

Desenvolvimento de Software Para Aplicação da Métrica de Estimativa Através da Análise de Pontos Por Casos de Uso

Matheus Ricalde de Souza¹, Gustavo Stangherlin Cantarelli¹

¹Centro Universitário Franciscano
Santa Maria – RS – Brasil

matheusricaldee@gmail.com, gus.cant@gmail.com

Abstract. *This paper presents a study for Use Case Points and from this, the development of a software in Java programming and the database manager, SQL Server database, which performs the analysis of Use Case Points by calculating the language complexity of actors and use cases automatically from the description of the use cases and the environmental and technical factors manually by filling done by the user.*

Resumo. *O presente trabalho traz um estudo sobre Pontos por Casos de Uso e a partir deste, o desenvolvimento de um software, na linguagem de programação Java e com o gerenciador de banco de dados SQL Server, que realiza a análise de Pontos por Casos de Uso calculando a complexidade dos atores e dos casos de uso automaticamente através do descritivo dos casos de uso e os fatores ambientais e técnicos através do preenchimento realizado pelo usuário.*

1. Introdução

A qualidade do desenvolvimento do produto de software sempre foi um grande desafio dentro das organizações de desenvolvimento do mesmo. A principal causa deste problema é a falta de um processo de desenvolvimento efetivo e claramente definido. Assim, a definição de um processo de software tornou-se um requisito essencial para a obtenção de produtos de software com qualidade, e para que seja possível a obtenção de uma melhora consistente, os processos necessitam de informações mais precisas sobre estimativas de tamanho de software, que podem ser realizadas através da análise de Pontos por Casos de Uso (PCU).

Os PCUs foram criados por Gustav Karner em 1993, como uma adaptação específica dos Pontos por Função. A análise de sistemas orientados a objetos já utiliza os diagramas de casos de uso para a descrição de funcionalidades do sistema em relação ao usuário, a análise de PCU foi criada para que seja possível estimar o tamanho do sistema nesta fase, utilizando documentos gerados nesta fase como alternativa para o cálculo dimensional. Uma vez que os casos de uso do sistema sejam levantados é possível estimar o tamanho do software baseado em um conjunto simples de métricas e modificadores.

Segundo Gielow et al. (2003), o modo como a métrica de PCU lança uma estimativa é o principal diferencial dela em relação a outras métricas de estimativas, já que o método estima o tamanho do software a partir do modo como os usuários o utilizarão, analisando a complexidade de ações requeridas por cada usuário e uma análise de alto nível dos passos necessários para a realização de cada tarefa.

A partir dos problemas encontrados em relação à credibilidade de estimativas realizadas por equipes de desenvolvimento de softwares, foi elaborado este estudo sobre métricas de estimativas de PCU propondo um sistema no qual é possível estimar o tamanho do software a partir do modelo de casos de uso.

O sistema desenvolvido no presente trabalho, realiza a estimativa da métrica de PCU através do cálculo do PCU não ajustados (calculado automaticamente através do descritivo dos casos de uso) X Complexidade dos Fatores Ambientais X Complexidade dos Fatores Técnicos, podendo o usuário ainda gerar relatórios simplificados, voltados para os clientes, e relatórios técnicos, voltados para a equipe de desenvolvimento.

Desta forma pode-se evidenciar como objetivos específicos, o desenvolvimento do sistema para realizar a análise da métrica de Pontos por Casos de Uso, atribuindo valores a complexidade dos atores e dos casos de uso através do descritivo do modelo de caso de uso e atribuir valores aos Fatores Ambientais e Técnicos através do preenchimento manual do usuário, realizar a geração de relatórios simplificados e complexos sobre PCU de cada projeto, definir uma metodologia para desenvolvimento desta proposta e validar a proposta através de estudos de caso específicos que contemplem o cenário proposto.

2. Trabalhos Correlatos

Todos os trabalhos apresentados a seguir trazem aplicações de Pontos por Casos de Uso. Dois deles tratam do desenvolvimento de um software baseado em Pontos por Casos de Uso e outro refere-se à aplicação de Pontos por Casos de Uso em empresas.

2.1. Comparing Effort Estimates Based On Use Case Points With Estimatives

O trabalho desenvolvido por Anda (2002) relata os resultados de um estudo realizado para avaliar o método que estima o esforço de desenvolvimento de software baseado em PCU, comparando-o com estimativas feitas por especialistas em desenvolvimento de software.

Os resultados demonstram que o método de Pontos por Casos de Uso podem ser usados com grande sucesso na estimativa de esforço de desenvolvimento de software. Além disso, a combinação de estimativas de peritos e estimativas baseadas em Pontos por Casos de Uso podem ser benéficas quando os avaliadores não têm experiência específica com o domínio da aplicação dos Pontos por Casos de Uso.

