

**UNIVERSIDADE DE CAXIAS DO SUL**  
**Centro de Computação e Tecnologia da Informação**  
**Curso de Bacharelado em Sistemas de Informação**

**Deise Scariott**

**MÉTRICAS DE SOFTWARE PARA PROCESSOS DE  
DESENVOLVIMENTO ORIENTADOS A CASOS DE USO**

**Caxias do Sul**

**2011**

**Deise Scariott**

**MÉTRICAS DE SOFTWARE PARA PROCESSOS DE  
DESENVOLVIMENTO ORIENTADOS A CASOS DE USO**

Trabalho de Conclusão de Curso para  
obtenção do Grau de Bacharel em  
Sistemas de Informação da Universidade  
de Caxias do Sul.

**Iraci Cristina da Silveira De Carli**  
**Orientadora**

**Caxias do Sul**  
**2011**

Dedico este trabalho ao Isma meu “namorado”  
que esteve sempre ao meu lado me dando forças e  
me incentivando a continuar.

## **AGRADECIMENTOS**

Agradeço a minha família por terem entendido a minha ausência nestes meses de dedicação ao TCC, especialmente ao meu “namorado” por ter aguentado tantas vezes as minhas lamentações.

Agradeço a orientadora pela paciência e disponibilidade ao longo deste semestre e aos meus colegas pelo conhecimento e informações compartilhados.

Em especial agradeço a Deus por sempre iluminar os meus caminhos.

## RESUMO

O presente trabalho apresenta o desenvolvimento de uma ferramenta para calcular a estimativa de tamanho de software. Esta será feita através do uso de uma métrica aderente a um processo de desenvolvimento orientado a casos de uso, objetivando auxiliar na obtenção de estimativas eficazes.

Para tanto serão abordadas e comparadas as métricas de *Function Point Analysis* (FPA), *Use Case Points* (UCP) e *Constructive Cost Model* (COCOMO), estabelecendo suas vantagens e desvantagens com base no processo de desenvolvimento, assim definindo a melhor prática a ser aplicada. A métrica escolhida juntamente com o processo de desenvolvimento mapeado, serão os requisitos para a construção ferramenta de estimativa.

**Palavras Chaves:** Métricas, Estimativa de Software, *Function Point Analysis*, *Use Case Points*, *Constructive Cost Model*

## ABSTRACT

This work presents the development of a tool for calculating and estimating software's size. This will be made through the use of an adherent metric to a development process orientated to use cases, purposing support to obtain efficient estimations.

For this achievement will be take and compared the metrics of Function Point Analisis (FPA), Use Case Points (UCP) and Construtive Cost Model (COCOMO), definingtheadvantagesanddisavantagesbasedonthedevelopmentprocessand in this way, defining the best practice to be applied. The chosen metric united to the mapped process will be the requirements to build estimation tool software.

**Key words:** Metrics, Software Estimation, Function Point Analysis, Use Case Points, Constructive Cost Model

## LISTA DE FIGURAS

Figura 1 - Organizações de acordo com as métricas utilizadas .....	18
Figura 2 - FPA: Passo necessários para a contagem.....	20
Figura 3 - Processo de desenvolvimento .....	46
Figura 4 - Sub-processo entender escopo .....	48
Figura 5 - Sub-processo realizar análise detalhada.....	48
Figura 6 - Construção do software e help .....	49
Figura 7 - Validar a implementação.....	50
Figura 8 - Processo de desenvolvimento adaptado .....	56
Figura 9 - Adaptação: Sub-processo entender o escopo .....	57
Figura 10 - Adaptação: Sub-processo elaborar análise detalhada .....	57
Figura 11 - Diagrama de casos de uso .....	62
Figura 12 - Caso de uso estimar/rever esforço do projeto expandido.....	63
Figura 13 - Cadastro de fatores de ajuste.....	77
Figura 14 - Cadastro de fases.....	78
Figura 15 - Cadastro de Projetos.....	79
Figura 16 - Configurar projeto: Complexidade UC.....	80
Figura 17 - Configurar projeto: Complexidade atores .....	80
Figura 18 - Configurar projeto: Fatores de ajuste .....	81
Figura 19 - Cadastro da estimativa do projeto .....	82
Figura 20 - Complexidade dos casos de uso .....	82
Figura 21 - Complexidade dos atores .....	83
Figura 22 - Casos de uso não ajustados .....	84
Figura 23 - Casos de uso ajustados .....	86
Figura 24 - Comparativo de tempo das estimativas.....	87

