração das partidas dos lançamentos contábeis.

No princípio básico de contabilidade, toda movimentação de uma empresa, com base em um documento idôneo como uma nota fiscal, devemos fazer pelo menos dois lançamentos na contabilidade, um a débito e outro a crédito. A amarração está na facilidade do sistema conseguir identificar a partir de um lançamento, qual é o outro lançamento, chamado de “contra-partida” pelos contadores. No exemplo de partidas simples, uma partida a débito e outra a crédito do mesmo valor, não é complexo de identificar, mas existem casos em que possam ter lançamentos de “n” débitos para “n” créditos, tornando difícil para o usuário identificar no sistema as partidas neste caso, sem um campo que faça a amarração destas partidas.

Na contabilidade SIGER os lançamentos são agrupados por lotes de lançamentos, sendo que estes lotes podem ser do tipo mensal (permitindo apenas lançamentos de um determinado mês) ou diário (permitindo apenas lançamentos de uma determinada data). No lançamento de um pagamento de duplicata, por exemplo, devem ser feitos 2 lançamentos: um débito na conta de “títulos a pagar” e um crédito na conta de “fornecedores”. Para fazer estes lançamentos eram informados os dados na janela e após confirmar a gravação estes registros não tinham nenhum vínculo, algo que pudesse identificar os lançamentos como um dependente do outro. Agora com a amarração feita pelo código do lançamento, ao informar este código em uma consulta ou alteração, é possível carregar em um grid todas as partidas do lançamento, facilitando ao contador a conferência dos lançamentos.

Com este ajuste na gravação dos lançamentos, foi permitido criar uma série de melhorias no módulo de contabilidade, diminuindo o trabalho dos clientes/usuários na conciliação da contabilidade. Na perspectiva da Rech, o módulo de contabilidade tem um diferencial considerável perante os sistemas de contabilidade concorrentes. A seguir será analisada a ferramenta APF, fazendo um comparativo entre a ferramenta e a contagem manual dos pontos por função.

4 ANALISANDO A FERRAMENTA APF

4.1 Características da ferramenta APF

A ferramenta “APF-Análise de Pontos de Função” versão 2.5.0.0, foi desenvolvida em Brasília-DF por Ivan Mecnas. Segundo o manual do aplicativo, a versão desta ferramenta está em conformidade com o manual do IFPUG versão 4.1, porém desde a versão 2.2.0.0 também tem métodos de contagem indicativa e contagem estimada conforme a NESMA.

Pontos fortes da ferramenta:

- Manual on-line com instruções da ferramenta e com conceitos da APF para mensuração;
- Possibilidade de registrar as contagens separando por aplicativo ou por projeto;
- Possibilidade de gerenciar usuários, grupos de usuários, ambiente e equipe de projeto;
- Possibilidade de fazer análise estimativa de custo de projeto.

Pontos fracos:

- Utiliza Paradox para armazenamento dos dados, ao invés de um banco de dados;
- Empresa não é do Rio Grande do Sul, comprometendo o suporte técnico aos usuários;

- Software possui muitos recursos que provavelmente não serão utilizados no processo da empresa.

4.2 Contagem de teste

4.2.1 Contagem manual

O primeiro passo do processo de contagem de pontos por função é estabelecer o tipo de contagem a ser feita. Como o projeto estudado refere-se a um sistema existente e que propõe melhorias, o tipo de contagem obviamente será de um projeto de melhoria. Para podermos visualizar melhor a fronteira da aplicação, abaixo segue os passos do processo de registro dos lançamentos contábeis, sob a perspectiva do usuário:

1. Acessar o SIGER;
2. Criar lote de lançamentos ou selecionar lote já existente para adicionar lançamentos;
3. Efetuar a entrada dos lançamentos;
4. Consistir os dados e atualizar a base de dados.

O escopo de contagem será o ajuste da manutenção de lançamentos do módulo de contabilidade, compreendendo a inclusão, alteração e exclusão de lotes e/ou lançamentos. Dentro da contagem das funções do tipo dados podemos identificar como ALI as seguintes tabelas:

1. Tabela de plano de contas;
2. Tabela de lotes;
3. Tabela de lançamentos;
4. Tabela de complementos de lançamentos;
5. Tabela de rateio dos lançamentos por centros de custo.

Dentro da contagem das funções do tipo dados podemos identificar como AIE as seguintes tabelas:

1. Tabela de parâmetros do sistema;

2. Tabela de parâmetros da empresa;
3. Tabela de usuários do SIGER;
4. Tabela de históricos padrões da contabilidade;
5. Tabela de percentuais padrões de rateio por centros de custo;
6. Tabela de clientes e fornecedores.