2.2. Effort Estimation Tool Based On Use Case Points

Seguindo na linha da aplicação de Pontos por Casos de Uso, Kusumoto (2004) desenvolveu um estudo visando o desenvolvimento de um software denominado U-EST, este implementado em Java e Xerces2 Java Parser.

Como primeiro passo, o usuário desenha o caso de uso e o salva no formato XMI. Então, o software XMI analisador extrai automaticamente os atores e casos de uso a partir do arquivo de entrada de modelo de casos de uso salvo. Então, o software analisa a complexidade deles e calcula os Pontos por Casos de Uso Não Ajustados. Feito isso, o software mostra a lista de atores e casos de uso com sua complexidade. E por fim são definidos os elementos técnicos e ambientais e a calculadora de PCU gera os resultados que são armazenados no banco de dados. A estimativa de esforço é calculada multiplicando o número de homens/hora pelo cálculo gerado pelos PCU.

2.3. Ferramenta de Suporte ao Cálculo dos *Use Case Points*

Gielow (2003) apresentou à Universidade Regional de Blumenau um estudo que tem como objetivo o desenvolvimento de uma ferramenta para cálculo de Pontos por Casos de Uso a partir da leitura de diagramas gerados pela ferramenta CASE Rational Rose. Para implementação do trabalho foi utilizado o ambiente de desenvolvimento Visual C++ 6.0.

A tela principal do software apresenta como opções abrir arquivo UML (*Unified Modeling Language*) do CASE Rational Rose, salvar o arquivo, abrir um arquivo salvo e imprimir um relatório referente ao cálculo. Depois de abrir um arquivo UML do CASE Rational Rose, o usuário define o tipo de ator de acordo com a complexidade do mesmo. Definido o grau de complexidade, o usuário pode preencher os fatores técnicos e ambientais, informando o grau de influência de cada um. E, por fim, o usuário insere a quantidade de homens/hora e seleciona a opção efetuar cálculo, o qual será realizado pelo software.

2.4. Considerações

Os três trabalhos apresentados visam auxiliar o uso de Pontos por Casos de Uso, tanto por meio de comparações como pelo desenvolvimento de um software para o auxílio do uso de Pontos por Casos de Uso. Assim como fez Kusumoto (2004), este trabalho utilizou a linguagem de programação Java, porém, com a diferença de que a aplicação não está voltada para a geração da complexidade dos atores e dos casos de uso a partir de um diagrama UML, mas sim para a geração dos mesmos através de um descritivo de caso de uso.

3. Referencial Teórico

Neste capítulo, serão apresentados conceitos essenciais para o entendimento deste trabalho. Serão abordados assuntos como Java e seu Kit de desenvolvimento, conceitos de Pontos de Casos de Uso, o sistema gerenciador de banco de dados SQL Server e a metodologia ICONIX.

3.1. Pontos Por Casos de Uso (PCU)

Karner (1993) baseou o modelo de Pontos de Casos de Uso caso no modelo de Pontos de Função. Começando com a medição da funcionalidade do sistema baseado no modelo de caso de uso em um número chamado Pontos por Caso de Uso Não Ajustados. Fatores técnicos envolvidos no desenvolvimento desta funcionalidade também são avaliados, semelhante aos Pontos de Função. O último passo na estimativa

não é, no entanto, a partir dos Pontos de Função e é um novo fator chamado Fator Ambiental proposto pelo autor. Os Pontos de Casos de Uso (PCU) são produto destes três fatores.

De acordo com Freire (2003), a técnica de Pontos de Casos de Uso foi desenvolvida para que seja possível estimar o tamanho de um sistema na fase de levantamento de requisitos, utilizando-se dos próprios documentos gerados nesta fase de análise como subsídio para o cálculo.

Segundo Cohn (2005), os Pontos por Casos de Uso em um projeto são calculados a partir da equação entre os itens listados a seguir:

- I. O total da complexidade dos casos de uso do sistema;
- II. O total da complexidade dos atores no sistema;
- III. Fatores técnicos (como a portabilidade, o desempenho, a facilidade de manutenção) que não são descritos como casos de uso;
- IV. O ambiente em que será desenvolvido o projeto (assim como a linguagem e a motivação da equipe).

A Figura 1 ilustra a atribuição de exemplo da complexidade dos atores e dos casos de uso, onde no lado esquerdo da figura são demonstrados o peso e a complexidade de cada um deles e no lado direito, a atribuição da quantidade e o valor dos mesmos, realizando assim o cálculo dos Pontos de Casos de Uso Não Ajustados através da soma do total dos valores dos atores com o total dos valores dos casos de uso.