## LISTA DE TABELAS

Tabela 1–FPA: Complexidade funcional dos ALI e AIE.....	21
Tabela 2–FPA: Contribuição das funções do tipo de dados.....	22
Tabela 3–FPA: Objetivos de cada tipo de transação.....	22
Tabela 4– FPA: Complexidade EE, SE e CE.....	24
Tabela 5–FPA: Contribuição das funções do tipo de transação.....	24
Tabela 6–FPA: Pontos de função não ajustados.....	25
Tabela 7–UCP: Tipos de ator x peso.....	28
Tabela 8–UCP: Tipos de casos de uso x peso por transações.....	29
Tabela 9–UCP: Tipos de casos de uso x peso por classes (entidades).....	30
Tabela 10–UCP: Fatores de complexidade técnica.....	31
Tabela 11 – UCP: Fatores de complexidade ambiental.....	32
Tabela 12–COCOMO: Taxas de conversão de UFP para SLOC.....	34
Tabela 13–COCOMO: Direcionadores de custos.....	38
Tabela 14–COCOMO: Fatores de escala.....	39
Tabela 15– COCOMO: Combinação dos direcionadores de custo.....	40
Tabela 16 – COCOMO: Sistema de classificação de pontos por objeto.....	42
Tabela 17–COCOMO: Taxas de produtividade para pontos por objeto.....	42
Tabela 18 - Comparativo métricas.....	53
Tabela 19 - Cálculo dos casos de uso não ajustados.....	84
Tabela 20 - Cálculo dos casos de uso ajustados.....	85

## LISTA DE EQUAÇÕES

Equação 1 – FPA: Fator de ajuste .....	26
Equação 2 – FPA: Projeto de desenvolvimento.....	26
Equação 3 – FPA: Projeto de melhoria.....	27
Equação 4 – FPA: Aplicação .....	27
Equação 5 – UCP: Valor dos pontos por caso de uso não ajustados .....	30
Equação 6 – UCP: Fatores de complexidade técnica.....	31
Equação 7 – UCP: Fator de complexidade ambiental.....	31
Equação 8 – UCP: Valor dos pontos por casos de uso .....	32
Equação 9 – UCP: Estimativa de horas de programação.....	32
Equação 10 – COCOMO: Expoente referente aos fatores de escala .....	39
Equação 11 – COCOMO: Esforço modelo <i>Post-Architecture</i> .....	39
Equação 12 - COCOMO: Esforço modelo <i>Early Design</i> .....	41
Equação 13 - COCOMO: Novos pontos por objeto .....	42
Equação 14 - COCOMO: Esforço modelo <i>Application Composition</i> .....	42
Equação 15 - COCOMO: Tempo de Desenvolvimento .....	43

## LISTA DE ABREVIATURAS E SIGLAS

<b>Sigla</b>	<b>Significado em português</b>	<b>Significado em inglês</b>
ACAP	Capacidade do analista	Analyst Capability
AIE	Arquivos de Interface Externa	
ALI	Arquivos Lógicos Internos	
APEX	Experiência na aplicação	Applications Experience
AR	Arquivos Referenciados	
BPMN	Padrão de modelagem de processos de negócio.	Business Process Modeling Notation.
CE	Consultas Externas	
COCOMO		Constuctive Cost Model
CPLX	Complexidade do produto	ProductComplexity
CPM	Manual de práticas de contagem	Counting Practices Manual
DATA	Tamanho da base de dados	Data Base Size
DOCU	Documentação associada às necessidades do ciclo de vida	Documentation Match to Life-Cycle Needs
ECF	Fatores Ambientais	Environmental Complexity Factor
EE	Entradas Externas	
ER	Entidade Relacionamento.	Entity Relationship.
FA	Fator de Ajuste	
FLEX	Flexibilidade de desenvolvimento	Development Flexibility
FPA	Análise por pontos de função	Function Point Analysis
FSM	Medição Funcional de Tamanho	Functional Size Measurement
GI	Influência	
IEC	Comissão Eletrotécnica Internacional	International Electrotechnical Commission
IFPUG	Grupo internacional dos usuários de pontos de função	International Function Point User Group
ISO	Organização Internacional para Padronização	International Organization for Standardization
KSLOC	Milhares de linhas de código	K Source Linesof Code
LTEX	Experiência com as linguagens e ferramentas de desenvolvimento	Languageand Tool Experience
NESMA	Associação de métricas da Holanda	Netherlands Software Metrics Association
NOP	Novos pontos por objeto	
OO	Orientação a Objetos	
PCAP	Capacidade dos programadores	Programmer Capability