Resumindo, os ALI que são tabelas alteradas dentro da fronteira de contagem do projeto de melhoria, totalizam cinco tabelas. Os AIE que são apenas tabelas acessadas dentro da fronteira de aplicação totalizam seis tabelas. Seguindo a tabela fornecida pelo IFPUG chegamos aos seguintes resultados:

Nome	Tipo de dado	Complexidade
Tabela de plano de contas	ALI	Baixa
Tabela de lotes	ALI	Baixa
Tabela de lançamentos	ALI	Média
Tabela de complementos de lançamentos	ALI	Baixa
Tabela de rateio dos lançamentos por centros de custo	ALI	Alta
Tabela de parâmetros do sistema	AIE	Baixa
Tabela de parâmetros da empresa	AIE	Baixa
Tabela de usuários do SIGER	AIE	Baixa
Tabela de históricos padrões da contabilidade	AIE	Baixa
Tabela de percentuais padrões de rateio por centros de custo	AIE	Média
Tabela de clientes e fornecedores	AIE	Baixa

Tabela 7 – Tabela de funções do tipo dado da contagem de teste

Dentro da contagem das funções do tipo transação, podemos identificar como EE as seguintes funções:

1. Inclusão de lotes;
2. Alteração de lotes;
3. Exclusão de lotes e lançamentos;
4. Inclusão de lançamentos;
5. Alteração de lançamentos;

6. Exclusão de lançamentos.

Dentro da contagem das funções do tipo transação, podemos identificar como SE as seguintes funções:

1. Atualização de saldos do plano de contas;
2. Extrato de contas contábeis.

Dentro da contagem das funções do tipo transação, podemos identificar como CE as seguintes funções:

1. Listagem de lotes;
2. Consulta de lotes em janela;
3. Listagem de lançamentos;
4. Consulta de lançamentos em janela.

Seguindo as tabelas fornecidas pelo IFPUG chegamos ao seguinte resultado:

Nome	Tipo de transação	Complexidade
Inclusão de lotes	EE	Baixa
Alteração de lotes	EE	Baixa
Exclusão de lotes e lançamentos	EE	Alta
Inclusão de lançamentos	EE	Alta
Alteração de lançamentos	EE	Alta
Exclusão de lançamentos	EE	Alta
Atualização de saldos do plano de contas	SE	Média
Extrato de contas contábeis	SE	Média
Listagem de lotes	CE	Baixa
Consulta de lotes em janela	CE	Baixa
Listagem de lançamentos	CE	Média
Consulta de lançamentos em janela	CE	Média

Tabela 8 – Tabela de complexidade das funções do tipo transação da contagem de teste

Conforme as tabelas de contribuição do ponto por função das funções do tipo dado e transação, fornecidas pelo IFPUG chegamos ao seguinte resultado:

Nome	Tipo de dado	Complexidade	PF
Tabela de plano de contas	ALI	Baixa	7
Tabela de lotes	ALI	Baixa	7
Tabela de lançamentos	ALI	Média	10
Tabela de complementos de lançamentos	ALI	Baixa	7
Tabela de rateio dos lançamentos por centros de custo	ALI	Alta	15
Tabela de parâmetros do sistema	AIE	Baixa	5
Tabela de parâmetros da empresa	AIE	Baixa	5
Tabela de usuários do SIGER	AIE	Baixa	5
Tabela de históricos padrões da contabilidade	AIE	Baixa	5
Tabela de percentuais padrões de rateio por centros de custo	AIE	Média	7
Tabela de clientes e fornecedores	AIE	Baixa	5

Tabela 9 – Resultado dos PF das funções do tipo dado da contagem teste

Nome	Tipo de transação	Complexidade	PF
Inclusão de lotes	EE	Baixa	3
Alteração de lotes	EE	Baixa	3
Exclusão de lotes e lançamentos	EE	Alta	6
Inclusão de lançamentos	EE	Alta	6
Alteração de lançamentos	EE	Alta	6
Exclusão de lançamentos	EE	Alta	6
Atualização de saldos do plano de contas	SE	Média	5
Extrato de contas contábeis	SE	Média	5
Listagem de lotes	CE	Baixa	3
Consulta de lotes em janela	CE	Baixa	3
Listagem de lançamentos	CE	Média	4

Consulta de lançamentos em janela	CE	Média	4
-----------------------------------	----	-------	---

Tabela 10 – Resultado dos PF das funções do tipo transação da contagem teste

Concluindo, após a contagem chegamos a um resultado de 132 pontos por função para este trecho do projeto de melhoria que foi estudado. Novamente reforçamos a questão do fator de ajuste, o qual não foi aplicado porque esta questão não está compreendida na norma ISO, apenas no manual do IFPUG.