Mensurando Complexidade dos Atores				
Ator	Interface	Peso	Qtd. Atores	Valor
Simple	Interface de programa (API)	1	3	3
Médio	Protocolo (Ex.:TCP/IP) ou interface em modo texto	2	2	4
Complexo	Interface gráfica	3	1	3
Total			6	10
Mensurando Complexidade dos Use Cases				
Caso de	Descrição	Peso	Qtd. Casos de Uso	Valor
Simple	< 3 transações ou < 5 classes de análise	5	3	15
Médio	4-7 transações ou 5 a 10 classes de análise	10	2	20
Complexo	> 7 transações ou > 10 classes de análise	15	1	15
Total			6	50
PCUNA			60	

PCUNA = Pontos de Casos de Uso Não Ajustados

Figura 1 – Exemplo da Complexidade dos Atores e dos Casos de Uso.

A Figura 2 demonstra a atribuição de exemplo dos Fatores Técnicos com o número de identificação do fator, a descrição e o peso na lado esquerdo da figura. O Fator Técnico é calculado através da soma da multiplicação do peso de cada fator vezes o valor atribuído (lado direito da figura) ao mesmo. A Complexidade dos Fatores Técnicos (CFT) é realizada através do cálculo: $CFT = 0,6 + (0,01 \times FT)$ (equação definida por Gustav Karner).

Considerando Fatores Técnicos do Projeto				
Fator	Descrição	Peso	Atribuído	Valor
T1	Sistema distribuído	2	2	4
T2	Objetivos de performance	1	3	3
T3	Eficiência on-line	1	4	4
T4	Complexidade de processamento	1	1	1
T5	Código reusável em outras aplicações	1	2	2
T6	Facilidade de instalação	0,5	1	0,5
T7	Facilidade de uso	0,5	3	1,5
T8	Portabilidade	2	4	8
T9	Facilidade de alterações (<i>changeability</i>)	1	2	2
T10	Concorrência	1	1	1
T11	Segurança	1	5	5
T12	Acesso direto a terceiros	1	2	2
T13	Treinamento para usuário	1	1	1
			FatorT	35
			FCT	0,95

FCT = Fator de Complexidade Técnica

Figura 2 – Exemplo da Complexidade dos Fatores Técnicos.

A Figura 3 demonstra a atribuição de exemplo dos Fatores Ambientais com o número de identificação do fator, a descrição e o peso (no lado esquerdo da figura). O Fator Ambiental é calculado através da soma da multiplicação do peso de cada fator vezes o valor atribuído ao mesmo (lado direito da figura). A Complexidade dos Fatores Ambientais (CFA) é realizada através do cálculo: $CFA = 1,4 + (-0,03 \times FA)$ (equação definida por Gustav Karner).

Considerando Fatores Ambientais				
Fator	Descrição	Peso	Atribuído	Valor
F1	Familiaridade da equipe com RUP	1,5	2	3
F2	Experiência da equipe	0,5	3	1,5
F3	Experiência da equipe em OO	1	1	1
F4	Capacidade dos analistas da equipe	0,5	1	0,5
F5	Motivação	1	2	2
F6	Estabilidade dos requisitos	2	3	6
F7	Estagiários ou funcionários em tempo parcial	-1	1	-1
F8	Domínio da tecnologia e configuração do ambiente	-1,5	0	0
			FatorA	13
			FA	1,01

FA = Fator Ambiental

Figura 3 – Exemplo da Complexidade dos Fatores Ambientais.

A Figura 4 ilustra o último passo para a realização da estimativa através de PCU, onde são demonstrados os PCU (baseado nos exemplos anteriores) e onde se deve atribuir as pessoas-hora por unidade de PCU (o criador da métrica sugere o uso de 20 pessoas-hora) e o tamanho da equipe. Sendo assim possível calcular as estimativas.

Pontos de Caso de Uso			
PCU	PCUNA *FCT*FA	57,57	
Pessoa-hora por unidade de PCU		20	pessoa-hora/PCU
Tamanho da equipe		20	pessoas
Estimativa em horas		57,57	horas
Estimativa em meses		0,36	meses

Figura 4 – Exemplo de PCU.

A Figura 4 ilustra o último passo para a realização da estimativa através de PCU, onde são demonstrados os PCU (baseado nos exemplos anteriores) e onde se deve atribuir as pessoas-hora por unidade de PCU (o criador da métrica sugere o uso de 20 pessoas-hora) e o tamanho da equipe. Sendo assim possível calcular as estimativas.

3.2 Banco de Dados SQL Server

Devido ao desempenho e integração com o Java serem de bom desempenho, o SQL Server foi o sistema gerenciador de banco de dados (SGBD) escolhido para todos os dados que serão armazenados no servidor do presente trabalho.

As consultas do SQL Server são feitas em SQL (*Structured Query Language*), uma linguagem padrão para bancos de dados relacionais em que é permitido adicionar, atualizar, recuperar e excluir informações em um banco de dados.

3.3 Linguagem de Programação Java

A linguagem de programação Java é considerada de alto nível, adquirida pela Oracle em 2010 e desenvolvida pela Sun Microsystems (ORACLE, 2013).

