



PUC

ISSN 0103-9741

Monografias em Ciência da Computação
nº 16/05

Avaliação do Tamanho Funcional de Ferramentas de E-learning

**Claudia Hazan
Hugo Fuks
Carlos José Pereira de Lucena**

Departamento de Informática

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO
RUA MARQUÊS DE SÃO VICENTE, 225 - CEP 22453-900
RIO DE JANEIRO - BRASIL**

Avaliação do Tamanho Funcional de Ferramentas de E-learning *

Claudia Hazan

claudinha@inf.puc-rio.br

Hugo Fuks

hugo@inf.puc-rio.br

Carlos José Pereira de Lucena

lucena@inf.puc-rio.br

Abstract: E-learning tools are key factors to the success of Web-based Education. There has been substantial research evaluating such tools on criteria regarding the quality of the finished product. However, little has been said so far about the quality and productivity of the tools' development process. This paper proposes the Function Point Analysis (FPA) method as a means to functional size assessment of e-learning tools. It also presents how the method was used and the results collected in the evaluation of the AulaNet learningware.

Keywords: Metrics, e-learning tools assessment, Function Point Analysis (FPA).

Resumo: As ferramentas de *e-learning* constituem um dos elementos chave para o sucesso dos cursos ministrados à distância pela Web. A literatura tem explorado bastante a avaliação destas ferramentas, sob o ponto de vista da qualidade do produto. No entanto, a qualidade e produtividade do processo de desenvolvimento de tais ferramentas têm sido pouco abordadas. Este artigo tem como objetivo propor o método Análise de Pontos por Função (APF) para avaliação do tamanho funcional de ferramentas de *e-learning* e apresentar a aplicação do método na avaliação do tamanho funcional do Ambiente AulaNet.

Palavras chave: Métrica, avaliação de ferramentas de e-learning, Análise de Pontos por Função (APF)

* Trabalho patrocinado pelo Ministério de Ciência e Tecnologia da Presidência da República Federativa do Brasil

Responsável por publicações:

Rosane Teles Lins Castilho
Assessoria de Biblioteca, Documentação e Informação
PUC-Rio Departamento de Informática
Rua Marquês de São Vicente, 225 - Gávea
22453-900 Rio de Janeiro RJ Brasil
Tel. +55 21 3114-1516 Fax: +55 21 3114-1530
E-mail: bib-di@inf.puc-rio.br

Sumário

1. INTRODUÇÃO	1
1.1. ORGANIZAÇÃO DO ARTIGO.....	2
2. ESTADO DA ARTE.....	2
2.1. PROPOSTA	7
3. ELABORAÇÃO	7
3.1. ARQUIVOS LÓGICOS INTERNOS	8
3.2. ARQUIVOS DE INTERFACE EXTERNA	9
3.3. ENTRADAS EXTERNAS	9
3.4. SAÍDAS EXTERNAS.....	10
3.5. CONSULTAS EXTERNAS.....	12
3.6. CÁLCULO DOS PONTOS DE FUNÇÃO NÃO AJUSTADOS	14
3.7. CÁLCULO DO FATOR DE AJUSTE	15
4. DISCUSSÃO E CONTRIBUIÇÕES	18
5. CONCLUSÕES	19
REFERÊNCIAS BIBLIOGRÁFICAS	20

1. Introdução

O interesse das organizações em como gerar conhecimento útil em tempo de mudanças e incertezas é crescente. A tecnologia gera ambientes que dão suporte às diferentes formas de relacionamento humano. No caso específico da Internet, a estratégia cliente-servidor permite a criação de espaços de compartilhamento e troca de informação. Acopladas a tais ambientes há ferramentas para movimentar informação e facilitar o contato entre humanos. Estes ambientes virtuais favorecem a descentralização e distribuição de informações relativas ao conhecimento humano [Fuks, 2000a].

As ferramentas ou ambientes virtuais de aprendizagem desempenham um papel fundamental no apoio à administração, criação, manutenção e execução de cursos à distância. O AulaNet é um ambiente de software baseado na Web, desenvolvido desde 1997 no Laboratório de Engenharia de Software - LES - do Departamento de Informática da PUC-Rio, que fundamenta-se nas seguintes premissas:

- Colaboração: além dos conteúdos criados pelo autor do curso, como Aulas, Documentações, Bibliografias etc., o ambiente também dá suporte a conteúdos colaborados – Conferências, Debates e Tarefas, que são construídos no decorrer de um curso pelos aprendizes, trabalhando em grupo. Colaboração é o aspecto essencial do AulaNet, que segue o Modelo de Colaboração 3 C descrito em [Fuks 1999]. Em linhas gerais, este modelo descreve que, para haver colaboração, criando um ambiente propício à aprendizagem em grupo, os participantes devem [Fuks, 2002]:
 - a) Comunicar: compartilhar idéias e assumir compromissos;
 - b) Coordenar: planejar a execução destes compromissos e estar em sintonia com o trabalho dos outros participantes;
 - c) Cooperar: realizar as tarefas que lhe foram atribuídas dentro do grupo.
- Independência de *expertise*: o autor do curso e seus aprendizes não precisam ser especialistas em Internet para criar, manter e assistir aos cursos na Web;
- Reuso: o ambiente possibilita a reutilização de conteúdos já existentes em mídia digital, através da importação de arquivos.

A modalidade de treinamentos baseado em Ensino à Distância (EaD) tem se tornado bastante atraente para as organizações, devido ao benefício do aprendiz realizar o curso no seu tempo disponível e na sua estação de trabalho. No caso das grandes organizações, geograficamente espalhadas, um treinamento baseado em EaD apresenta o benefício adicional de redução de custos de deslocamento dos participantes e do instrutor. Muitas organizações também têm investido fortemente em comunidades virtuais colaborativas de aprendizagem, as quais podem ser implementadas por meio do ambiente de *e-learning*. Estas comunidades geram e armazenam o conhecimento de maneira organizada, propiciando a gestão do conhecimento corporativo.

A literatura da área de *e-learning* tem discutido bastante sobre os participantes ou atores deste ambiente e seus papéis (Administrador, Coordenador, Docente Co-autor, Mediador e Aprendiz). Também têm sido abordado as funcionalidades ou componentes relevantes para a construção de ambientes de *e-learning* ou *learningware*¹.

¹ *Learningware* é a expressão utilizada para denominar um groupware dedicado a aprendizagem colaborativa na Web. O AulaNet é um ambiente baseado em uma abordagem de *learningware* para criação, aplicação e gerenciamento de cursos pela Internet [Fuks, 2000b].

No entanto, a literatura tem discutido muito pouco sobre a gestão do desenvolvimento de uma ferramenta de *e-learning*. Suponha que uma organização decida investir na construção de um ambiente de *e-learning*, como avaliar-se o esforço e os custos para o projeto de desenvolvimento de software? o que é economicamente mais vantajoso adquirir-se um pacote e pagar-se pelas customizações ou especificar-se e subcontratar-se o desenvolvimento de um ambiente de *e-learning* customizado? Além disso, é freqüente a realização de projetos de manutenção evolutiva nas aplicações implantadas devido à evolução de requisitos [Hazan, 2003c]. Conforme os usuários utilizam o produto de software, estes demandam novas funcionalidades. Os requisitos também evoluem devido às mudanças nos processos da organização [Kotonya, 1998]. Então, como avaliar o custo e o esforço associados aos projetos de manutenção evolutiva de uma ferramenta de *e-learning*?

Com a finalidade de apoiar a gestão de projetos de desenvolvimento de ambientes de *e-learning*, este trabalho propõe o método Análise de Pontos por Função para a avaliação do tamanho funcional de ferramentas de *e-learning*. O tamanho funcional é uma medida de tamanho de *software*, baseada em uma avaliação padronizada dos requisitos dos usuários. A Contagem de Pontos de Função descrita neste trabalho está baseada manual de práticas de contagem (CPM) versão 4.2, publicado pelo IFPUG (*International Function Point Users Group*) [IFPUG, 2004]. O tamanho funcional de um software, baseado na métrica de Pontos de Função, constitui um atributo importante de normalização utilizado na definição de indicadores da qualidade e produtividade de software [Hazan, 1999]. Os indicadores da qualidade apoiam uma análise da qualidade do processo e do produto. Os indicadores de produtividade permitem uma avaliação da produtividade do processo de desenvolvimento de software e suportam a geração de estimativas de esforço, custos e prazo para os novos projetos de desenvolvimento e de manutenção evolutiva [Boehm, 2000] [Hazan, 2003b]. Os indicadores também são de grande utilidade para realização de *benchmarking*.

1.1. Organização do artigo

Este trabalho encontra-se organizado da seguinte forma: a seção 2 apresenta o Estado da Arte na área de métricas funcionais e de sua aplicação no dimensionamento de tamanho de projetos de software, destacando método Análise de Pontos por Função e os benefícios de sua aplicação. A seção 3 tem como propósito mostrar a Contagem de Pontos de Função das funcionalidades do ambiente AulaNet para o papel de Aprendiz, descritas no “Manual do Aprendiz do AulaNet” [EDUWEB], também é calculado o Fator de Ajuste da contagem, considerando o ambiente AulaNet como um todo. A seção 4 apresenta uma análise dos resultados da contagem obtidos na seção 3, visando apoiar a gestão dos projetos de manutenção evolutiva do AulaNet, bem como o desenvolvimento de novos ambientes de *learningware*. A seção 5 conclui o trabalho e identifica possíveis trabalhos futuros.

2. Estado da arte

O tamanho do projeto de software tem relação direta com o esforço e os custos do desenvolvimento deste projeto. Pode-se obter o tamanho de um projeto com base nas funções requeridas por quem demanda o software [Fenton, 2002]. Assim, o tamanho do projeto de software influencia a solução técnica adotada e constitui uma informação básica para o processo de gestão do projeto [Ross]. Atualmente são várias as métricas de tamanho funcional existentes, dentre outras pode-se destacar: *Function Point Analysis* ou Análise de Pontos de Função; *Bang*; *Feature Points*; *Boeing's ED Function Point*; *Mark II*; *Use Case Point* ou Pontos de Casos de Uso, *COSMIC Full Function Point* [Garmus, 2001].