4.2.2 Contagem utilizando a ferramenta APF

Para termos um ponto para comparação, foi feita a mesma mensuração utilizando a ferramenta APF, de Ivan Mescenas. A ferramenta APF exige uma determinada ordem para gravar as mensurações, por exemplo, primeiro é necessário cadastrarmos o aplicativo e em seguida o projeto destinado a aplicação cadastrada na etapa anterior. Ao realizarmos o cadastro do projeto para a aplicação, a ferramenta obriga que o primeiro projeto seja o próprio projeto de desenvolvimento que originará a aplicação. Como estamos mensurando um projeto de melhoria, é preciso cadastrarmos e encerrarmos a etapa de mensuração do projeto de desenvolvimento primeiro. Logo, chegamos a conclusão que para utilizarmos a ferramenta desenvolvida por Ivan Mescenas, devemos começar pela mensuração de novas aplicações. Isto torna lenta a implantação do uso da ferramenta na empresa na totalidade, porque devemos primeiro cadastrar todas as aplicações já existentes e simular a mensuração de projeto de desenvolvimento para estas aplicações. Isto é uma tarefa muito dispendiosa para uma *software house*, porque é necessário destinar tempo e recurso humano para esta tarefa e conciliar com a demanda normal da empresa.

Foram inseridos os tipos de dados e tipos de transações, porém para que fosse feito o cálculo, foi necessário contarmos manualmente os dados. Esta contagem manual será desnecessária com o protótipo SICLA-APF, porque está previsto o acesso ao dicionário de campos do SIGER, trazendo automaticamente os tipos de dados ao referenciar a contagem as tabelas envolvidas.

Outro item que observamos na ferramenta foi ao calcularmos os pontos por função a partir dos dados inseridos. Apesar de não cadastrarmos os itens de influência para o ajuste dos pontos por função, a ferramenta fez o ajuste automático. No resultado final, obtivemos 130

pontos não-ajustados. A ferramenta aplicou um fator de ajuste de 0,65, chegando ao resultado de 84,5 pontos por função ajustados.

4.2.3 Contagem utilizando o protótipo SICLA-APF

Para validação do protótipo SICLA-APF, foi feita a mesma contagem no protótipo. O resultado da contagem foi o mesmo da contagem manual, 132 pontos. A diferença para a contagem manual é que no SICLA-APF a atribuição da complexidade e pontos por função é feita automaticamente, com base nas tabelas do IFPUG. Infelizmente até ao final deste estudo, a empresa Rech ainda não havia concluído a implementação do dicionário de dados. Com o dicionário de dados concluído, podemos criar um grande facilitador na contagem dos tipos de dados que é o reconhecimento automático dos arquivos e campos referenciados.

Durante este processo de contagem, foi possível termos uma noção do quanto pode ser complexo e até confuso tentarmos mensurar projetos de desenvolvimento manualmente, sem o auxílio de um software. A proposta discutida a seguir será a construção de um protótipo que venha auxiliar esta tarefa.

4.3 Por que desenvolver a própria ferramenta?

Ao avaliarmos a ferramenta APF da MCSOft, foram pesados os pontos fortes e fracos de se adotar a ferramenta aos processos da empresa, porém tem ainda a questão cultural da Rech, que sempre procurou desenvolver suas próprias ferramentas sem depender de terceiros. Um exemplo disso é que além do próprio ERP, a empresa desenvolveu e faz manutenção do seu próprio CRM, o SICLA.

Apesar de o aplicativo APF ter muitos recursos, atender padrões de mensuração de aplicações e também ser gratuito, ele é muito complexo para o propósito Rech. Novamente ressaltamos a proposta deste estudo, que foi a construção de um protótipo baseado na APF, dentro da norma ISO e que futuramente será incorporado ao SICLA. A Rech, através dos seus diretores, sempre teve o costume de desenvolver as suas próprias ferramentas por vários motivos:

- Poder melhor adapta-la ao seu processo;

- Não depender de ajuste de terceiros;
- Não depender de suporte;
- Poder modificar a aplicação quando necessário e da maneira desejada.

Os diretores da Rech acreditam que desenvolvendo a sua própria ferramenta a empresa independe de terceiros na manutenção e suporte. Além disso, os desenvolvedores da empresa tem um conhecimento apurado dos processos, garantindo assim uma ferramenta mais otimizada as suas necessidades. Outro ponto que pesou para o desenvolvimento deste protótipo foi que a ferramenta APF faz a contagem individual e não disponibiliza uma opção para gerenciar projetos contados. Com o gerenciamento de projetos contados podemos montar um plano mestre de produção de projetos. Esta é outra necessidade dos colaboradores, porque não adianta somente sabermos o tamanho de um projeto, é preciso ainda compararmos com outros projetos e montarmos uma escala de projetos e desenvolvedores para produzi-los.

5 PROTÓTIPO SICLA_APF

5.1 Metodologia de análise

A metodologia de análise empregada no protótipo é estruturada. Não foi optada por orientação objetos porque a intenção é adaptar o protótipo ao SICLA e este ainda não utiliza esta metodologia. O protótipo desenvolvido foi batizado de SICLA_APF, justamente porque será posteriormente agregado ao SICLA. No desenvolvimento do protótipo foi utilizada a linguagem Delphi 2006, em conjunto com o banco de dados Firebird versão 1.5. Em seguida detalharemos a modelagem de dados.