A compilação de um programa em Java gera um *bytecode*, e para executá-lo, é necessário que um interpretador leia o *bytecode* e repasse as instruções ao processador da máquina específica, denominado JVM (*Java Virtual Machine*). Não é necessário recompilar um programa para que ele rode em uma máquina diferente, basta que a plataforma tenha uma implementação que permita a emulação do JVM (DIAS e FONTES, 2003).

O Java é utilizado para desenvolver aplicativos de grande porte, aprimorar funcionalidades de Servidores Web, fornecer aplicativos para dispositivos móveis e muitos outros propósitos (DEITEL, 2010).

3.4 Java Development Kit (JDK)

Desde o ano de 1995 em que foi lançado, o JDK recebe atualizações e versões periódicas. Atualmente está na sua sétima versão, que traz consigo a Máquina Virtual Java, o compilador, API's do Java e outras ferramentas utilitárias.

O JDK tem como principais funções compilar, executar e depurar aplicações Java (DIAS e FONTES, 2003).

4. Metodologia

A metodologia de desenvolvimento a utilizada foi o ICONIX. Desenvolvida pela ICONIX software *Engineering*, a metodologia permitiu que a análise e representação dos problemas fossem feitas de forma adequada através de seus componentes.

O ICONIX é iterativo e incremental, situado entre o RUP (*Rational Unified Process*) e o XP (*Extreme Programming*), ou seja, entre a complexidade de um e a simplicidade do outro (SILVA E VIDEIRA, 2001).

O presente trabalho consiste no desenvolvimento de um sistema que realize o cálculo da métrica de Pontos por Casos de Uso. O sistema consiste em calcular a complexidade dos atores e dos casos de uso automaticamente a partir do descritivo dos casos de uso, este digitado pelo usuário, calculando assim os PCU Não-Ajustados. O sistema ainda permite que o usuário atribua valores aos Fatores Ambientais e aos Fatores Técnicos manualmente, e ainda especificar o tamanho da equipe para a realização da métrica. Ao final do cálculo da métrica de PCU o usuário poderá gerar relatórios técnicos e simplificados, sendo o técnico mais complexo para que a equipe de desenvolvimento entenda todo o cálculo e o simplificado voltado ao cliente.

O uso do sistema auxiliará os desenvolvedores de software a obter estimativas mais precisas sobre o desenvolvimento de produtos de software.

Para que os usuários possam manipular os dados no sistema foi desenvolvida uma interface utilizando a linguagem Java e o sistema gerenciador de banco de dados SQL Server.

Na Figura 5, ilustrada a seguir, o usuário irá realizar a ponderação dos fatores técnicos e ambientais manualmente e após digitar o descritivo dos casos de uso o sistema irá realizar a ponderação dos casos de uso e dos atores. O sistema também será responsável por gerar os relatórios técnicos e simplificados.

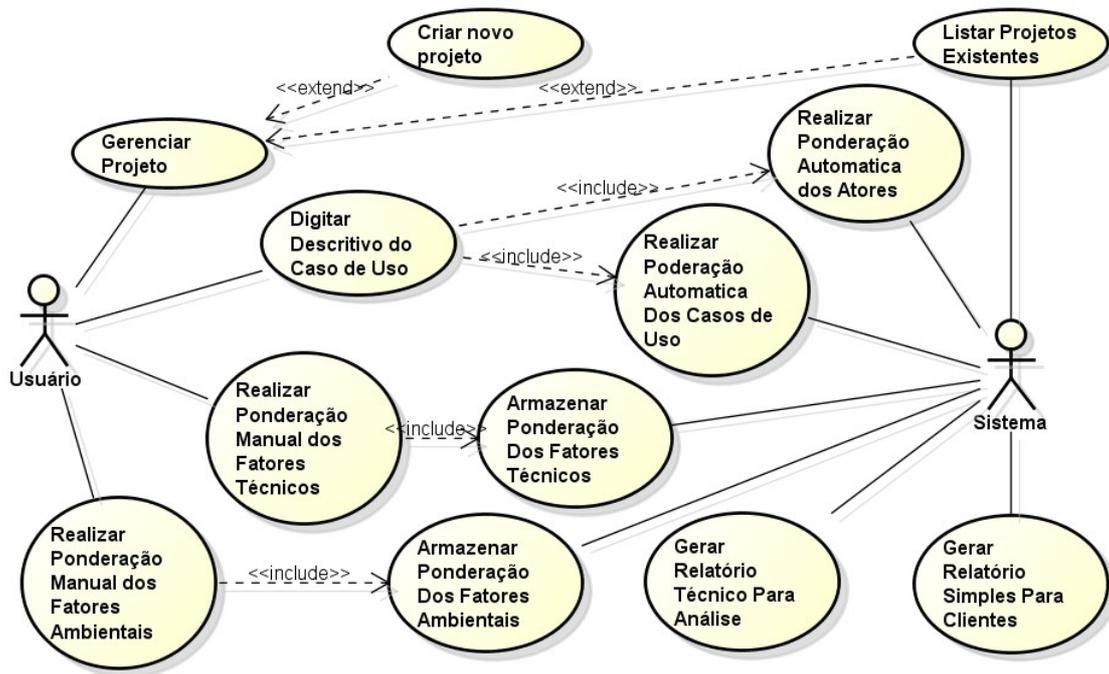


Figura 5 - Diagrama de Casos de Uso do sistema.