Esta seção tem como propósito apresentar as métricas: Pontos de Função (PF) e Pontos de Caso de Uso, que tem sido mais utilizadas na indústria para avaliação de tamanho funcional de projetos de software [Andrade, 2004].

A Análise de Pontos por Função (APF), proposta por Allan Albretch em 1979, é um método para a medição de projetos de desenvolvimento e de manutenção evolutiva de software, que visa estabelecer o tamanho funcional do software em Pontos de Função (PF), considerando as funcionalidades implementadas, sob o ponto de vista do usuário. Os principais objetivos da APF são: Medir a funcionalidade requisitada e recebida pelo usuário; e Medir projetos de desenvolvimento e de manutenção de software, independentemente da tecnologia utilizada [Vazquez, 2004] [IFPUG, 2004].

A Análise de Pontos de Função (APF) é atualmente a métrica mais utilizada pelo mercado [Andrade, 2004] [IFPUG]. A utilização de APF traz vários benefícios: Para o comprador de software, possibilita determinar o tamanho da ferramenta adquirida por meio da contagem de Pontos de Função de todas as funcionalidades incluídas. Para o gestor, serve como apoio à análise da qualidade e da produtividade do desenvolvimento do ambiente, além de ser um mecanismo eficaz para estimar esforço, custos e recursos envolvidos em projetos de manutenção do ambiente [Hazan, 2003b]. Além disso, a métrica Pontos de Função apoia a implantação de modelos da qualidade do processo de desenvolvimento de software, tais como o CMM [Hazan, 2003a] e o CMMI [Dekkers, 2002]. Além disso, a métrica Pontos de Função tornou-se um padrão internacional, sendo reconhecida pela norma ISO/IEC 20926 [Dekkers, 2003].

O Procedimento de Contagem de Pontos de Função, ilustrado na **Figura 1**, consiste nos seguintes passos [IFPUG, 2004]:

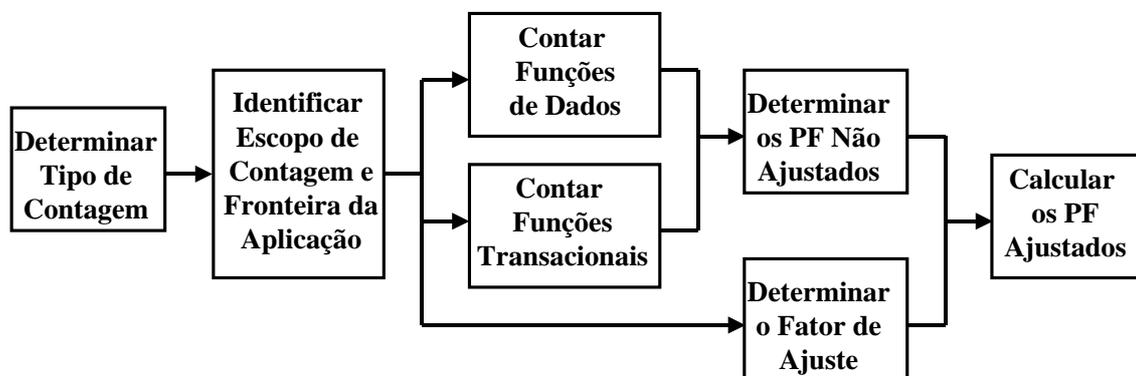


Figura 1: Fluxograma do Procedimento de Contagem de Pontos de Função [Hazan, 2000]

Determinar o Tipo de Contagem de Pontos por Função

O primeiro passo na contagem de Pontos por Função é determinar o tipo de contagem a ser realizada. As contagens de Pontos por Função podem ser associadas a projetos ou aplicações. Existem três tipos de contagens de Pontos de Função:

- Contagem de Pontos por Função de Projetos de Desenvolvimento
- Contagem de Pontos por Função de Aplicações Instaladas
- Contagem de Pontos por Função de Projetos de Manutenção

Identificar o Escopo de Contagem e Fronteira da Aplicação

O escopo de contagem define a funcionalidade que será incluída em uma contagem de Pontos por Função específica. A fronteira da aplicação indica o limite lógico entre a aplicação que está sendo medida, o usuário e outras aplicações que interagem com a aplicação que está sendo medida.

Determinar a Contagem de Pontos por Função Não Ajustados

Os Pontos por Função não ajustados refletem as funcionalidades fornecidas pela aplicação ou projeto de desenvolvimento para o usuário. As funcionalidades da aplicação fornecidas para o usuário são avaliadas em termos de *O QUE É* entregue pela aplicação. A Contagem de Pontos por Função não ajustados leva em consideração dois tipos de funções: **Funções de Dados** e **Funções Transacionais**.

Contagem das Funções de Dados

As Funções de Dados representam a funcionalidade fornecida para o usuário relativa aos requisitos de dados internos e externos à aplicação. As Funções de Dados são os **Arquivos Lógicos Internos** e os **Arquivos de Interface Externa** (Figura 2).

- Um **Arquivo Lógico Interno (ALI)** é um grupo de dados logicamente relacionados ou informações de controle, identificado pelo usuário, mantido dentro da fronteira da aplicação. O objetivo principal de um Arquivo Lógico Interno é armazenar os dados mantidos através de um ou mais processos elementares da aplicação.
- Um **Arquivo de Interface Externa (AIE)** é um grupo de dados logicamente relacionados ou informações de controle, identificado pelo usuário, referenciados pela aplicação, no entanto são mantidos dentro da fronteira de outra aplicação. O objetivo principal de um Arquivo de Interface Externa é armazenar os dados referenciados através de um ou mais processos elementares da aplicação sendo contada. Isto significa que um **Arquivo de Interface Externa** de uma aplicação deve ser um **ALI** de uma outra aplicação.

Contagem das Funções Transacionais

As funções transacionais representam as funcionalidades de processamento de dados fornecidas para o usuário. As Funções Transacionais são as **Entradas Externas**, as **Saídas Externas** e as **Consultas Externas**. (Figura 2)

- Uma **Entrada Externa (EE)** é um processo elementar que processa dados ou informação de controle que entram pela fronteira da aplicação. O objetivo principal de uma Entrada Externa é manter um ou mais ALI e/ou alterar o comportamento do sistema.
- Uma **Saída Externa (SE)** é um processo elementar que envia dados ou informação de controle para fora da fronteira da aplicação. O objetivo principal de uma Saída Externa é apresentar informação para um usuário através de um processamento lógico e não apenas uma simples recuperação de dados ou informação de controle. O processamento lógico deve conter no mínimo uma fórmula matemática ou cálculo, ou criar dados derivados, ou manter ALIs ou alterar o comportamento da aplicação.
- Uma **Consulta Externa (CE)** é um processo elementar que envia dados ou informação de controle para fora da fronteira da aplicação. O objetivo principal de uma Consulta Externa é apresentar informação para o usuário através da recuperação de dados ou informação de controle. O processamento lógico não contém fórmulas matemáticas ou cálculos e não cria dados derivados. Nenhum Arquivo Lógico Interno é mantido durante o processamento, nem o comportamento do sistema é alterado.

Determinar o Valor do Fator de Ajuste

O valor do Fator de Ajuste indica a funcionalidade geral fornecida para o usuário da aplicação. O fator de ajuste abrange as 14 Características Gerais do Sistema (CGS) que estimam a magnitude das funcionalidades gerais da aplicação. Cada característica possui descrições associadas que ajudam a determinar o nível de influência da característica. Os níveis de influência variam em uma escala de 0 (nenhuma influência) a 5 (forte influência).

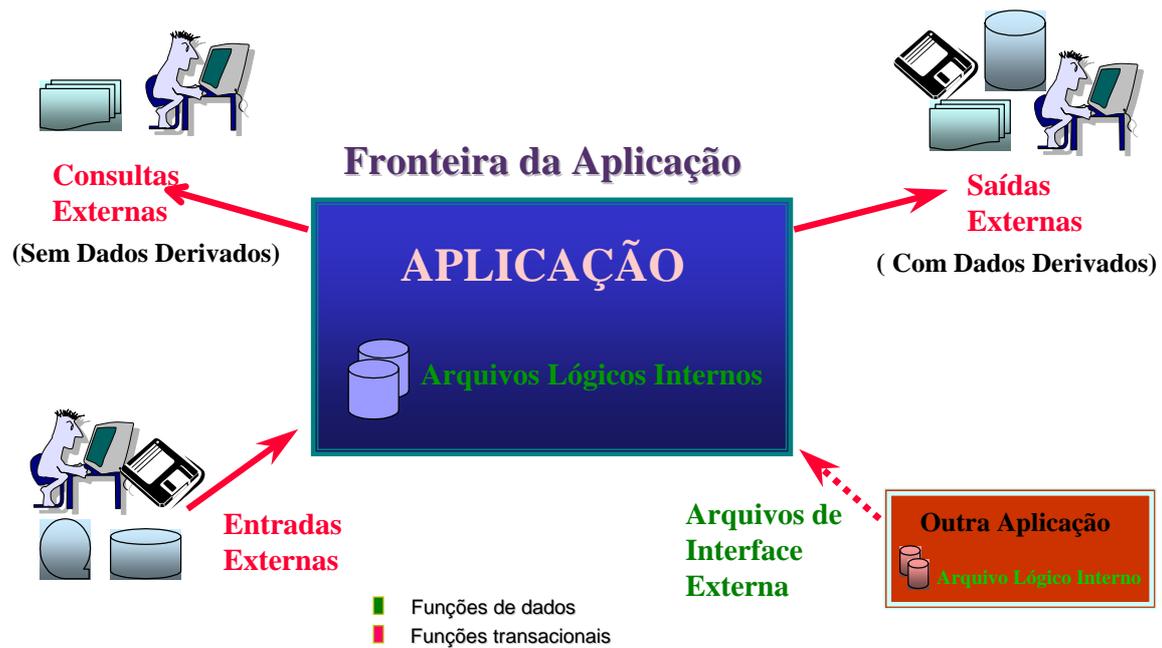


Figura 2: Tipos Funcionais da Análise de Pontos por Função [Hazan, 1999]

Calcular os Pontos por Função Ajustados

Os Pontos por Função Ajustados são calculados usando uma fórmula específica para uma Contagem de PF de um Projeto de Desenvolvimento, um Projeto de Manutenção, ou uma Aplicação Implantada.