PCON	Rotatividade de pessoal	Personnel Continuity
PLEX	Experiência na plataforma de desenvolvimento	Platform Experience
PMAT	Maturidade do processo	Process Maturity
PMI	Instituto de Gerenciamento de Projetos	Project Management Institute
PREC	Precedência	Precedentedness
PROD	Produtividade	
PVOL	Volatilidade da plataforma	Platform Volatility
RELY	Confiabilidade requerida do software	Required Software Reliability
RESL	Arquitetura e resolução de risco	Architecture/Risk Resolution
RUP	Processo Unificado da Rational	Rational Unified Process
RUSE	Reusabilidade requerida	Developed for Reusability
SCED	Cronograma de desenvolvimento requerido	Required Development Schedule
SE	Saídas Externas	
SITE	Desenvolvimento distribuído	Multisite Development
SLOC	Linhas de fonte do código	Source Lines of Code
STOR	Restrição de armazenamento principal	Main Storage Constraint
TCF	Fatores de Complexidade Técnica	Technical Complexity Factor
TD	Tipos de Dados	
TEAM	Coesão da equipe	Team Cohesion
TIME	Restrição de tempo de execução	Execution Time Constraint
TOOL	Utilização de Ferramentas de Software	Use of Software Tools
TR	Tipos de Registros	
UAW	Peso do ator não ajustados	Unadjusted Actor Weight
UCP	Pontos por Casos de Uso	Use Case Point
UFP	Pontos de função não ajustados	Unadjusted Function Points
UML	Linguagem de modelagem unificada.	Unified Modeling Language.
USC	Universidade do Sul da Califórnia	University of Southern California
UUCP	Pontos por Casos de Uso não Ajustados	Unadjusted Use Case Point
UUCW	Peso do caso de uso não ajustado	Unadjusted Use Case Weight

## SUMÁRIO

<b>1. INTRODUÇÃO .....</b>	<b>14</b>
1.1. Objetivos .....	16
1.2. Objetivos Específicos.....	16
1.3. Estrutura do Texto.....	16
<b>2. ESTUDO DE MÉTRICAS.....</b>	<b>18</b>
2.1. Análise de Pontos de Função .....	18
2.1.1. Determinar o Tipo de Contagem.....	20
2.1.2. Definir a Fronteira da Aplicação e Escopo da Contagem .....	20
2.1.3. Contar as Funções do Tipo de Dados.....	21
2.1.4. Contar Funções do Tipo de Transação.....	22
2.1.5. Determinar a Contagem de Pontos de Função não Ajustados .....	24
2.1.6. Determinar o valor do fator de ajuste.....	25
2.1.7. Determinar o Número de Pontos de Função Ajustados .....	26
2.2. Pontos por Casos de Uso.....	27
2.2.1. Contagem dos Atores .....	28
2.2.2. Contagem dos Casos de Uso .....	29
2.2.3. Equação para Obter o Valor dos Pontos por Caso de Uso não Ajustados .....	30
2.2.4. Cálculo dos Fatores de Ajuste.....	30
2.2.5. Equação para Obter o Valor dos Pontos por Casos de Uso Ajustados .....	32
2.3. COCOMO.....	32
2.3.1. Modelo Post-Architecture .....	33
2.3.2. Modelo Early Design .....	40
2.3.3. Modelo Application Composition.....	41
2.3.4. Equação do Tempo de Desenvolvimento.....	43
2.3.5. Equipe Média e Custo do Projeto.....	43
2.4. Considerações sobre as métricas.....	43
<b>3. MAPEAMENTO DO PROCESSO DE DESENVOLVIMENTO.....</b>	<b>45</b>
3.1. Processo de Desenvolvimento Atual.....	45
3.1.1. Qualificação e Estimativa.....	47
3.1.2. Análise e arquitetura.....	48
3.1.3. Construção e Testes.....	49
3.1.4. Liberação .....	50