A Figura 6 exibe o diagrama entidade-relacionamento do banco de dados que é utilizado para a persistência dos dados.

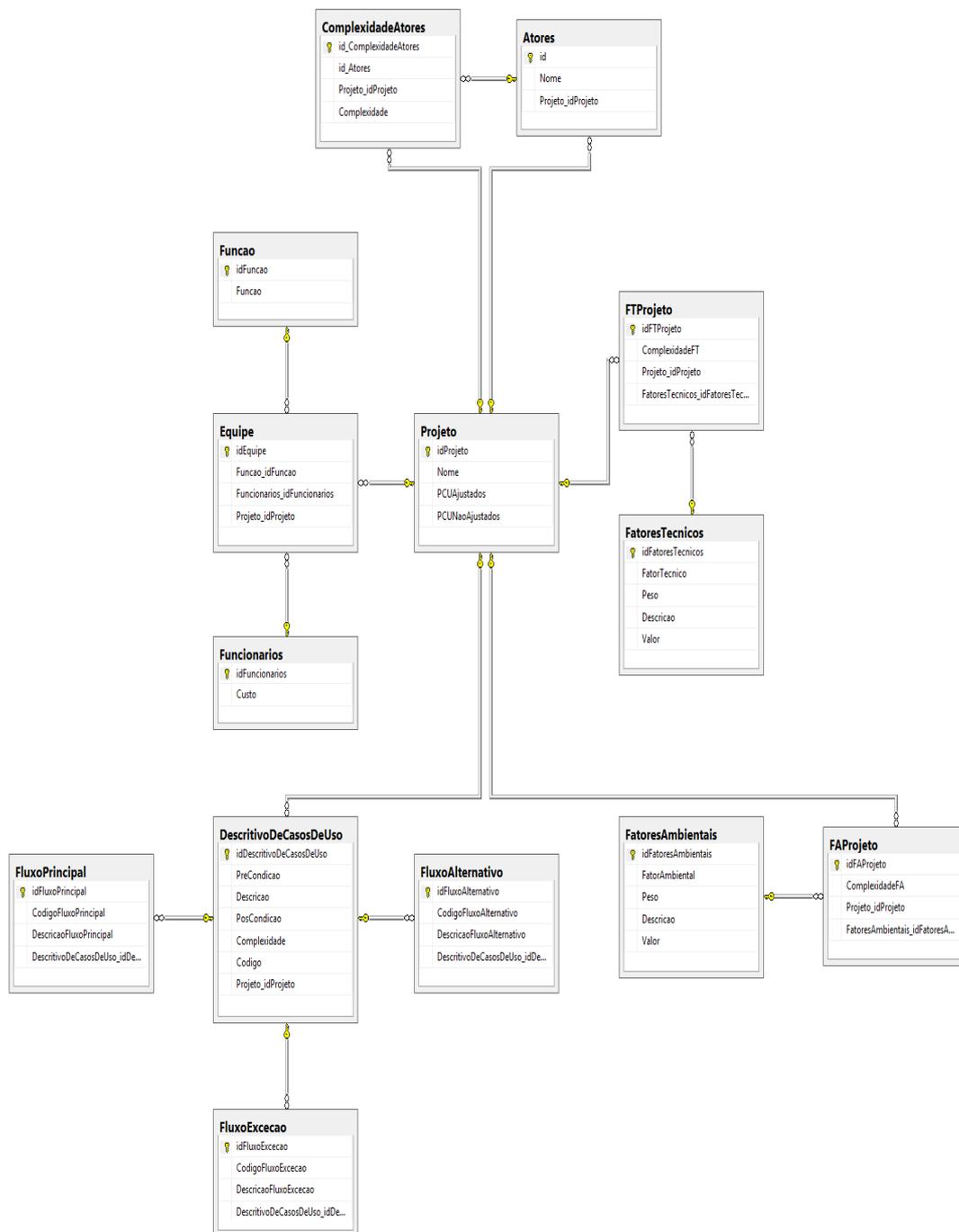


Figura 6 - Diagrama Entidade-Relacionamento do Banco de Dados

6. Resultados Obtidos

Um dos principais problemas encontrados na métrica de PCU é a falta de agilidade com que possa ser realizada a estimativa da mesma. O sistema desenvolvido busca uma maior agilidade na estimativa de custo de um projeto.

Diante da proposta foram desenvolvidas interfaces para a atribuição dos fatores técnicos, dos fatores ambientais e para o preenchimento dos descritivos de casos de uso.