A Análise de Pontos por Função também tem sido utilizada com sucesso na indústria no suporte ao Planejamento e Acompanhamento de Projetos de Software. A geração de estimativas de tamanho, esforço, cronograma e custo constituem a base para o estabelecimento do Plano do Projeto de Software [SEI, 2002]. A principal vantagem da utilização de APF é que o método possibilita a geração de estimativas logo nas fases iniciais do processo de desenvolvimento, e apoia a geração de reestimativas, fundamentais para o replanejamento [Hazan, 2003a].

A métrica Pontos por Casos de Uso (PCU) foi proposta por Gustav Karner [Karner, 1993] com o propósito de estimar recursos para projetos de software orientados a objeto, modelados por meio de especificação de Casos de Uso. Em linhas gerais, o método de contagem de Pontos por Caso de Uso consiste nos seguintes passos:

1. Contar os atores e identificar sua complexidade;
2. Contar os casos de uso e identificar sua complexidade;
3. Calcular os PCUs não ajustados;
4. Determinar o fator de complexidade técnica;
5. Determinar o fator de complexidade ambiental;
6. Calcular os PCUs ajustados.

Com o resultado desta medição e sabendo-se a produtividade média da organização para produzir um PCU, pode-se então estimar o esforço total para o projeto. A seguir são mostradas as principais vantagens da métrica Pontos de Função (PF) sobre a métrica Pontos por Caso de Uso (PCU).

A métrica Pontos por Caso de Uso foi apresentada para o mercado como uma solução simples para as estimativas de tamanho, especialmente para os que não conheciam as Práticas de Contagem de Pontos por Função, descritas em [IFPUG, 2004], ou as consideravam complexas para serem aplicadas. Além disso, existe uma falsa noção de que a Análise de Pontos por Função não pode ser aplicada para dimensionar sistemas orientados a objetos. Como a contagem de Pontos de Função, considera os requisitos funcionais do sistema, de maneira independente da metodologia utilizada, os PFs também podem ser aplicados para sistemas cujos requisitos tenham sido modelados utilizando-se a especificação de casos de uso. Enquanto que, o PCU somente pode ser aplicado em projetos de software cuja especificação tenha sido expressa em casos de uso. Note que a contagem de PF independe da forma como os requisitos do software foram expressos. Esta vantagem da APF foi citada pelo próprio Gustav Karner [Karner, 1993].

Não é possível na prática aplicar PCU na medição de aplicações existentes cuja documentação esteja desatualizada ou sequer exista. A alternativa seria escrever os casos de uso destas aplicações para só então medi-las. Porém isto tornaria a medição inviável devido ao alto custo. Com a APF é possível realizar a medição analisando-se a própria aplicação em uso. A medição de aplicações instaladas pode ser útil na elaboração de contratos de *outsourcing* de sistemas e na aquisição de pacotes. Por exemplo na avaliação da aquisição ou subcontratação do desenvolvimento de ferramentas de *e-learning*.

Não existe um padrão único para a escrita do caso de uso. Diferentes estilos na escrita dos caso de uso ou na sua granularidade podem levar a resultados diferentes na medição por PCU. A APF não sofre deste problema pois independe da forma como os requisitos são expressos ou documentados.

Devido ao processo de medição do PCU ser baseado em casos de uso, o método não pode ser empregado antes de concluída a análise de requisitos do projeto. Na maioria das vezes há a necessidade de se obter uma estimativa antes da finalização desta etapa. O processo de contagem de PF também só pode ser empregado após o levantamento dos requisitos do projeto, na fase de projeto lógico. Porém existem técnicas estimativas de tamanho em Pontos de Função que podem ser aplicadas com sucesso antes da análise de requisitos ser concluída. Os principais métodos de estimativa de Pontos de Função utilizados são os seguintes: Contagem Indicativa [NESMA], Contagem Estimativa [Hazan, 2004] e Estimativas Percentuais [ISBSG] [UNISYS, 1995].

O método do PCU não contempla a medição de projetos de melhoria funcional do software (manutenção evolutiva), somente projetos de desenvolvimento. A APF contempla a medição de projetos de desenvolvimento, projetos de manutenção evolutiva e aplicações implantadas. Note que a APF pode ser usado para dimensionar os projetos concluídos de uma organização, possibilitando a geração de uma *baseline* para as estimativas.

Não existe um grupo de usuários ou organização responsável pela padronização ou evolução do método PCU; e as referências bibliográficas sobre o assunto é escassa. Para a APF existe o IFPUG (*International Function Point Users Group*) que é o responsável por manter o Manual de Práticas de Contagem [IFPUG, 2004]. Além disso, o método PCU não é aderente à norma ISO/IEC 14143 que define um modelo para a medição funcional de software. A APF, conforme o manual do IFPUG versão 4.1.1, está padronizada sob a norma ISO/IEC 20926.

Não existe um programa de certificação de profissionais que conheçam a técnica do PCU e saibam aplicá-la de forma adequada. A APF possui o programa de certificação CFPS (*Certified Function Point Specialist*) promovido pelo IFPUG [IFPUG], garantindo assim a utilização correta do método APF.

O fator ambiental inserido no PCU é subjetivo, dificultando a consistência da aplicação do método em programas de métricas de software e *benchmarking* entre organizações, pois torna o tamanho de um projeto variável; sem que sua funcionalidade sequer mude. Se um mesmo projeto for entregue a duas equipes distintas a contagem dos Pontos por Caso de Uso deste projeto será também diferente em cada situação. Ou seja, o mesmo projeto possui dois tamanhos distintos.

Embora muitas organizações e referências bibliográficas considerem os Pontos por Caso de Uso como o estado da arte em métricas de tamanho de software [Andrade, 2004] [Cunha, 2003] [Damodaram], analisando-se as desvantagens expostas acima pode-se concluir que o PCU não traz nenhum benefício adicional em relação à APF. Também é importante destacar a maturidade da métrica PF na indústria de software (25 anos de criação e evolução) em relação à métrica PCU (menos de 10 anos). E ainda, na aplicação de métricas para estimar-se tamanho, a métrica de Pontos de Função possui uma vantagem adicional que é a sua utilização como insumo para o COCOMO II [Boehm, 2000], principal modelo utilizado pela indústria para a derivação das estimativas de esforço, prazo e custo a partir das estimativas de tamanho. Assim, a avaliação do tamanho funcional do ambiente AulaNet proposto neste trabalho é realizada baseando-se no procedimento de contagem de Pontos de Função descrito no Manual de Práticas de Contagem (CPM 4.2) [IFPUG, 2004].

2.1. Proposta

Este artigo tem como propósito descrever uma aplicação do procedimento de contagem de Pontos de Função na avaliação do tamanho funcional das funcionalidades do ambiente AulaNet para o papel de Aprendiz, descritas no “Manual do Aprendiz do AulaNet” [EDUWEB]. A avaliação do tamanho funcional de um ambiente de *e-learning* possibilita uma análise de custo e benefício da aquisição de uma ferramenta ou da contratação do desenvolvimento de uma ferramenta customizada. O tamanho de um projeto de software também possibilita a derivação das estimativas de esforço, de custo e de prazo para o desenvolvimento do projeto. As estimativas de tamanho, esforço, prazo e custo constituem as principais estimativas a serem geradas, documentadas e acompanhadas em um processo efetivo de gestão de projetos de software. Este trabalho também apresenta uma aplicação dos modelos simplificados de derivação de estimativas de esforço, custo e prazo a partir do tamanho do projeto estimado em Pontos de Função para as funcionalidades do papel Aprendiz do Ambiente AulaNet.

3. Elaboração

A Contagem de Pontos de Função reflete as funcionalidades fornecidas pela aplicação para o usuário. As funcionalidades são avaliadas considerando a Visão do Usuário, que é uma descrição das funções aprovada pelo usuário, representando uma descrição formal das necessidades do negócio na linguagem do usuário [IFPUG, 2004]. A contagem de Pontos de Função considera os cinco tipos de funções descritos na seção anterior, a saber: Arquivo Lógico Interno, Arquivo de Interface Externa, Entrada Externa, Saída Externa e Consulta Externa (**Figura 2**). Esta seção tem como propósito mapear as funcionalidades descritas no “Manual do Aprendiz do AulaNet” [EDUWEB] nos tipos funcionais da APF. Uma vez que o requisito funcional da aplicação é identificado e mapeado em um tipo de função, este é então classificado em Simples, Médio ou Complexo de acordo com as tabelas de avaliação de complexidade descritas em [IFPUG, 2004]. O próximo passo é avaliar a contribuição para a contagem de Pontos de Função Não Ajustados, baseando-se no tipo funcional e na sua complexidade [IFPUG, 2004].

3.1. Arquivos Lógicos Internos

Um **Arquivo Lógico Interno (ALI)** é um grupo de dados ou informações de controle, identificado pelo usuário, mantido por meio de um ou mais processos elementares da aplicação sendo contada. Cada ALI possui uma complexidade funcional associada. Esta complexidade leva em consideração a quantidade de Registros Lógicos (subgrupos de dados) e de Itens de Dados (campos únicos e não repetidos) reconhecidos pelo usuário, contidos no ALI em questão [Hazan, 2000].