5.2 Estrutura de dados

Para armazenamento dos dados foi utilizado o banco de dados Firebird versão 1.5. Posteriormente será migrado para a mesma versão do Firebird do SICLA. Por que foi utilizado o Firebird? Justamente para facilitar a integração das informações, permitindo que os projetos registrados no SICLA possam armazenar as contagens dos projetos.

Para a modelagem dos dados foram analisadas duas ferramentas: O Microsoft Visio e o DBDesigner. Foi escolhida a ferramenta DBDesigner4 porque foi considerada fácil e intuitiva a sua utilização.

Tabelas acessadas:

- Projetos: Tabela já existente no SICLA, nesta tabela estão todas as informações dos projetos da Rech. Esta tabela será o ponto de ligação entre a base do SICLA e as tabelas criadas no protótipo a serem agregadas futuramente.

Tabelas criadas no protótipo:

- Categoria:
 - Chave primária (principal):
 - Código (integer);
 - Campos:
 - Descrição (char).
- Tipo de contagem:
 - Chave primária (principal):
 - Código (integer);
 - Campos:
 - Descrição (char).
- Complexidade:
 - Chave primária (principal):
 - Código (integer);
 - Campos:
 - Descrição (char).
- Contagem:
 - Chave primária (principal):
 - Código (integer);
 - Campos:
 - Projeto (integer) – acessa tabela de projeto;
 - Tipo contagem (integer) – acessa tabela de tipo de contagem.
- Funções (dados/transação):
 - Chave primária (principal):
 - Contagem (integer) – acessa tabela de contagem;
 - Sequencial (integer).
 - Campos:
 - Categoria (integer) – acessa tabela de categoria;
 - Descrição (char);
 - Quant.registros/arquivos (integer);
 - Campos (integer);
 - Pontos por função (integer);

- Complexidade (integer) – acessa tabela de complexidade.

Para entendermos melhor a estrutura dos dados, podemos observar a figura abaixo:

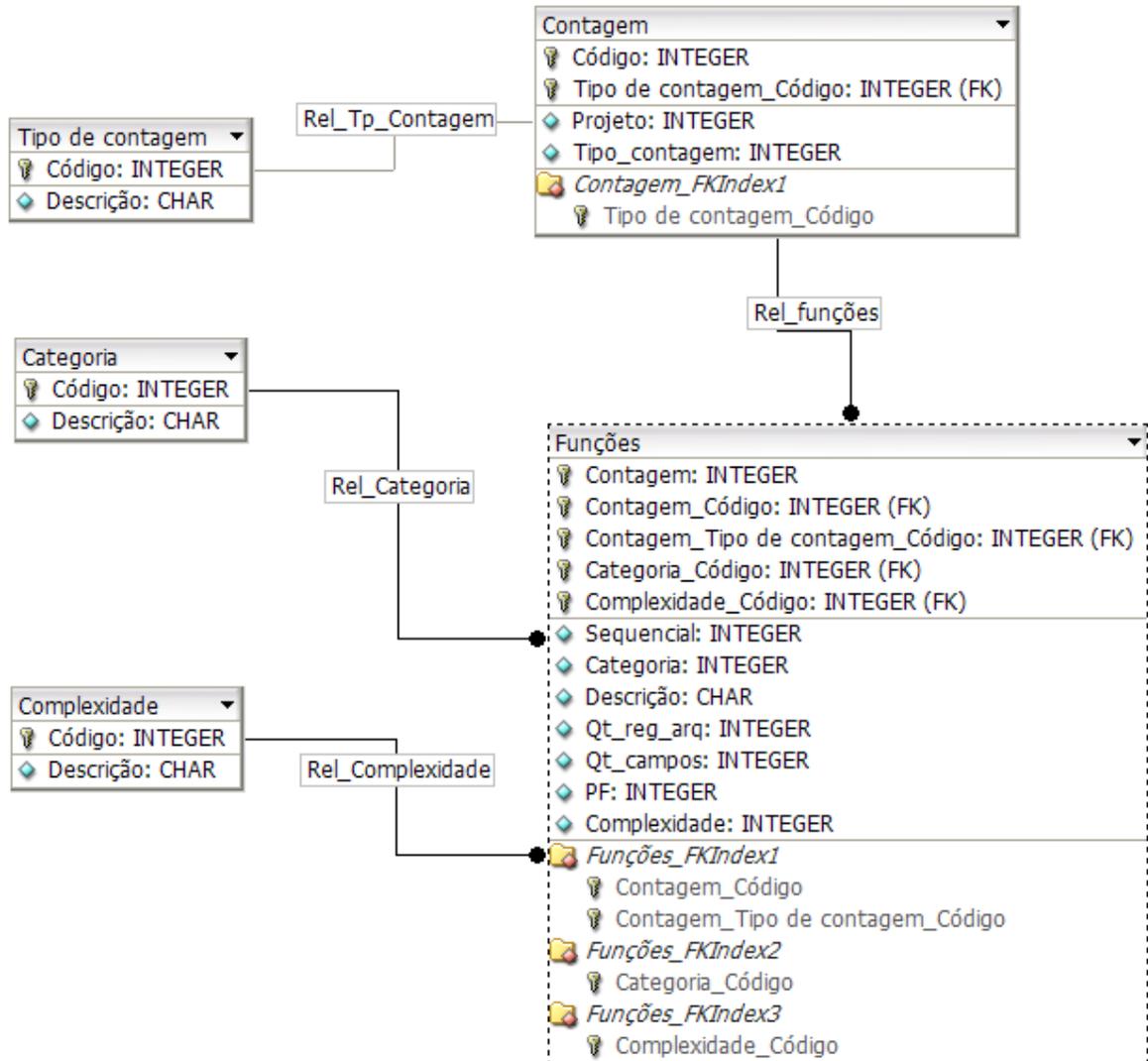


Figura 5 – Estrutura de dados do protótipo SICLA_APF

Podemos observar na figura que temos 4 relacionamentos:

- Rel_Tp_Contagem: Estabelece o relacionamento entre a tabela de tipo de contagem com a tabela principal de nome contagem. A tabela tipo de contagem indica se a contagem de pontos por função é para projeto de desenvolvimento, melhoria ou aplicação. A tabela contagem é para armazenar as informações dos pontos por função da contagem a ser feita;

- **Rel_Categoria:** Estabelece o relacionamento entre a tabela de categorias e a tabela de funções da contagem. A tabela categoria indica o tipo de função a ser contado: ALI, AIE, EE, SE ou CE;
- **Rel_Complexidade:** Estabelece o relacionamento entre a tabela complexidade e a tabela de funções da contagem. A tabela complexidade indica se a função relacionada é de complexidade baixa, média ou alta;
- **Rel_funções:** Estabelece o relacionamento entre a tabela contagem e a tabela funções da contagem. A tabela contagem deve possuir pelo menos 1 registro equivalente na tabela funções da contagem.

A seguir explanaremos o funcionamento do protótipo SICLA-APF.

5.3 Funcionamento do protótipo

O funcionamento do protótipo proposto é bem simples, inicialmente foi projetado apenas para funcionar isoladamente do SICLA e para atender os passos básicos de mensuração que atende o padrão ISO descrito anteriormente.

As tabelas de tipo de contagem, categorias e complexidade já estão criadas com os registros padrões, ou seja: para a tabela de tipo de contagem já tem os registros de projeto de desenvolvimento, projeto de melhoria e aplicação. Nas categorias temos ALI, AIE, EE, SE e CE; e na tabela de complexidade os registros de complexidade baixa, média e alta. Um facilitador criado no protótipo foi a associação automática da complexidade (baixa, média e alta) e atribuição dos pontos por função de cada ALI, AIE, EE, SE e CE. Esta associação é baseada na informação dos tipos de dados e arquivos referenciados e nas tabelas fornecidas pelo IFPUG, contidas no algoritmo do SICLA-APF. Outro facilitador previsto mas não implementado ainda é o reconhecimento automático dos tipos de dados e arquivos referenciados a partir de um dicionário de dados. Isto não foi possível ainda porque a Rech ainda não concluiu a implementação de seu dicionário de dados.

Abaixo seguem as janelas do protótipo, o menu foi construído apenas para o protótipo poder executar e acessar as janelas de manutenção das tabelas. Basicamente o menu ficou dividido em menu horizontal com as opções de cadastros, contagem e saída do protótipo. No menu cadastros encontram-se as manutenções das tabelas de tipo de contagem,

categorias e complexidade. O menu contagem ficou apenas a manutenção da tabela de contagem dos pontos por função. As duas próximas figuras ilustram os menus.



Figura 6 – Menu de cadastros do protótipo SICLA_APF



Figura 7 – Menu contagem do protótipo SICLA_APF

A janela de manutenção ilustrada abaixo se refere à tabela de tipo de contagem. Nesta tabela estão definidos os tipos de contagem de pontos por função, que são projeto de desenvolvimento, projeto de melhoria e aplicação.

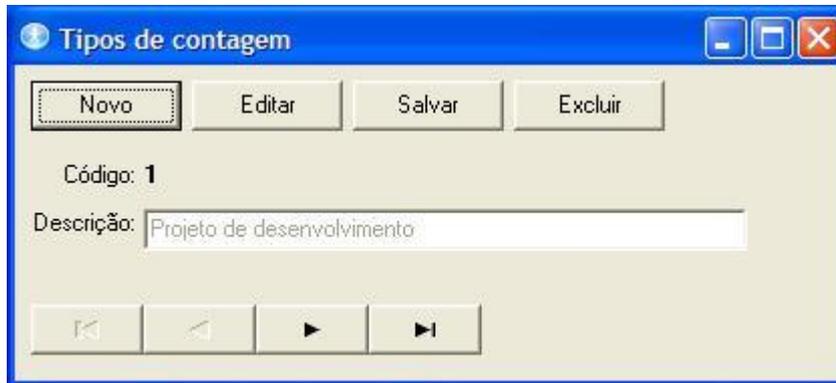


Figura 8 – Janela de manutenção do tipo de contagem do protótipo SICLA_APF

A janela de manutenção ilustrada abaixo se refere à tabela de categorias. Nesta tabela estão definidas as funções do tipo dado (ALI e AIE) e as funções do tipo transação (EE, SE e CE).