A Figura 7 exibe a interface principal do sistema.

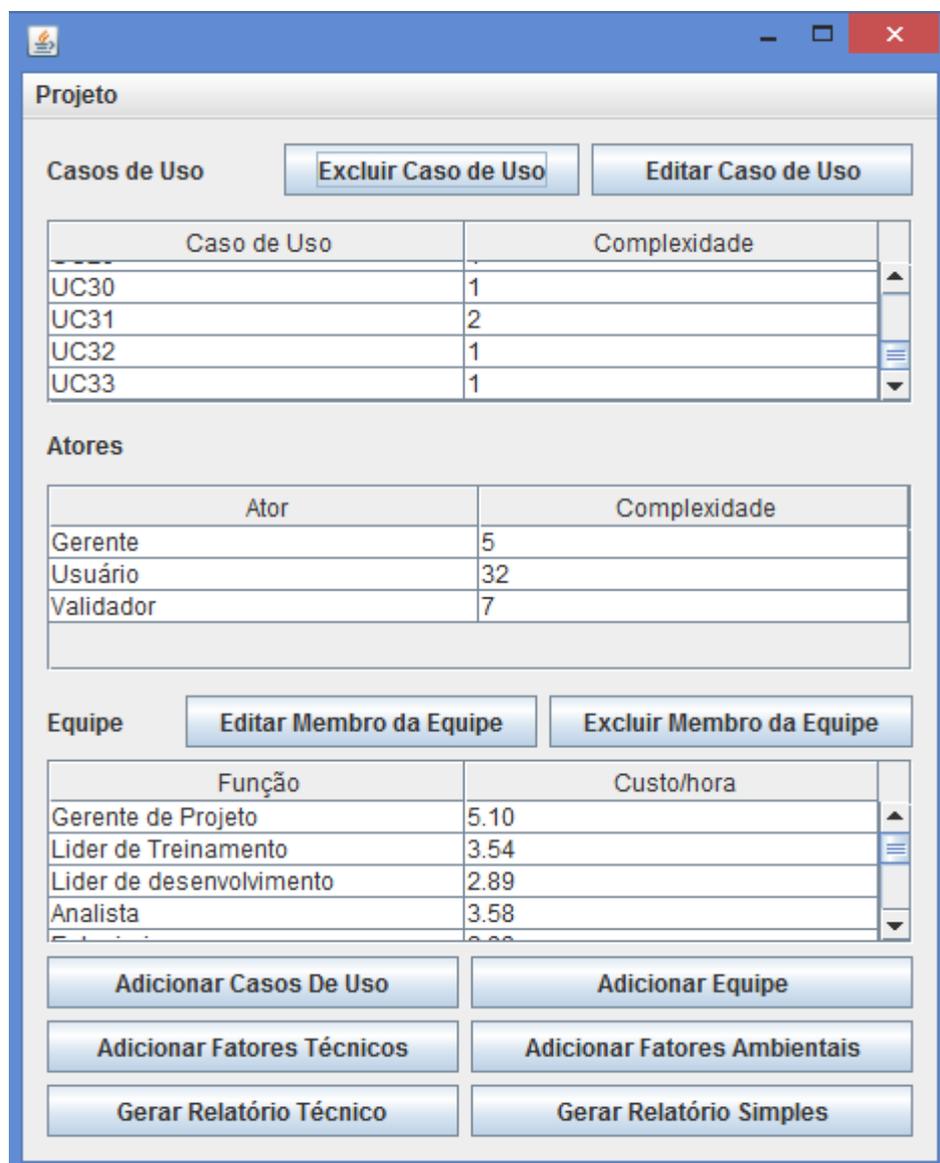


Figura 7 – Interface Principal do Sistema.

Conforme observa-se, no exemplo da Figura 7, o usuário após preencher os descritivos dos casos de uso do projeto, automaticamente o sistema calcula a complexidade dos casos de uso e dos atores do projeto. O usuário adicionou ao projeto os membros da equipe e seus respectivos custo/horas.

Após o usuário informar todos os dados necessários, pode-se então gerar relatórios técnicos e simplificados da análise de PCU do projeto.

A Figura 8 demonstra o relatório simplificado gerado a partir do exemplo da Figura 7.

Projeto Exemplo - Relatório Simples.

Pontos Por Casos De Uso Não Ajustados - PCUNA
Realizado Através da Soma das Complexidades dos Atores e dos Casos de Uso

170,00

Pontos Por Casos Ajustados - PCU
Realizado Através da Multiplicação do PCU vezes a Complexidade dos Fatores
Técnicos vezes a Complexidade dos Fatores Ambientais

175,77

Estimativa em Dias
Usando 8 Horas de Trabalho por Dia

36,62

Estimativa de Custo
Custo Total do Projeto

R\$ 9219,22

Figura 8 – Relatório Simplificado gerado pelo sistema.