Os Arquivos Lógicos Internos podem ser identificados pela análise do modelo de dados ou diagrama de classes da aplicação. Devem ser consideradas as entidades fortes reconhecidas pelo usuário. As entidades de relacionamento são ALIs, apenas quando possuem atributos próprios. Quando não há um modelo de dados disponível, os ALIs podem ser identificados observando-se as funcionalidades da aplicação [Hazan, 2003b]. No AulaNet foram identificados e classificados os seguintes Arquivos Lógicos Internos (**Tabela I**):

TABELA I: Arquivos Lógicos Internos do AulaNet acessíveis ao papel de Aprendiz

Arquivo Lógico Interno	Descrição	Complexidade
Curso	Informações dos cursos e de turmas	SIMPLES
Departamento	Informações dos Departamentos	SIMPLES
Instituição	Informações da Instituição	SIMPLES
Participante	Informações cadastrais dos participantes.	SIMPLES
Papeis	Informações sobre os papeis do participante no curso	SIMPLES
Serviços	Informações de serviços configurados para os cursos	SIMPLES
Progresso – Participante	Dados de progresso do aprendiz em um curso específico, Itens visitados pelo participante em cada serviço	SIMPLES
Contato Professor	Informações de mensagens enviadas aos docentes	SIMPLES
Categoria	Informações sobre categorias das mensagens	SIMPLES
Mensagem Lista de Discussão	Requisitos de armazenamento das mensagens enviadas para a lista de discussão	SIMPLES
Discussão	Informações de controle sobre as listas de discussões	SIMPLES
Mensagens Conferência	Requisitos de armazenamento das mensagens enviadas para conferência e da avaliação das mensagens	SIMPLES
Conferencia	Informações de controle sobre as conferências	SIMPLES
Debate	Informações de controle sobre os Debates	SIMPLES
Participante – Debate	Informações sobre o desempenho do participante no debate	SIMPLES
Avisos	Informações sobre os avisos cadastrados para o curso	SIMPLES
Bibliografia	Informações sobre a Bibliografia relativa ao curso	SIMPLES
Webliografia	Informações sobre a referências aos <i>links</i> relativos ao curso	SIMPLES
Aulas	Informações sobre as aulas	SIMPLES
Conteúdo Aula	Informações sobre o conteúdo das aulas	SIMPLES
Anotação	Informações sobre as anotações de participantes para uma aula	SIMPLES
Tarefas	Informações sobre as tarefas	SIMPLES
Tarefas – Participante	Informações sobre uma tarefa específica de um participante e do desempenho do participante na tarefa	SIMPLES
Prova	Informações sobre as avaliações	SIMPLES
Questões	Informações sobre questões das avaliações	SIMPLES
Participante – Prova	Informações sobre o desempenho do participante em uma avaliação	SIMPLES
Co-Autoria	Informações sobre conteúdo com co-autoria de aprendizes	SIMPLES
Documentação	Informações sobre as documentações contidas no AulaNet	SIMPLES

A estratégia utilizada na identificação dos ALIs foi a de considerar as funções de dados do papel aprendiz como parte da aplicação AulaNet, e não como uma aplicação distinta. Assim, a identificação dos ALIs pode parecer inconsistente pelo fato de que os processos elementares que mantêm alguns ALIs identificados não aparecem na contagem de Entradas Externas. Por exemplo, o **ALI Bibliografia**, que contém dados das referências bibliográficas relativas ao curso, é mantido pelo papel **Docente**. No entanto, como estes dados são referenciados por funcionalidades deste módulo, por exemplo - **Lista de Bibliografia** (consulta das bibliografias armazenadas no ALI Bibliografia), considera-se correto identificá-los na Contagem das Funções de Dados.

3.2. Arquivos de Interface Externa

Um **Arquivo de Interface Externa** (AIE) é um grupo de dados logicamente relacionados ou informações de controle, identificado pelo usuário, referenciados pela aplicação, no entanto são mantidos por outra aplicação. O objetivo de um AIE é armazenar dados referenciados através de um ou mais processos elementares dentro da fronteira da aplicação que está sendo contada. Ou seja, um AIE de uma aplicação tem que ser um ALI de outra aplicação.

O AulaNet não referencia dados de outras aplicações, portanto não possui Arquivos de Interface Externa. Note que a definição da fronteira da aplicação e escopo da contagem é essencial para a Análise de Pontos de Função [Garmus, 2001]. Caso as funcionalidades do aprendiz fossem vistas como uma aplicação distinta, os ALIs atualizados por outros módulos da aplicação AulaNet seriam considerados Arquivos de Interface Externa.

É bastante comum que aplicações de *e-learning* corporativas utilizem os dados definidos em sistemas de controle de acesso da organização para acesso do aprendiz ao curso. Nesses casos, os dados de acesso (CPF, *login*, *password*, perfil) são considerados Arquivos de Interface Externa. Note que estes dados são referenciados pela aplicação de *e-learning* e são atualizados dentro da fronteira de outro sistema - o sistema de controle de acesso.

3.3. Entradas Externas

Uma **Entrada Externa** (EE) é um processo elementar que processa dados ou informações de controle que vem do lado de fora da fronteira da aplicação. O objetivo da Entrada Externa é manter um ou mais Arquivos Lógicos Internos e/ou alterar o comportamento do sistema. Quando há entrada de dados, pelo menos um ALI é atualizado. Por exemplo, no processo *Solicitar Registro no AulaNet*, o aprendiz preenche um formulário com seus dados cadastrais (nome, e-mail, descrição profissional, instituição, departamento, *homepage*, identificação e senha), estes dados são gravados no Arquivo Lógico que contém informações dos *Participantes*. No caso da entrada de informações de controle, nenhum Arquivo Lógico é atualizado, no entanto o comportamento da aplicação é alterado. Por exemplo, no processo *Trocar Mensagens - Enviar Mensagem Instantânea*, o aprendiz seleciona um aprendiz conectado ao ambiente e envia uma mensagem. Note que a mensagem não é gravada em nenhum Arquivo Lógico, esta funciona como uma informação de controle que altera o comportamento da aplicação, especificamente para o usuário receptor da mensagem.

Cada Entrada Externa possui uma complexidade funcional associada, baseando-se na quantidade de Arquivos Referenciados (Arquivos Lógicos Internos lidos ou mantidos e Arquivos de Interface Externa lidos durante o processo) e Itens de Dados (campos não repetidos, reconhecidos pelo usuário) [Hazan, 2000]. Note que os ambientes de *e-learning* possuem muitas funcionalidades de *Workflow*, como por exemplo enviar trabalho, corrigir trabalho, concluir correção, etc. Cada uma destas funcionalidades constitui um processo elementar da aplicação distinto, e portanto, deve ser considerada Entrada Externa.

Muitas vezes, as Entradas Externas possuem consultas implícitas para as alterações e exclusões de dados. Por exemplo, na funcionalidade Alterar Anotação, o aprendiz visualiza os dados das anotações para efetuar as alterações. Nestes casos, considera-se as duas funcionalidades: a Entrada Externa - *Alterar Anotação* e a Consulta Externa - *Consultar Anotação*. Note que se a consulta implícita da alteração for idêntica a da exclusão e/ou a consulta propriamente dita, a funcionalidade de consulta é contada apenas uma vez.

No AulaNet foram identificadas e classificadas as seguintes Entradas Externas (Tabela II):

TABELA II: Entradas Externas do AulaNet – Papel de Aprendiz

Entradas Externas	Descrição	Complexidade
Solicitar Registro no AulaNet	Permite ao aprendiz registrar-se no ambiente, preenchendo um formulário com dados cadastrais, que são gravados.	SIMPLES
Alterar Dados de Registro	Possibilita a alteração de dados cadastrais do usuário, tais como: e-mail, senha, <i>homepage</i> , etc.	SIMPLES
Pedir Matrícula	O aprendiz solicita matrícula em um curso. O arquivo de papeis é atualizado	COMPLEXA
Trancar Matrícula	O aprendiz seleciona o curso, o arquivo de papeis é atualizado	COMPLEXA
Enviar Mensagem Instantânea	O aprendiz seleciona um participante conectado no ambiente para enviar uma mensagem. O processo elementar, consiste na entrada de informações de controle	SIMPLES
Enviar Mensagem Conferência	Possibilita ao aprendiz enviar uma mensagem para a conferência	COMPLEXA
Entrar no Debate	O aprendiz entra no Debate, entrada de informações de controle	SIMPLES
Enviar Mensagem no Debate	Aprendiz envia uma mensagem para o Debate. Entrada de informações de controle	SIMPLES
Incluir Anotação	O Aprendiz inclui anotações sobre uma aula específica	SIMPLES
Alterar Anotação	O Aprendiz altera anotações sobre uma aula específica	SIMPLES
Excluir Anotação	O Aprendiz exclui anotações sobre uma aula específica	SIMPLES
Submeter Arquivo	O aprendiz inclui um arquivo para resolver a tarefa	SIMPLES
Remover Arquivo	O aprendiz exclui um arquivo submetido para tarefa	SIMPLES
Definir item inicial	Permite ao Aprendiz definir o item inicial em uma tarefa	SIMPLES
Resolver Avaliação	Permite ao Aprendiz resolver as questões da avaliação	COMPLEXA

3.4. Saídas Externas

Uma **Saída Externa (SE)** é um processo elementar que envia dados ou informação de controle para fora da fronteira da aplicação. O seu objetivo principal é apresentar informação através de processamento lógico adicional à recuperação de dados ou informação de controle. O processamento deve conter no mínimo um cálculo, ou criar de dados derivados, ou manter um ou mais ALIs ou alterar o comportamento do sistema. Cada função identificada como SE possui uma complexidade funcional associada, baseando-se na quantidade de Arquivos Referenciados (ALIs lidos ou mantidos e AIEs lidos) e Itens de Dados (campos não repetidos, reconhecidos pelo usuário) [Hazan, 2000].