3.1.5. Considerações Finais .....	50
<b>4. USO DE UMA MÉTRICA NO PROCESSO DE DESENVOLVIMENTO .....</b>	<b>52</b>
4.1. Métrica a ser Aplicada.....	52
4.2. Adaptações no Processo de Desenvolvimento .....	55
<b>5. FERRAMENTA DE ESTIMATIVA.....</b>	<b>60</b>
5.1. Fase de Análise.....	60
5.1.1. Diagrama de Casos de Uso .....	61
5.1.2. Descrição dos Casos de Uso .....	63
5.1.3. Protótipos de Interface .....	70
5.1.4. Diagrama de Classes Conceitual.....	70
5.1.5. Diagrama de Atividades.....	71
5.2. Fase de Projeto.....	71
5.2.1. Diagramas de Interação .....	71
5.2.2. Diagrama de classes de projeto.....	72
5.3. Implementação.....	72
5.4. Testes.....	73
5.5. Requisitos para Instalação .....	74
<b>6. SIMULAÇÃO DE UTILIZAÇÃO DA FERRAMENTA .....</b>	<b>75</b>
6.1. Características do Projeto .....	75
6.2. Simulação de Estimativa do Projeto Utilizando a Ferramenta .....	76
6.3. Comparação das Estimativas .....	86
<b>7. CONCLUSÃO .....</b>	<b>88</b>
<b>8. REFERÊNCIAS .....</b>	<b>90</b>
<b>9. ANEXOS .....</b>	<b>92</b>

## 1. INTRODUÇÃO

Ao pensar em desenvolvimento de software lembra-se de imediato de fatores como linguagens de programação, tecnologia, requisitos, funcionalidades, processos e qualidade. Todos estes fatores citados são importantes e devem ser levados em consideração na construção de um sistema.

Além das variáveis citadas acima, são identificados outros pontos que dizem respeito ao desenvolvimento de sistemas em ambientes corporativos, onde são contemplados não um, mas sim diversos softwares ou funcionalidades ao mesmo tempo. Então entram em evidência aspectos como a escassez de recursos, diferentes prioridades, prazos de entrega, todos pontos importantes para o desenvolvimento de produtos.

A definição dos processos de desenvolvimento bem estruturados e aderentes a necessidades da empresa é importante, bem como a medição da eficiência e eficácia destes processos. Um aspecto crítico dessas medições é o tamanho do software, o qual tem influencia na gestão dos projetos em pontos essenciais como a distribuição de recursos e as estimativas de custos. (ISO/IEC 14143-1:2007).

As medidas de tamanho de software surgiram com o objetivo de estimar o esforço (número de pessoas/hora) e o prazo, associados ao desenvolvimento dos programas e sistemas, facilitando assim o trabalho de acomodar as diversas necessidades que precisam ser atendidas, dentro da capacidade de realização da empresa (ANSELMO, 2010).

Muitos métodos para realizar esta medição já foram propostos, o primeiro utilizado foi a contagem SLOC (*Source Lines of Code*) na qual era realizada uma medida física do projeto, mensurado através da quantidade de linhas de código (PRESSMAN, 2006).

Existiram ainda outras medidas baseadas nas características técnicas do *software*, porém, tais medidas sofriam limitações por não serem baseadas nas regras de negócio, ou requisitos do usuário. Além de serem de difícil entendimento, não podiam ser realizadas no início do processo de desenvolvimento.