Nas Figuras 9,10 e 11 é demonstrado o relatório técnico gerado pelo sistema a partir do projeto criado como exemplo, este voltado a equipe de desenvolvimento e mais completo que o relatório simplificado, demonstrando detalhadamente como foi possível obter os PCUs.

Projeto Exemplo - Relatório Técnico.

Lista dos Fatores Técnicos com Suas Respectivas Complexidades

Descrição	Complexidade
T1 - Sistema Distribuído	2.0
T2 - Tempo de Resposta	1.0
T3 - Eficiência	3.0
T4 - Processamento Complexo	3.0
T5 - Código Reusável	3.0
T6 - Facilidade de Instalação	1.5
T7 - Facilidade de Uso	1.0
T8 - Portabilidade	4.0
T9 - Facilidade de Mudança	2.0
T10 - Concorrência	3.0
T11 - Recursos de Segurança	2.0
T12 - Acessível por Terceiros	2.0
T13 - Requer Treinamento	4.0

Lista dos Fatores Técnicos com Suas Respectivas Complexidades

Descrição	Complexidade
E1 - Familiaridade com o Processo	3.0
E2	1.0
E3 - Experiência em Orientação a Objetos	3.0
E4 - Presença de Analista Experiente	1.0
E5 - Motivação	2.0
E6 - Requisitos Estáveis	4.0
E7 - Desenvolvedores em Meio Expediente	-2.0
E8 - Linguagem de Programação Difícil	-3.0

Lista dos Casos de Uso Com Suas Respectivas Transações

- Simple - Com Menos de 3 Transações
- Médio - Entre 4 e 7 Transações
- Complexo - Com Mais de 7 Transações

Descrição	Numero de Transações
UC01	1

UC02	6
UC03	3
UC04	3
UC05	1
UC06	1
UC07	2
UC08	1
UC09	1
UC10	1
UC11	1
UC12	1
UC13	1
UC14	1
UC15	1
UC16	1
UC17	1
UC18	1
UC19	1
UC20	1
UC21	1
UC22	1
UC23	1
UC24	1
UC25	1
UC26	1
UC27	1
UC28	1
UC29	1
UC30	1
UC31	2
UC32	1
UC33	1

Lista dos Atores Com Suas Respectivas Transações

- Simple - Com Menos de 3 Transações
- Médio - Entre 4 e 7 Transações
- Complexo - Com Mais de 7 Transações

Ator	Numero de Transações
Gerente	5
Usuário	32
Validador	7

Figura 9– Páginas 1 e 2 do Relatório Técnico gerado pelo sistema.