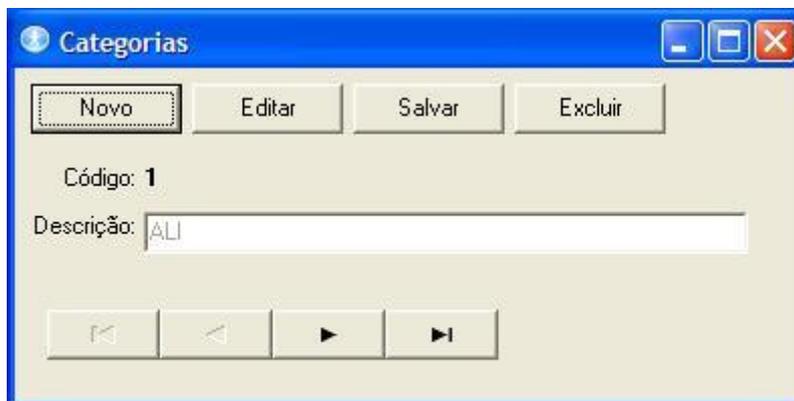


Figura 9 – Janela de manutenção das categorias do protótipo SICLA_APF

A janela de manutenção ilustrada abaixo se refere à tabela de complexidade. Nesta tabela estão definidos os níveis de complexidade dos tipos de dados e transação, que estão classificados em baixa, média e alta.

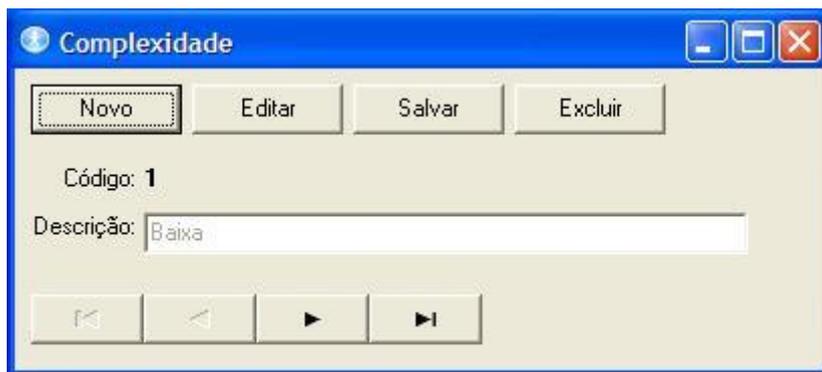


Figura 10 – Janela de manutenção dos tipos de complexidade do protótipo SICLA_APF

A janela de manutenção ilustrada abaixo se refere à tabela de contagem. Esta tabela é a principal do protótipo. Nesta tabela que serão registradas as contagens. Além de acessar a tabela principal de contagem, nesta janela também acessa a tabela de funções da contagem e por consequência as demais tabelas do menu cadastros. No campo projeto é definido o projeto do SICLA a ser contado, será a chave de ligação da tabela de contagem com a tabela de projetos do SICLA. O campo tipo acessa a tabela de tipo de contagem. O campo PF se refere ao total de pontos por função da contagem, calculado automaticamente. No *group-box* funções são exibidos os ALI, AIE, EE, CE e SE.