Baseada no conceito de Medição Funcional de Tamanho (*Functional Size Measurement* - FSM) surge a Análise por Pontos de Função (*Function Point Analysis* – FPA) publicada por Albrecht (1979). Diferente das métricas anteriores visa tratar fatores externos do software, ou seja, características importantes para o usuário, podendo ser aplicada no início do processo de desenvolvimento (ISO/IEC 20926:2009).

Em 1981 Barry Boehm introduz uma hierarquia de modelos de estimativa denominada

COCOMO (*Constructive Cost Model*), sendo este um modelo paramétrico baseado em dados históricos, utilizando equações matemáticas para medir esforço, prazo e tamanho da equipe. Este modelo tornou-se um dos modelos mais utilizados evoluindo com isso sua abrangência através do COCOMO II (PRESSMAN, 2006).

Ainda baseado neste contexto de medição funcional de software, Karner (1993) propõe uma metodologia baseada na FPA, porém, aplicada a softwares desenvolvidos no modelo orientado a objetos com modelagem UML. A contagem por Pontos por Casos de Uso (*Use Case Point - UCP*) toma por base o modelo de especificar sistemas da UML (*Unified Modeling Language*), utilizando casos de uso para realizar estimativa dos projetos já na fase de levantamento de Casos de Uso (ANSELMO, 2010).

Esta métrica utiliza a produtividade média da organização para produzir um UCP, aliado as medições feitas com base na modelagem de casos de uso. Sua aplicação é dada através da contagem de atores e casos de uso, identificação de sua complexidade, determinação da complexidade de fatores técnicos e ambientais e cálculos de UCP ajustados.

No desenvolvimento de software a utilização de uma metodologia para a estimativa aderente aos processos da organização é uma decisão importante. Deste modo, é necessário analisar quais os pontos positivos e negativos de cada metodologia e confrontar as relevâncias destes com no ambiente em que a métrica será aplicada.

Torna-se difícil justificar o tamanho dos projetos de desenvolvimento se não houver um processo único bem definido, ou seja, se estes projetos não tiverem sido estimados utilizando a mesma medida. Utilizando uma só métrica se torna possível passar da informação abstrata de que um projeto é “pequeno” ou “grande”, para uma dimensão de tamanho mensurável e justificável.

Outro aspecto importante é a dificuldade de evolução dos parâmetros utilizados para a medição de software, se a cada projeto for utilizada uma metodologia distinta para tal. Deste modo não será possível a utilização de dados históricos, como forma de melhoria em projetos futuros, pois as alternativas utilizadas em um projeto não serão efetivamente eficazes em outros, tendo em vista que não há um método estabelecido para a o processo de estimativa (VAZQUEZ, 2008).

Um desafio para a construção de software de qualidade é encontrar uma solução para métricas de desenvolvimento aderente ao processo, que não gere trabalho desnecessário e ofereça formas de armazenar e extrair dados históricos, assim tornando-se cada vez mais assertiva e eficaz.

## **1.1. Objetivos**

O objetivo deste trabalho é propor uma abordagem para o uso de uma métrica de estimativa do tempo de projetos de software, que seja aderente ao processo de desenvolvimento orientado a casos de uso de uma empresa específica, a qual será o cenário alvo de estudo deste trabalho.

Para isso, será construído um protótipo de ferramenta objetivando sistematizar a definição de tempo dos projetos, facilitando o aprimoramento do processo através de informações armazenadas, de forma a promover aprendizado e especialização.

## **1.2. Objetivos Específicos**

Para atingir o objetivo geral serão seguidos os seguintes objetivos específicos:

- Mapear e revisar o processo de desenvolvimento de software, enfatizando o estudo na modelagem de casos de uso.
- Identificar as métricas mais utilizadas atualmente ressaltando as principais funcionalidades de cada uma e definir a melhor prática a ser aplicada no cenário alvo.
- Propor adaptações no processo de desenvolvimento estudado, objetivando a melhor utilização do processo de estimativa definido.
- Implementar um protótipo de sistema de estimativa, possibilitando a execução e armazenamento das medições dos projetos. Com isso melhorar os parâmetros do processo, através do conhecimento obtido ao decorrer da sua utilização.
- Simular um projeto utilizando a solução construída.