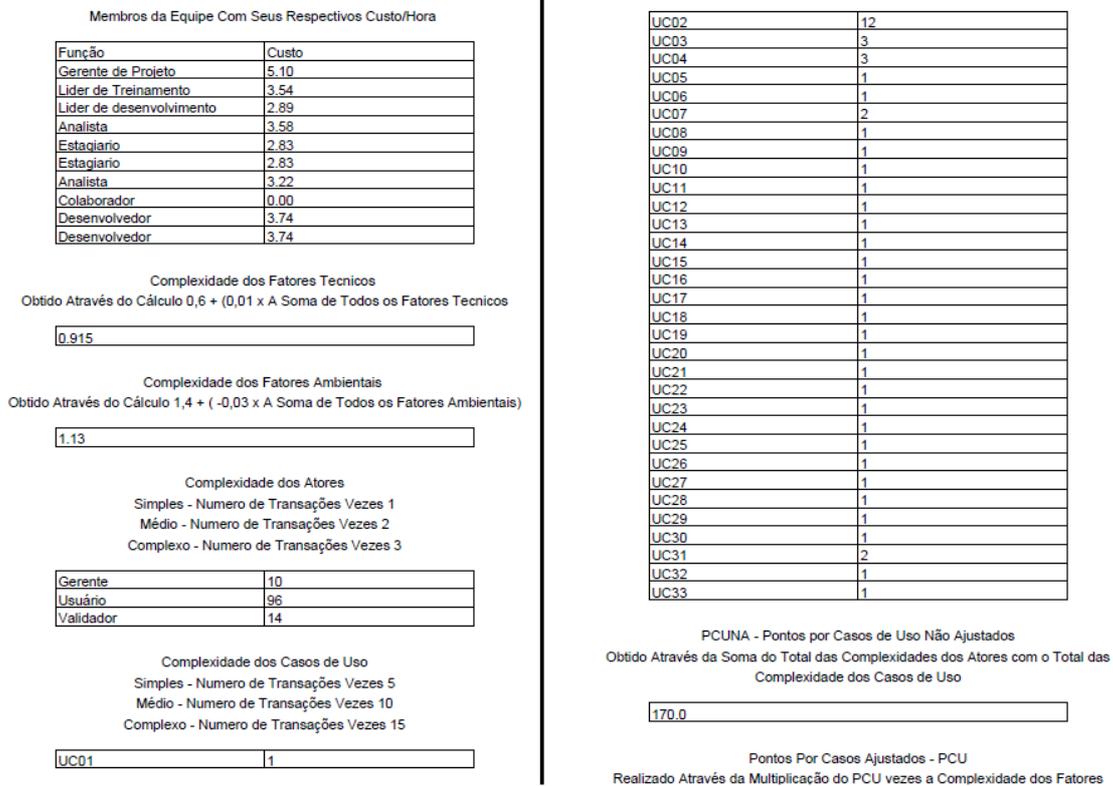


Figura 10– Páginas 3 e 4 do Relatório Técnico gerado pelo sistema.

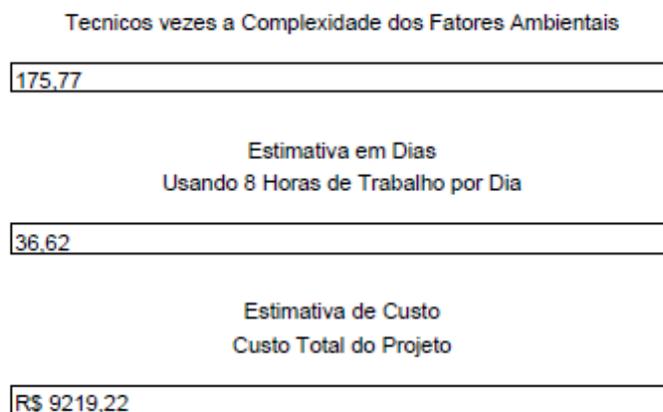


Figura 11– Pagina 5 do Relatório Técnico gerado pelo sistema.

7. Conclusões

Para a obtenção de uma melhora considerável os processos devem conter informações mais precisas sobre estimativas de tamanho de software, esta podendo ser realizada através de PCU. O sistema desenvolvido neste trabalho contribui na obtenção de

estimativas sobre o tamanho de software através da análise da métrica de PCU, gerando relatórios simplificado e técnicos, em virtude da alta precisão da métrica e também pela falta de sistemas no mercado para a realização da mesma.

O estudo e o desenvolvimento do sistema foram de grande valia para o conhecimento da comunicação entre linguagem de programação Java e o banco de dados SQL Server e ainda permitiu uma ampliação do conhecimento da métrica escolhida, bem como a realização de seu cálculo.

Os relatórios gerados pelo sistema obtiveram as estimativas esperadas, permitindo concluir que a implantação do sistema é perfeitamente viável em empresas de desenvolvimento de software. Os testes realizados ainda permitiram o auxílio ao usuário, de maneira adequada, não somente na questão de agilidade, mas principalmente, na eliminação da etapa de treinamento de pessoas para a realização de estimativas através de PCU.

Sugere-se o aperfeiçoamento e uma melhor usabilidade do sistema. Na versão atual o usuário necessita realizar sua estimativa de custo em um computador, a sugestão é o desenvolvimento de um sistema para análise de PCU para dispositivos móveis, buscando uma melhor portabilidade do mesmo.

Referências

ANDA, B.(2002) “Comparing Effort Estimates Based On Use Case Points With Expert Estimates”. Empirical Assessment in software Engineering (EASE). Keele, Toronto.

COHN, Mike.(2005) “Estimating With Use Case Points”, 3^a Ed. Estados Unidos: Addison-Wesley.

DEITEL, Paul J.; DEITEL, Harvey M. (2010) “Java – Como Programar”. 8^a Ed. Brasil: Prentice Hall.

FREIRE, Herval. (2013) “Calculando estimativas: o método de pontos de caso de uso”, <http://www.cnnt.com.br/files/usecasepoints.pdf>, Agosto.

GIELOW, Sandra Carla. (2003), “Ferramenta de Suporte ao Cálculo dos *Use Case Points*”, Blumenau: Universidade Regional de Blumenau.

INDRUSIAK, Leandro Soares. (1996), ” Linguagem Java”, Porto Alegre: Universidade Federal do Rio Grande do Sul (UFRGS).

KARNER, Gustav. (1993), “Metrics for objectory”, Tese – Linkoping University Department of Computer and Information Science. Suécia..

KUSUMOTO, S. (2013), “Effort Estimation Toll Based On Use Case Points Method”, http://pdf.aminer.org/000/492/872/estimating_effort_by_use_case_points_method_tool_and_case.pdf, Agosto.

ORACLE. Sun Microsystems.(2013), <http://www.oracle.com/us/sun/index.html>, Agosto.

SILVA, Alberto; VIDEIRA, Carlos. (2001), “UML, Metodologias e Ferramentas CASE”. Lisboa: Centro Atlântico.

SQL Server. (2013), “SQL Server”, www.microsoft.com/SQLServer, Agosto.