No AulaNet todos os relatórios foram identificados como Saídas Externas por apresentarem valores calculados ou dados derivados. Este comportamento também é esperado para as outras aplicações de *e-learning*, devido aos usuários destes ambientes (Administrador, Coordenador, Docente Co-autor, Mediador e Aprendiz) terem necessidades de visualizar informações consolidadas. Algumas consultas também foram identificadas como Saídas Externas, pois contêm informações derivadas. Por exemplo, a *Consulta de Participantes no Debate*, identifica os moderadores com um símbolo ao lado do nome exibido.

No AulaNet foram identificadas e classificadas as seguintes Saídas Externas (Tabela III):

TABELA III: Saídas Externas do AulaNet – Papel de Aprendiz

Saídas Externas	Descrição	Complexidade
Entrar no Ambiente AulaNet	Permite entrar no AulaNet, que disponibilizará uma lista de opções de acordo com o perfil do participante	MÉDIA
Consumo do Curso no AulaNet	O AulaNet apresenta as informações gerais do curso e o controle remoto, contendo os serviços do curso	COMPLEXA
Contato com os Docentes	Possibilita o contato do aprendiz com o docente. O ambiente registra as mensagens trocadas	MÉDIA
Lista de Mensagens enviadas para Docentes	Apresenta uma lista de mensagens trocadas, contendo o total de mensagens enviadas	MÉDIA
Lista de Participantes	Apresenta uma lista de membros da turma, identificando quem está conectado, através de esfera verde ou vermelha	SIMPLES
Enviar Mensagens para Lista de Discussão	Possibilita o envio de mensagens para todos os membros do curso. As mensagens enviadas são armazenadas	COMPLEXA
Lista de Mensagens da Lista de Discussão	O AulaNet disponibiliza uma lista das mensagens enviadas para Lista de Discussão contendo o total de mensagens	COMPLEXA
Listas das Conferências	O AulaNet disponibiliza uma lista das conferências	MÉDIA
Consulta da Conferência	O AulaNet apresenta as mensagens trocadas em uma conferência específica	COMPLEXA
Consulta de Participantes no Debate	Apresenta uma lista de participantes ativos no Debate, identificando os moderadores	SIMPLES
Lista de Aulas	Apresenta as aulas disponíveis para o curso	MÉDIA
Lista de Conteúdos	Apresenta uma lista de conteúdos para uma aula específica	COMPLEXA
Lista de Tarefas	Apresenta uma lista de tarefas selecionadas para a turma	MÉDIA
Lista de Avaliações	Permite ao aprendiz visualizar as avaliações	COMPLEXA
Lista de Webbliografia	Apresenta uma lista com links para sites da internet	SIMPLES
Lista de Documentos	Apresenta uma lista das documentações do curso	SIMPLES
Lista de Conteúdos da Documentação	Apresenta uma lista de conteúdos da documentação selecionada	SIMPLES
Relatório por Participante	Conceito Médio dos Participantes em todos os serviços	COMPLEXA
Relatório por Participante	Porcentagem média de contribuição nos serviços	COMPLEXA
Relatório por Participante	Apresenta dados sobre Contato com os Docentes	SIMPLES
Relatório por Participante	Apresenta dados sobre Listas de Discussão	SIMPLES
Relatório por Participante	Apresenta dados sobre Conferências	COMPLEXA
Relatório por Participante	Apresenta dados sobre Debates	COMPLEXA
Relatório por Participante	Apresenta dados sobre Tarefas	COMPLEXA
Relatório por Participante	Apresenta dados sobre Avaliação	COMPLEXA
Relatório por Participante	Apresenta dados sobre Co-Autoria de Aprendiz	MÉDIA
Relatório por Categoria	Apresenta dados sobre Contato com os Docentes	MÉDIA
Relatório por Categoria	Apresenta dados sobre Lista de Discussão	MÉDIA
Relatório por Categoria	Apresenta dados sobre Conferências	MÉDIA
Relatório por Serviços	Sumário de contribuições dos participantes em serviços	COMPLEXA
Relatório por Serviços	Contribuições detalhadas dos participantes em serviços	COMPLEXA

3.5. Consultas Externas

Uma **Consulta Externa (CE)** é um processo elementar que envia dados ou informação de controle para fora da fronteira da aplicação. O seu objetivo principal é apresentar informação para o usuário através da recuperação de dados ou informação de controle de um Arquivo Lógico Interno ou Arquivo de Interface Externa. O processamento lógico de uma CE não contém fórmulas matemáticas ou cálculos, não cria dados derivados, não mantém ALIs e não altera o comportamento do sistema. Cada Consulta Externa possui uma complexidade funcional associada, baseando-se na quantidade de Arquivos Referenciados (Arquivos Lógicos Internos e Arquivos de Interface Externa lidos durante o processo) e Itens de Dados (campos não repetidos, reconhecidos pelo usuário) [Hazan, 2000].

As Consultas Externas identificadas no AulaNet estão apresentadas **Tabela IV**. Note que as funções *list-boxes* são contadas apenas uma vez, mesmo que estas apareçam em várias telas da aplicação. Como exemplo de *list-boxes*, tem-se as funções: *Consulta da Instituição – List Box*, *Consulta de Departamento –List Box* e *Consulta de Categoria de Assuntos – List Box (Tabela IV)*. É importante destacar que as *list-boxes* que recuperam dados de tabelas estáticas (não consideradas ALIs ou AIEs) não são identificadas na contagem de PF. Note que nenhuma *list-box*, ilustrada na **Figura 3** foi considerada Consulta Externa, pois estas funções não recuperam dados de Arquivos Lógicos (ALIs ou AIEs).

O formulário apresenta os seguintes campos:

- E-mail:
- Sistema Operacional:
- Seu browser:
- Velocidade de Conexão:
- Área do ambiente:
- Serviço onde ocorre o problema:
- O problema se repete?:
- Data de ocorrência do problema (dd/mm/aaaa):
- Descrição, em detalhes, do problema:

Figura 3: Enviar Dados de Problemas para o Coordenador do Curso

Note que a funcionalidade de *help* não foi contada. Embora, esta funcionalidade possa parecer uma Consulta Externa, seguindo rigidamente o manual de práticas de contagem (CPM 4.2) [IFPUG, 2004], observa-se que toda CE deve recuperar dados de um ou mais Arquivos Lógicos (ALIs ou AIEs). Assim, as funcionalidades de *help* não devem ser contadas, a menos que os dados apresentados no *help* possam ser mantidos por meio de um processo elementar da aplicação.

Uma funcionalidade bastante comum em aplicações de *e-learning* é o uso do correio eletrônico. O envio de e-mail sem atualização de ALIs, por exemplo *enviar e-mail para o coordenador*, é uma Consulta Externa. Caso algum ALI seja atualizado durante o processo de envio de e-mail, por exemplo *Contato com Docentes*, o processo é considerado uma SE.

TABELA IV: Consultas Externas do AulaNet - Papel de Aprendiz

Consultas Externas	Descrição	Complexidade
O que é AulaNet?	Apresenta um tutorial sobre o AulaNet	SIMPLES
Contato	Permite o envio de mensagens para a administração do ambiente	SIMPLES
Dicas	Esta funcionalidade atende ao requisito de Help de tela	SIMPLES
Enviar e-mail para o Coordenador	Recupera e-mail do coordenador, chama outlook para o envio do e-mail	SIMPLES
Listar Cursos Oferecidos	Lista todos os cursos oferecidos no ambiente AulaNet	COMPLEXA
Consulta do Curso	Apresenta detalhes de um curso selecionado na lista	SIMPLES
Problemas	Permite relatar um erro ou mau funcionamento do ambiente para o administrador	SIMPLES
Consulta da Instituição – List box	Lista das Instituições. Trata-se de uma recuperação de dados do Arquivo Lógico Interno Instituições	SIMPLES
Consulta de Departamento - List Box	Lista dos Departamentos. Trata-se de uma recuperação de dados do Arquivo Lógico Interno Departamentos	SIMPLES
Recuperar Senha	Solicita o e-mail do aprendiz e envia um e-mail, contendo a senha e a identificação do aprendiz	SIMPLES
Consulta da Alteração do Registro	Apresenta informações do Registro do Aprendiz para serem alteradas na Função “Alterar Registro”	SIMPLES
Lista de Cursos (Pedir Matricula)	O AulaNet apresenta uma lista dos cursos disponíveis para matrículas no ambiente	MÉDIA
Consulta de Curso (Pedir Matricula)	O usuário seleciona o curso no qual deseja se matricular e a ferramenta mostra informações do curso	COMPLEXA
Lista de Cursos (Trancar Matricula)	O AulaNet apresenta uma lista dos cursos que o aprendiz está matriculado	MÉDIA
Consulta de Mensagens enviadas para Docentes	Aprendiz consulta detalhes de uma mensagem específica enviada para os Docentes	MÉDIA
Consulta Dados do Participante	Apresenta os dados do Participante selecionado	SIMPLES
Consulta de Mensagens da Lista de Discussões	Aprendiz consulta detalhes de uma mensagem específica enviada para a Lista de Discussão	MÉDIA
Consulta Categoria-Assuntos – Listbox	Lista das Categorias da mensagem. Trata-se de uma recuperação de dados do Arquivo Lógico Interno Categorias	SIMPLES
Consulta de Mensagens da Conferência	Apresenta os dados de uma mensagem específica enviada para conferência	MÉDIA
Salvar Conteúdo do Debate	O conteúdo do debate é enviado para o correio eletrônico do participante	SIMPLES
Lista de Avisos Enviados	Apresenta uma lista com todos os avisos enviados	SIMPLES
Consulta Conteúdo da Aula	O AulaNet apresenta informações sobre o conteúdo selecionado na lista de conteúdos da aula	SIMPLES
Consulta de Anotação	Apresenta uma anotação feita para uma aula específica	SIMPLES
Consulta de Tarefa	Apresentar informações sobre uma tarefa específica	MÉDIA
Consulta Avaliação	Permite ao aprendiz consultar o conteúdo da avaliação	SIMPLES
Lista de Bibliografia	Apresenta uma lista de bibliografias	SIMPLES
Consulta Webliografia	Apresenta o conteúdo de um link da Webliografia	SIMPLES
Lista de Conteúdos	Apresenta uma lista de todos conteúdos do curso	SIMPLES
Download de Arquivos	Possibilita o Download de um conteúdo específico	SIMPLES

Outra funcionalidade bastante comum em ambientes de e-learning é o *download*. Existem duas formas de classificar o *download*, a saber: Caso o processo apenas recupere a informação e salve na estação de trabalho do usuário, este é considerado uma CE, como exemplo tem-se a função *download* de Arquivos do AulaNet (**Tabela IV**). Caso o *download* além de recuperar a informação apresente algum dado derivado, por exemplo uma barra de progresso, apresentando quanto já foi recuperado e qual o percentual que falta ser recuperado, então este é considerado uma SE. É importante destacar que o recurso de barra de progresso deve ser projetado especificamente para a aplicação. As funcionalidades fornecidas pelo *Browser* (*Internet Explorer, Netscape, Mozila*) não fazem parte do escopo da contagem da aplicação.