## **1.3. Estrutura do Texto**

Primeiramente será realizado um estudo detalhado sobre as métricas de software, afim de obter um comparativo das suas características para a definição da métrica que será aplicada no processo alvo.

Em seguida será apresentado o processo de desenvolvimento que é alvo deste trabalho, onde serão mapeados e descritos os processo que relacionam-se diretamente com a utilização de uma métrica de software.

Para possibilitar a utilização da métrica escolhida serão propostas adaptações para o processo de desenvolvimento, com base nas práticas existentes e nas características da métrica de estimativa definida, observando para causar o menor impacto possível na organização e nos padrões utilizados pela empresa.

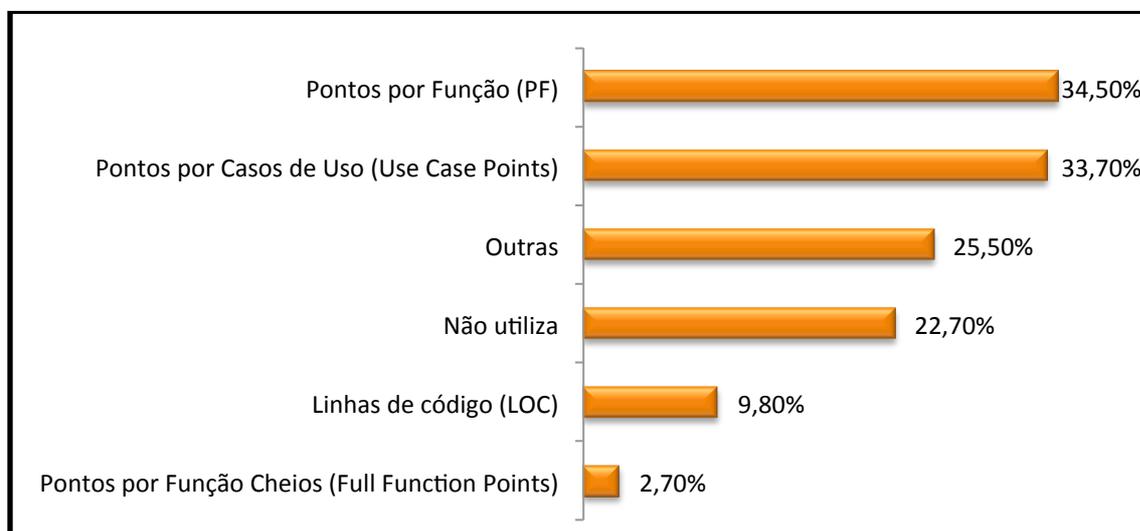
Com estas informações levantadas será realizada a modelagem dos requisitos para a implementação de um protótipo de ferramenta para a estimativa da métrica escolhida o ambiente de desenvolvimento estudado.

Em seguida será realizado o desenvolvimento do software propriamente dito, contemplando aplicação dos testes funcionais do protótipo e a simulação de um projeto utilizando o software desenvolvido.

## 2. ESTUDO DE MÉTRICAS

Segundo a Pesquisa de Qualidade no Setor de Software Brasileiro (2009), as métricas mais utilizadas pelas empresas de desenvolvimento de software no Brasil são Análise de Pontos de Função e Pontos por Casos de Uso, conforme pode ser observado na Figura 1— Organizações de acordo com as métricas utilizadas.

Esta pesquisa foi aplicada em trezentos e quarenta e três organizações sendo que destas duzentos e sessenta e quatro responderam, o equivalente a aproximadamente setenta e sete por cento. Vale ressaltar que a questão permitia múltiplas respostas.



**Figura 1 - Organizações de acordo com as métricas utilizadas**

**Fonte: Pesquisa de Qualidade no Setor de Software Brasileiro 2009.**

Com base no elevado índice de utilização de Pontos por Função e Pontos por Caso de Uso no cenário nacional, serão abordadas neste trabalho estas duas métricas e o modelo COCOMO II, o qual hoje apresenta a solução mais completa e vastamente utilizada na estimativa de software, considerando desde a estimativa de tamanho ao custo do projeto.