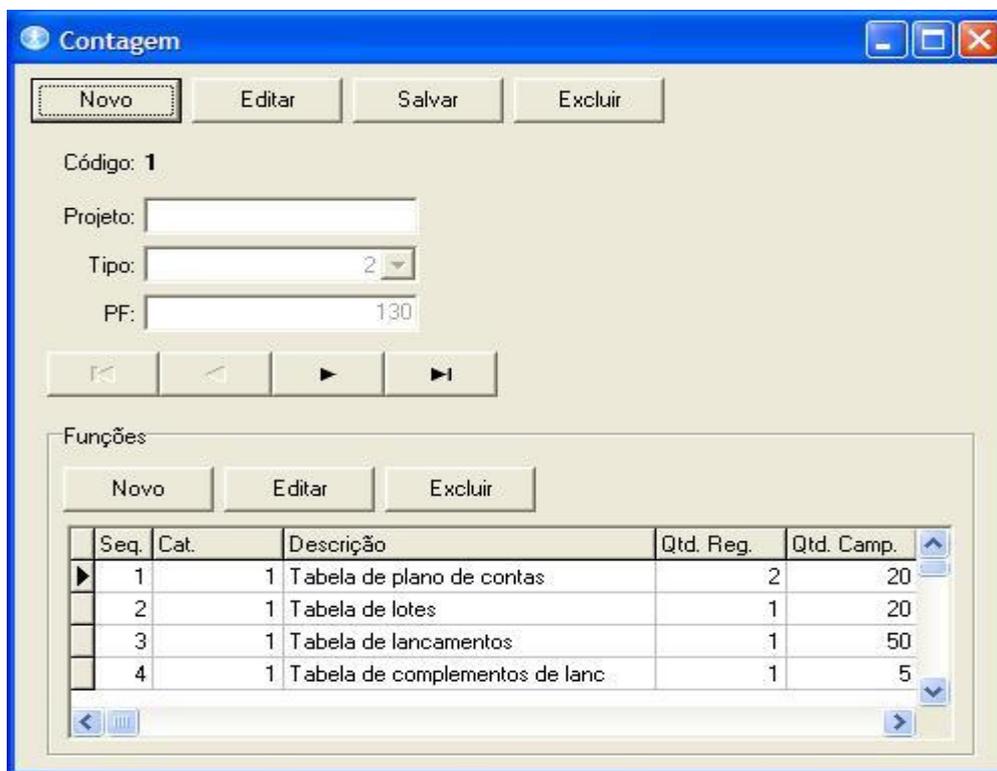
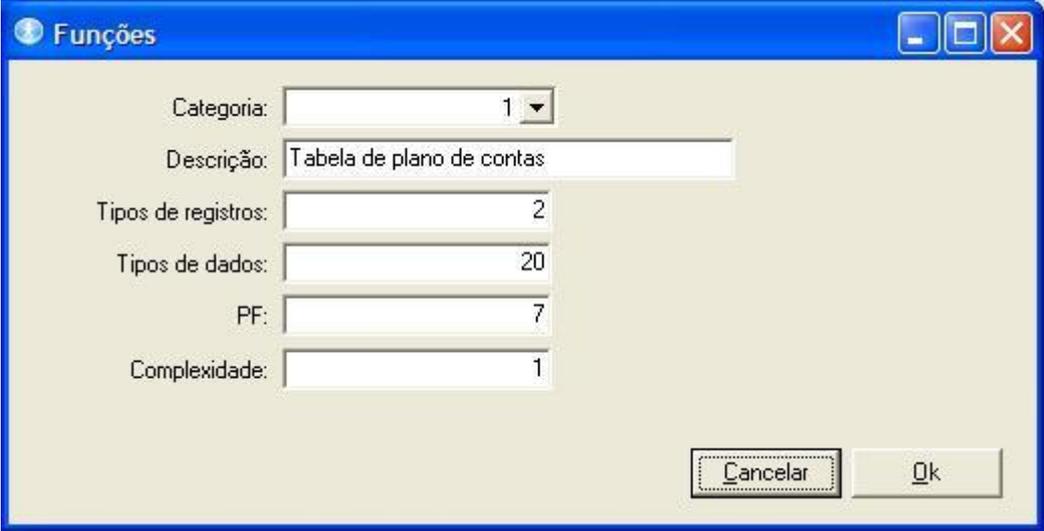


Figura 11 – Janela de manutenção da contagem do protótipo SICLA_APF

A janela de manutenção ilustrada abaixo se refere à tabela de funções que é dependente da tabela contagem. Esta janela é acessada a partir da janela ilustrada anteriormente. Na tabela de funções estão definidos os tipos de dados e transação da contagem a ser realizada. Nesta janela ainda são acessadas as tabelas de categorias e complexidade. No campo “Tipos de registros” quando se tratar de um tipo de dado, neste campo o rótulo será tipos de registros. Se for um tipo de transação o rótulo mudará para “arquivos referenciados”. Os campos PF e complexidade serão atribuídos automaticamente, a partir da informação dos campos “Tipos de registros” e “Tipos de dados”, quando função do tipo dado. Quando for função do tipo transação é a partir dos campos “Arquivos referenciados” e “Tipos de dados”.



Categoria:	1
Descrição:	Tabela de plano de contas
Tipos de registros:	2
Tipos de dados:	20
PF:	7
Complexidade:	1

Figura 12 – Janela de manutenção das funções da contagem do protótipo SICLA_APF

Ao iniciar a contagem, o protótipo pede os dados do registro de contagem como descrição e tipo de contagem e solicita as funções de tipo dados e transação para apuração dos pontos por função. Nesta parte o protótipo é bem simples e neste processo básico de contagem se assemelha muito a ferramenta APF analisada anteriormente.