3.6. Cálculo dos Pontos de Função Não Ajustados

A quantidade de Arquivos Lógicos Internos, Arquivos de Interface Externa, Entradas Externas, Consultas Externas e Saídas Externas, assim como a complexidade funcional associada, determinam a contribuição das funções para a contagem de Pontos por Função Não Ajustados. A **Tabela V** ilustra a Contagem de Pontos de Função Não Ajustados do AulaNet, consolidando os resultados das **Tabelas I, II, III e IV**.

TABELA V: Cálculo de Pontos por Função Não Ajustados

TIPO DE FUNÇÃO	COMPLEXIDADE/ CONTRIBUIÇÃO FUNCIONAL				TOTAL
Arquivo Lógico Interno	Simple	28	X 7	196	196
	Média	0	X 10	0	
	Complexa	0	X 15	0	
Arquivo de Interface Externa	Simple	0	X 5	0	0
	Média	0	X 7	0	
	Complexa	0	X 10	0	
Entrada Externa	Simple	11	X 3	33	57
	Média	0	X 4	0	
	Complexa	4	X 6	24	
Saída Externa	Simple	7	X 4	28	176
	Média	10	X 5	50	
	Complexa	14	X 7	98	
Consulta Externa	Simple	21	X 3	63	99
	Média	6	X 4	24	
	Complexa	2	X 6	12	
Total de Pontos de Função não Ajustados					528

Note que as funções de dados (Arquivos Lógicos Internos e Arquivos de Interface Externa) constituem as funções com maior contribuição funcional na APF. Analisando-se as funcionalidades do aprendiz (**Tabela V**) pode-se constatar que a quantidade de Consultas Externas - 29 CEs e Saídas Externas - 31 SEs é bem maior que a quantidade de Entradas Externas - 15 EEs o que era esperado pelo perfil das funcionalidades deste módulo. Ou seja, as funcionalidades de apresentação de informações predominam em relação às de processamento de dados ou informações de controle.

O *International Software Benchmarking Standards Group* (ISBSG) [ISBSG] é uma organização sem fins lucrativos que mantém um repositório contendo indicadores de projetos de *software* internacionais para ajudar na gerência de recursos de Tecnologia da Informação (TI), visando a melhoria das estimativas de projetos, análise de riscos, análise de produtividade e *Benchmarking*. O banco de dados do ISBSG de 2003 [ISBSG] apresenta as seguintes estatísticas para distribuição percentual de funcionalidades de uma aplicação:

- Arquivos Lógicos Internos: 23,5%
- Arquivos de Interface Externa: 8,1%
- Entradas Externas: 29,3%
- Consultas Externas: 14,8%
- Saídas Externas: 24,3%

Como foi dimensionado apenas um módulo do AulaNet não é possível analisarmos a distribuição de funcionalidades da aplicação como um todo. Se o módulo do aprendiz fosse considerado uma aplicação independente, muitos dos Arquivos Lógicos Internos seriam considerados Arquivos de Interface Externa, devido a estes grupos de dados serem mantidos em outros módulos da aplicação. Note que este módulo é mais voltado para consultas (CEs, SEs) do que atualização de base de dados (EEs), por tratar-se de funcionalidades do aprendiz. Isto é refletido na contagem de Pontos Função, tem-se 60 SEs e CEs e apenas 15 EEs.

3.7. Cálculo do Fator de Ajuste

O Fator de Ajuste influencia a Contagem de Pontos de Função não Ajustados em +/- 35% para obter-se os Pontos por Função Ajustados. O fator de ajuste é baseado nas **14 Características Gerais dos Sistemas** (CGS) que avaliam a funcionalidade geral da aplicação (requisitos não funcionais). Cada característica, descrita na **Tabela VI**, possui um Nível de Influência (NI) que varia em uma escala de zero (nenhuma influência) a cinco (forte influência). Para o cálculo do Fator de Ajuste deve-se somar os NIs de cada característica, obtendo-se Nível de Influência Total (NIT). O próximo passo é aplicar a seguinte fórmula:

$$\text{Fator de Ajuste} = (\text{NIT} \times 0,01) + 0,65$$

Embora, o Fator de Ajuste não seja atualizado pelo IFPUG desde a década de 80, não seja reconhecido pela ISO [Dekkers, 2003] e algumas empresas tenham desconsiderado. A decisão foi a de seguir o manual CPM 4.2 [IFPUG, 2004], considerando o cálculo fator de ajuste. A experiência da autora em Contagem de Pontos de Função de mais de 500 sistemas ao longo de 7 anos tem demonstrado que sistemas Batch ficam com um Fator de Ajuste Baixo (entre 0,8 e 0,9), em Sistemas *Mainframe on-line* ou *micro Stand Alone* o fator de ajuste fica em torno 1 não exercendo impacto na contagem dos Pontos de Função não ajustados, os Sistemas Cliente Servidor e Web possuem fator de ajuste mais altos (entre 1,1 a 1,2). Alguns sistemas Cliente Servidor mais complexos apresentaram fator de ajuste maior que 1,25. Como o AulaNet é uma aplicação Cliente/Servidor com requisitos não funcionais (ou características da qualidade) bastante rigorosos, por tratar-se de um produto comercial, espera-se um fator de ajuste maior que 1,25.

É importante destacar que a aferição do Fator de Ajuste tem apoiado a elicitação e a análise (revisão) de requisitos não-funcionais e até mesmo de requisitos funcionais. No entanto, como o cálculo do Fator de Ajuste é uma atividade difícil de ser realizada e demorada, especialmente para os profissionais sem experiência em contagem de PF, muitas empresas fixam um Fator de Ajuste para todas as aplicações na mesma plataforma.

A dificuldade do cálculo do fator de ajuste ocorre devido à grande quantidade de informações (14 características) a serem analisadas em sua aferição. É importante destacar que nova versão do Manual de Práticas de Contagem - CPM 4.2, publicada em 2004, enfatizou a aferição do fator de ajuste, fornecendo dicas para apoiar a identificação do nível de influência de cada Característica Geral do Sistema (CGS) [IFPUG, 2004].

TABELA VI: Características Gerais do Sistema [Hazan, 2000] [Garmus, 2001]

CARACTERÍSTICA	DEFINIÇÃO
Comunicação de Dados	Descreve o grau em que a aplicação comunica-se diretamente com o processador.
Processamento de Dados Distribuído	Descreve o grau em que a aplicação transfere dados entre seus componentes.
Desempenho	Descreve o grau em que considerações de tempo de resposta e <i>throughput</i> influenciam no desenvolvimento da aplicação.
Configuração Utilizada	Descreve o grau em que restrições de recursos computacionais influenciam no desenvolvimento da aplicação.
Volume de Transações	Descreve o grau em que o volume de transações do negócio influencia no desenvolvimento da aplicação.
Entrada de Dados <i>On-Line</i>	Descreve o grau em que os dados entram na aplicação através de transações interativas (<i>on-line</i>).
Eficiência do Usuário Final	Eficiência do Usuário Final descreve o grau das considerações sobre a facilidade de uso para o usuário da aplicação medidas.
Atualização <i>on-line</i>	Descreve o grau em que os ALIs são atualizados <i>on-line</i> .
Processamento Complexo	Descreve o grau em que o processamento lógico influencia o desenvolvimento da aplicação.
Reusabilidade	Descreve o grau em que a aplicação e o seu código fonte têm sido especificamente projetados, desenvolvidos, e suportados para serem reusados por outras aplicações.
Facilidade de Implantação	Descreve o grau em que a conversão de ambientes anteriores influencia no desenvolvimento da aplicação.
Facilidade Operacional	Descreve o grau em que a aplicação atende aos aspectos operacionais, tais como processos de inicialização, <i>backup</i> e <i>recovery</i> .
Múltiplos Locais	Descreve o grau em que a aplicação tem sido desenvolvida para executar em múltiplos locais ou organizações do usuário.
Facilidade de Mudanças	Facilidade de Mudança descreve o grau em que a aplicação tem sido desenvolvida para facilitar a modificação do processamento lógico ou estrutura de dados.

O Fator de Ajuste do AulaNet, apresentado na **Tabela VII**, é de 1,27. Isto significa que os Pontos de Função Não Ajustados do AulaNet são ajustados em 27% para cima com a aplicação o Fator de Ajuste. O gráfico da **Figura 4** fornece uma visão comparativa da avaliação das características do AulaNet. É importante ressaltar que a fronteira estabelecida no cálculo do Fator de Ajuste considerou toda a aplicação AulaNet, não apenas as funcionalidades descritas no Manual do Aprendiz, assim o fator de ajuste refere-se à aplicação AulaNet como um todo.

TABELA VII: Cálculo do Fator de Ajuste do AulaNet

Características Gerais dos Sistemas	NI	JUSTIFICATIVA
1 Comunicação de Dados	4	A Aplicação é mais que um <i>front-end</i> , mas suporta apenas um protocolo de comunicação TCP/IP.
2 Processamento Distribuído	4	O processamento é distribuído com transferência de dados on-line em ambas as direções. (Cliente-Servidor).
3 Desempenho	5	Os requisitos de performance estabelecidos são rigorosos, foi utilizada ferramenta de <i>stress</i> de carga.
4 Utilização do Equipamento	5	Existem restrições especiais da aplicação sobre os componentes distribuídos. Além de um <i>Webserver</i> , são utilizados outros servidores (SMTP, Java, <i>Chat</i> e Lista, SGBD e <i>Real Player</i>).
5 Volume de Transações	5	Os requisitos de performance estabelecidos são rigorosos, foi utilizada ferramenta de stress de carga.
6 Entrada de Dados On line	5	Aplicação on-line.
7 Eficiência do Usuário Final	4	Os requisitos de usabilidade são rigorosos. A aplicação minimiza a digitação do usuário.
8 Atualização On line	3	Atualização on-line dos Arquivos Lógicos Internos.
9 Processamento Complexo	5	Segurança; Cálculos Estatísticos; Suporte a entradas/saídas Multimídia; Validações e tratamento de exceções.
10 Reutilização do Código	5	A aplicação foi desenvolvida para facilitar o reuso, sendo customizada para uso através de parâmetros que podem ser atualizados pelo usuário.
11 Facilidade de Implantação	4	A aplicação possui ferramenta automática para instalação. Não há requisitos de conversão de dados.
12 Facilidade Operacional	3	A aplicação minimiza a montagem de fitas magnéticas e manuseio de formulários. Existem processos de backup definidos com a intervenção do operador (DBA).
13 Múltiplos Locais	5	Um plano de documentação e suporte foi elaborado e testado para suportar a aplicação em ambientes de <i>hardware</i> e <i>software</i> diferentes.
14 Facilidade de Mudanças	5	Atualização on-line de parâmetros de configuração da interface. Consultas e relatórios parametrizados.
Nível de Influência Total (NIT)	62	Fator de Ajuste = $(0,01 \times 62) + 0,65 = 1,27$

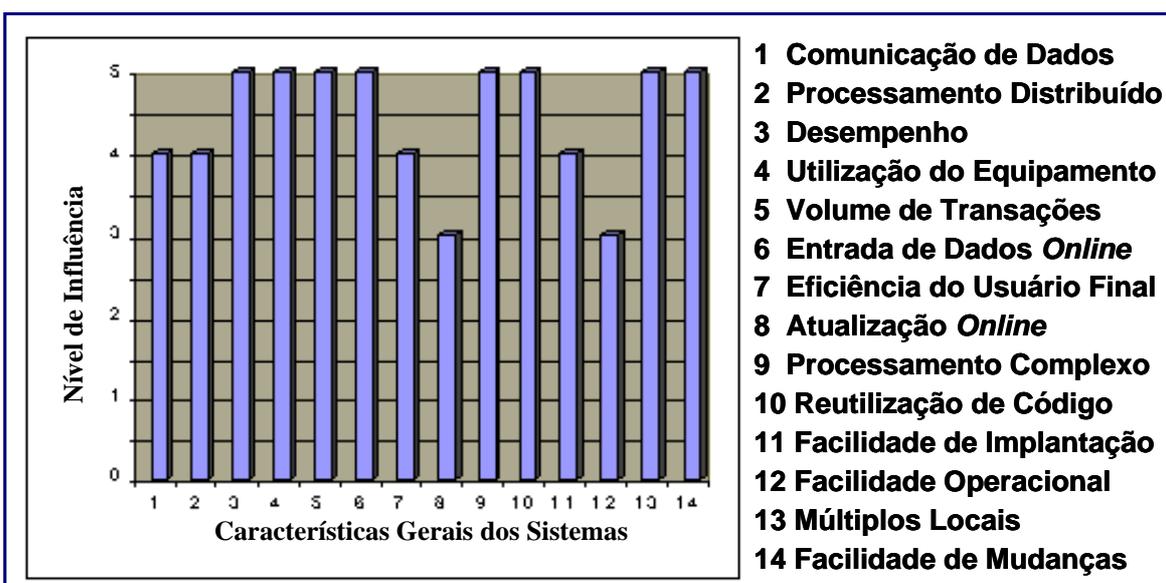


Figura 4: Avaliação das 14 Características Gerais do Sistema do AulaNet

O Cálculo de Pontos de Função Ajustados das funcionalidades do aprendiz do AulaNet é realizado segundo a fórmula:

$$PF \text{ Aplicação} = PF \text{ Não Ajustados} \times \text{Fator de Ajuste}$$

Assim, tem-se $PF = 528 \times 1,27 = 670,56$. Ou seja, esta parte da aplicação AulaNet possui **671 Pontos de Função Ajustados**. A indústria tem considerado aplicações com mais de 500 Pontos de Função como complexas [Jones, 1997].

4. Discussão e contribuições

Esta seção tem como propósito apresentar uma análise gerencial do resultado da contagem de Pontos de Função do AulaNet, descrita na seção 3. O objetivo é mostrar como a partir de uma estimativa de tamanho deriva-se as estimativas de esforço e de prazo de um projeto de software.

O AulaNet utiliza o JAVA como principal ferramenta de desenvolvimento. A produtividade média do JAVA fica em torno de 15 a 20 horas/PF [BFPUG] [ISBSG]. Considerando a produtividade de 15 horas/PF conclui-se que o desenvolvimento das funcionalidades do AulaNet voltadas especificamente para o papel aprendiz levariam em média 10.065 horas para serem desenvolvidas. Ou seja, seriam necessários 76,3 homem-meses para desenvolver esta parte do AulaNet, considerando um mês com 22 úteis e um dia com 6 horas de produtividade efetiva [Jones, 1997].

Jones propõe uma fórmula para o cálculo do melhor tempo de desenvolvimento, denominado Td, e da Região Impossível (RI) de desenvolvimento, descrita em [Jones, 1998]. O Td não é o menor tempo de desenvolvimento e sim o tempo onde há melhor relação custo e benefício entre recursos alocados ao projeto e o tempo de desenvolvimento [Hazan, 2004]. Na Região Impossível (RI), a adição de mais recursos ao projeto não implicará em redução no prazo. O cálculo da região impossível é utilizado para apoiar a elaboração de cronogramas. O cálculo do Td considera o volume em PF (V) e um expoente t ($Td = V^{**} t$). Para aplicações comerciais $t = 0,40$. Então, para o módulo do Aprendiz do AulaNet, $Td = 671^{**} 0,40 = 13,5$ meses. A Região Impossível encontra-se a 75% de Td, assim $RI = 0,75 \times 13,5 = 10,1$. Em uma análise empírica, seria impossível uma empresa desenvolver uma ferramenta de e-learning comercial contendo apenas as funcionalidades descritas no manual do aprendiz do AulaNet em menos de 10 meses. O comprometimento com o prazo menor que 13,5 meses representaria um alto risco.

Então, considerando o esforço de 76,3 homens_messes para o término do desenvolvimento em 13,5 meses, deve alocar-se ao projeto uma equipe de 5 a 6 recursos. No entanto, supondo a construção de uma ferramenta de *e-learning* corporativa comum para uma organização com as funcionalidades similares as do AulaNet (aprendiz), utiliza-se o expoente t de 0,35 (expoente máximo para aplicações), e então o $Td = 9,8$. Utilizando as fórmulas do modelo COCOMO II [Boehm, 2000], com os valores nominais, obtém-se um $Td = 9,34$. Assim, a alocação ideal de recursos ao projeto seria de 8 recursos para obter-se um prazo de 9,5 meses.

É importante destacar que os dados de prazo e de esforço apresentados estão baseados em *Benchmarking*, visto que não existe um banco de dados histórico disponível, contendo informações sobre tamanho e esforço de ambientes de *e-learning* desenvolvidos nas mesmas condições do AulaNet.

Em caso de projetos subcontratados, as estimativas de custo são obtidas multiplicando-se o total de Pontos de Função Ajustados pelo preço por Ponto de Função. Para os projetos internos, deve-se elaborar um cronograma, e então calcular-se o custo da hora dos profissionais alocados às tarefas inerentes ao projeto. Nos dois casos, é importante considerar-se outros custos do projeto, tais como: viagens, treinamentos, consultoria, aquisição de *Hardware* ou *Software*, etc [Hazan, 2004]. As práticas descritas na área de processo de Planejamento do Projeto de Software do modelo CMMI [SEI, 2002] também levam em consideração as estimativas de Recursos Computacionais Críticos. Nestas estimativas devem ser considerados os recursos de *Hardware* que precisam estar disponíveis em uma determinada fase do desenvolvimento do projeto. Por exemplo, um ambiente de teste específico, espaço em disco para um sistema entrar em produção, etc.

5. Conclusões

Com a expansão do ensino a distância pela Web, a tendência é que os ambientes de *e-learning* sofram manutenções evolutivas para atender as novas demandas de funcionalidades dos usuários. Portanto, a gestão de projetos de desenvolvimento e de manutenção das ferramentas de *e-learning* constituem importantes temas de pesquisa para trabalhos futuros. É importante destacar que a métrica de Pontos de Função constitui um dado de normalização para a construção de indicadores gerenciais, que por sua vez são considerados ferramentas estratégicas para uma gestão efetiva de projetos [McGarry, 2001].