No protótipo SICLA_APF podem ainda ser implementados uma quantidade de recursos considerável, como reconhecer a partir de um dicionário de dados funções do tipo dado e incluí-las automaticamente na contagem a ser feita. Este recurso é interessante porque diminui sensivelmente a análise e contagem manual, evitando ainda possíveis erros de digitação das funções. Outra implementação de grande impacto será desenvolver toda a parte gerencial de projetos baseando-se na APF. Pode ser implementada uma tabela de desenvolvedores onde para cada registro de desenvolvedor, além de dados cadastrais,

podemos informar o seu valor hora e estipular um média de horas por pontos por função. Outro recurso importante para o gerenciamento de projetos seria a implementação de um calendário com as horas úteis dentro do mês.

Com a contagem de pontos por função de cada projeto, os desenvolvedores e um calendário de horas úteis, podemos implementar um plano mestre de produção de projetos. Neste plano mestre podemos gerenciar a equipe de desenvolvedores, visualizar quem está disponível, definir uma lista de projetos prioritários. O plano mestre ainda pode ser aperfeiçoado, podemos implementar um calendário de manutenção de computadores, ou seja, programar atualizações de softwares, licenças e prever estas pausas na produção no plano mestre de produção. Podemos desenvolver vários relatórios e gráficos estatísticos/gerenciais dos projetos, com horas previstas x realizadas, custo dos projetos, horas produzidas de projetos, enfim várias análises de custos e prazos.

Enfim, além dos recursos anteriormente citados podemos ainda criar muitos outros recursos interessantes, mas para concluir este capítulo o próprio protótipo SICLA_APF pode ainda se tornar uma ferramenta independente do SICLA, podendo ser comercializada pela empresa como um novo produto.

CONCLUSÃO

A APF estudada neste trabalho de conclusão possui uma característica marcante que é a sua simplicidade, principalmente com relação ao cálculo dos pontos por função. Essa metodologia está entre as mais utilizadas para se estimar tamanho de projetos ou sistemas. A contribuição desta metodologia pode ser grandiosa e de valor inestimável para os processos da empresa se aplicada em todas as áreas possíveis, como gerência de projetos, estimativas de custo e produtividade e demais contribuições que a APF proporciona.

Estando o protótipo de APF proposto neste trabalho de conclusão produzido, este será imediatamente utilizado pela empresa Rech Informática, em caráter de testes. Estando o protótipo aprovado pelos colaboradores e pela direção da Rech, ele será agregado ao sistema de controle interno da empresa, o SICLA, o qual já possui uma parte destinada a controle de projetos da empresa.

A contribuição final deste trabalho de conclusão:

- Aprendizado dos colaboradores sobre a APF;
- Melhor gerenciamento dos projetos a partir de estimativas de tempo mais próximas do real;
- Melhor negociação com clientes ao estipular prazos das implementações.

Depois de implementados os recursos previstos no capítulo anterior e devidamente exercitados no processo da empresa, a ferramenta tem grande potencial de ser comercializada separadamente do SICLA, para outras empresas de desenvolvimento de sistemas.

REFERÊNCIAS BIBLIOGRÁFICAS

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

VASQUEZ, Carlos Eduardo; SIMÕES, Guilherme Siqueira; ALBERT, Renato Machado. **Análise de Pontos por Função. Medição, Estimativas e Gerenciamento de Projetos de Software.** São Paulo, 2005.

FIORINI, Soeli T.; STAA, Arndt von; BAPTISTA, Renan Martins. **Engenharia de software com CMMI.** Rio de Janeiro, 1998.

FILHO, Wilson de Pádua Paula. **Engenharia de Software: Fundamentos, Métodos e Padrões.** Rio de Janeiro, 2003.

JOÃO, Belmiro do Nascimento. **Metodologias de Desenvolvimento de Sistemas.** São Paulo, 1999.

DEMARCO, Tom. **Controle de projetos de Software.** São Paulo, 1991.

DEMARCO, Tom. **Análise estruturada e especificação de sistema.** Ed. Campus. São Paulo.

JONES, Capers - **Estimating Software Costs** - McGraw-Hill, 1998.

HAZAN, Cláudia. **Análise de Pontos por Função: Uma Ferramenta na Implantação do Modelo CMMI.** Brasília, 2003. Disponível em:

<[http://www.serpro.gov.br/publicacao/tematec/2003/ Portal do Serpro - Análise de Pontos por Função Uma Ferramenta na Implantação do Modelo CMM.htm](http://www.serpro.gov.br/publicacao/tematec/2003/Portal%20do%20Serpro%20-%20An%C3%A1lise%20de%20Pontos%20por%20Fun%C7%A7%C3%A3o%20Uma%20Ferramenta%20na%20Implanta%C7%A7%C3%A3o%20do%20Modelo%20CMM.htm).

HAZAN, Cláudia. **Uma aplicação da APF nas estimativas de projetos web**. Disponível em: <http://www.serpro.gov.br>

AZEVEDO, Douglas José Peixoto de. **FPA – Function Point Analysis**. Disponível em: <http://www.pr.gov.br/batebyte/edicoes/1997/bb68/fpa.htm>, acesso em Agosto/2007.

MANUAL RECH. **Manual de integração Rech**. Novo Hamburgo, Março/2007.