O modelo conceitual composto de componentes proposto por Lucena [Lucena, 2003] apoia a construção de novos ambientes de *groupware* e a implementação de novas funcionalidades nos ambientes existentes. O modelo facilita a identificação das funcionalidades, fornecendo suporte à medição do tamanho do ambiente por meio do método Análise de Pontos por Função. Note que a métrica de Pontos de Função é baseada nos requisitos funcionais da aplicação, sempre levando em consideração a visão do usuário. Assim, o método pode ser aplicado nas fases iniciais (requisitos) dos projetos de software.

Este trabalho também é relevante para cursos de APF ministrados na indústria e nas Universidades, nas disciplinas de Engenharia de Software, pois apresenta um exemplo real de Contagem de Pontos de Função de um *learningware*, que pode ser obtido e estudado (o AulaNet é *freeware* e está disponível em www.eduweb.com.br). A contagem descrita neste trabalho foi realizada pela autora, que atua como Consultora de Métricas há 7 anos e possui a certificação de especialista em Pontos de Função CFPS (*Certified Function Point Specialist*) [IFPUG], em conjunto com desenvolvedores e usuários do ambiente AulaNet.

Também é importante destacar a utilização da técnica Análise de Pontos por Função na garantia da completeza da especificação de requisitos. Uma contagem de Pontos de Função é realizada usando uma descrição suficientemente formal das necessidades do negócio na linguagem do usuário. Assim, a contagem pode servir também como uma revisão dos requisitos [Hazan, 2003a]. Este processo simplificado de inspeção de requisitos, baseado na contagem de Pontos de Função, possibilita a identificação de defeitos, quando estes ainda são baratos de se corrigir – na fase de requisitos. Segundo Krutchen [Krutchen, 1998], as inconsistências não detectadas na fase de requisitos constituem uma das maiores classes de problemas encontrados em projetos de software. Erros inseridos na fase de requisitos gastam mais tempo para serem encontrados e corrigidos do que os erros inseridos nas fases posteriores [Lutz, 1993]. Então, os defeitos introduzidos ao longo do processo de desenvolvimento de software devem ser identificados o quanto antes, se possível na fase em que foram produzidos [Jones, 1997] [Kristen, 1996].

Futuramente pretende-se expandir o trabalho de contagem de Pontos de Função para toda a ferramenta AulaNet e gerar um Banco de Dados histórico de produtividade. A principal vantagem desta base de dados será o apoio no planejamento das futuras manutenções evolutivas do AulaNet. Além disso, estes dados serão úteis para as organizações e universidades interessadas em desenvolver ferramentas de *e-learning* estimarem o esforço e o custo do projeto de desenvolvimento, bem como alocação de recursos.

Também recomenda-se como trabalho futuro o dimensionamento de outros ambientes de *e-learning* utilizando a métrica de Pontos de Função. Isto tornará possível a realização de *Benchmarking* de produtividade e qualidade do processo de desenvolvimento. Note que este trabalho será de grande utilidade para identificar-se as tecnologias de implementação, bem como o tipo de processo de desenvolvimento, com maior produtividade para construção de ambientes de *e-learning*.

Referências bibliográficas

Todos os sites foram consultados em 04 de março de 2005.

- [Andrade, 2004] Andrade, E. *Pontos de Casos de Uso e Pontos de função na gestão de estimativa de tamanho de projetos de softwares orientados a objetos*. Tese de Mestrado em Gestão do Conhecimento e Tecnologia da Informação, Universidade Católica de Brasília, 2004.
- [BFPUG] BFPUG: Brazilian Function Point Users Group. www.bfpug.com.br
- [Boehm,, 2000] Boehm, B. *Software Cost Estimation With COCOMO II*. Prentice Hall, New Jersey, 2000.
- [Cunha, 2003] Cunha, G. and Almeida, E. *Estimativa de projetos com base em Casos de Uso*. Simpósio Internacional de Melhoria de Processos de Software (SIMPROS), Recife, 2003 pp.53-60.
- [Damodaran] Damodaram, M and Washington, A. *Estimation using use case points*. Computer Science Program. Texas -Victoria: University of Houston. www.bfpug.com.br/Artigos/UCP/Damodaran-Estimation_Using_Use_Case_Points.pdf
- [Dekkers, 2002] Dekkers , C. *How Function Points Support the Capability Maturity Model Integration*. CrossTalk, February 2002.
- [Dekkers, 2003] Dekkers , C. *Measuring the logical or function a Size of Software Projects and Software Application*. Spotlight Software, ISO Bulletin May 2003.
- [EDUWEB] EDUWEB:
http://www.eduweb.com.br/portugues/elearning_tecnologia.asp
- [Fenton, 2002] Fenton, N., Neil, M. *Software Metrics: Roadmap*. in Finkelstein, A. (ed.) *The Future of Software Engineering*, ACM Press, p. 359-370, 2002.
- [Fuks, 1999] Fuks, H AT All. *Communication, Coordination and Cooperation in Distance Education*. Proceedings of AMCIS99 – Americas Conference on Information Systems, Milwaukee, USA, 1999 pp.130-132.
- [Fuks, 2000a] Fuks, H. *Aprendizagem e Trabalho Cooperativo no Ambiente AulaNet*. Monografias em Ciência da Computação n° 11/00, Departamento de Informática – PUC-Rio, 2000.

- [Fuks, 2000b] Fuks, H., Raposo, A.B. and Gerosa, M. A. *Engenharia de Groupware: Desenvolvimento de Aplicações Colaborativas*. in Anais da XXI Jornada de Atualização de Informática (JAI), Capítulo 3, 2002.
- [Fuks, 2002] Fuks, H., Raposo, A. B. and Gerosa, M. A. *Engineering Groupware for E-Business*. 1st Seminar on Advanced Research in Electronic Business (EBR'2002), Novembro 2002, Rio de Janeiro.
- [Garmus, 2001] Garmus, D. and Heron, D. *Function Point Analysis: Measurement Practices for Successful Software Projects*. Addison Wesley, 2001.
- [Hazan, 1999] Hazan, C. *Metodologia para o Uso de Indicadores na Gerência de Projetos de Desenvolvimento de Software*. Tese de Mestrado, IME, Rio de Janeiro, 1999.
- [Hazan, 2000] Hazan, C. *Análise de Pontos por Função: Uma Abordagem Gerencial*. Jornada de Atualização de Informática (JAI), Curitiba, 2000 .
- [Hazan, 2003a] Hazan, C. *Análise de Pontos por Função: Uma ferramenta na implantação do Modelo CMM*. Revista Tema, TemaTec, SERPRO, Março/2003.
- [Hazan, 2003b] Hazan, C. *Análise de Métodos para Geração de Estimativas de Tamanho de Projetos de Software*. Simpósio de Desenvolvimento e Manutenção de Software da Marinha (SDMS2003), Rio de Janeiro, 2003.
- [Hazan, 2003c] Hazan, C. and Leite, J. *Definição de Indicadores Gerenciais de Requisitos*. 6th International Workshop on Requirements Engineering (WER2003), Piracicaba, Brazil, November 2003, pp 285-301.
- [Hazan, 2004] Hazan, C., Staa, A. *Análise e Melhoria de um Processo de Estimativas de Tamanho de Projetos de Software*. VI Simpósio de Melhoria do Processo de Software - SIMPROS, São Paulo 2004.
- [IFPUG, 2004] IFPUG . *Counting Practices Manual*. Version 4.2, 2004.
- [IFPUG] IFPUG: International Function Point Users Group. <http://www.ifpug.org>
- [ISBSG] ISBSG: International Software Benchmarking Standard Group. <http://www.isbsg.org.au>
- [Jones, 1997] Jones, C. *Applied Software Measurement: Assuring Productivity and Quality*. Prentice Hall, Second Edition, 1997.
- [Jones, 1998] Jones, C. *Estimating Software Costs*. McGraw-Hill, 1998.
- [Karner, 1993] Karner, G. *Use Case Points - Resource Estimation for Objectory Projects*. Objective Systems SF AB, University of Linköping, Suécia, 1993.
- [Kotonya, 1998] Kotonya, G. and Sommerville, I. *Requirements Engineering: Processes and Techniques*. John Willey & Sons Ltd, New York, USA, 1998.
- [Lucena, 2003] Lucena, Fontoura and Crespo. *Um Estudo Comparativo sobre Ambientes de Educação na Web*. Conteúdo do curso TIAE. <http://www.les.inf.puc-rio.br/aulanet>
- [Lutz, 1993] Lutz, R. *Analyzing Software Requirements Errors in Safety-Critical Embedded Systems*. Proceedings of the IEEE International Symposium on Requirements Engineering, IEEE Computer Society Press, San Diego, CA, 126–133, January 1993.
- [Kristen, 1996] Kristen, G. *Total Quality Management with Object Orientation*. Business Objects Object Magazine, March-April 1996.

- [Krutchen, 1998] Krutchen, P. *The Rational Unified Process: An Introduction*. Addison-Wesley, 1998.
- [McGarry, 2001] McGarry, J. Et All. *Practical Software Measurement: Objective Information for Decision Makers*. Addison Wesley, 2001.
- [NESMA] NESMA - Netherlands Software Metrics Association. Indicative Function Point Count: www.nesma.org.nl
- [Ross] Ross, M. *Size Does Matter: Continuous Size Estimating and Tracking*. Quantitative Software Management, Inc. www.qsm.com/qw99cse.pdf
- [SEI, 2002] SEI. *Capability Maturity Model Integration (CMMI SM)*. Version 1.1 - CMU/SEI-2002-TR-012; Março 2002.
- [UNISYS, 1995] UNISYS. *Análise de Pontos de Função*. Apostila - Curso de Análise de Pontos de Função da UNISYS, 1995.
- [Vazquez, 2004] Vazquez, C. AT All. *Análise de Pontos de Função: Medição, Estimativas e Gerenciamento de Projetos de Software*. Editora Érica Ltda, São Paulo, 2004, pp17-35.