A partir deste estudo pretende-se observar as características de cada métrica e suas vantagens e desvantagens com relação ao processo de desenvolvimento alvo.

### 2.1. Análise de Pontos de Função

“A medição funcional é um termo geral para métodos de dimensionamento de

software baseados nas funções requeridas pelos usuários” Vazquez(2008, p. 40).

A análise de pontos de função (*Function Point Analysis – FPA*) surgiu na década de 70 com Allan Albrecht. Em 1986 com o crescimento contínuo de usuários foi fundado o IFPUG (*International Function Point User Group*), que publicou neste mesmo ano um manual propondo uma padronização da técnica de contagem.

Além do IFPUG existe a NESMA (*Netherlands Software Metrics Association*) a associação de métricas da Holanda, fundada em maio 1989. Esta por sua vez publicou um manual de contagem em 1990 baseada no manual do IFPUG.

Na da década de 90 já existiam diversos modelos de contagem com diferentes interpretações do método original proposto por Albrecht. Com o objetivo de garantir que todos estes funcionem de maneira similar foi desenvolvida a norma ISO/IEC 14143, que estabelece um conjunto de padrões para a medição de software.

Com relação a análise de pontos de função segundo Vazquez (2008) existem outros quatro métodos padronizados por ISO atualmente, são eles:

- IFPUG CPM<sup>1</sup>- ISO/IEC 20926;
- NESMA CPM - ISO/IEC 24570;
- Mark II CPM - ISO/IEC 20968;
- COSMIC-FFP *Measurement Manual*- ISO/IEC 19761.
- CPM (*Counting Practices Manual*)

O padrão desenvolvido pelo IFPUG normatizado pela ISO/IEC 20926 é o mais conhecido e utilizado. Contudo algumas propostas apresentadas pela NESMA são bastante utilizadas, como por exemplo, os três tipos de contagem por pontos de função em diferentes momentos de processo de desenvolvimento de software. São estes:

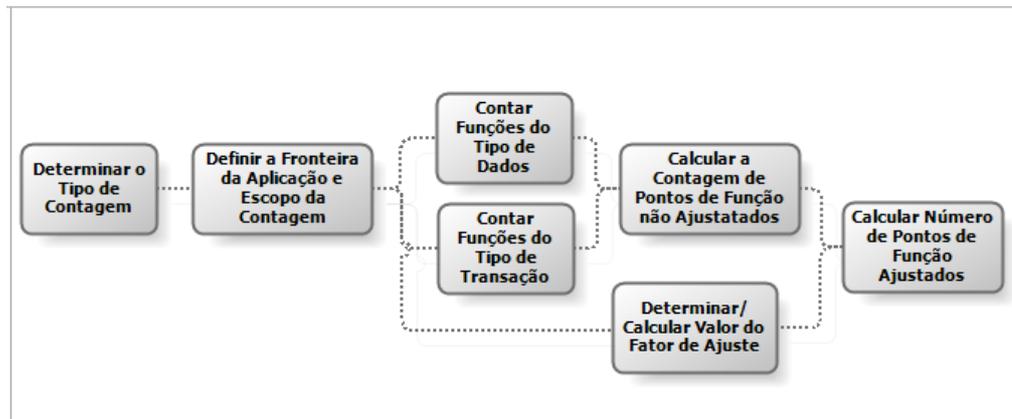
- Contagem indicativa no início do desenvolvimento;
- Revisão da contagem com base na complexidade das funções na fase de projeto;
- Contagem detalhada no término do desenvolvimento.

Utilizando esta linha observar-se que a FPA pode ser utilizada também para entender o aumento do escopo do projeto.

O Processo de contagem de pontos de função segundo Anselmo (2010), o qual se baseou na versão 4.3 do CPM publicado pelo IFPUG em janeiro de 2010, pode ser dividido nas etapas representadas na Figura 2 – FPA: Passos Necessários para a Contagem. Nos tópicos seguintes serão descritos os passos deste processo.

---

<sup>1</sup> CPM (*Counting Practices Manual*): Manual de praticas de contagem



**Figura 2 - FPA: Passo necessários para a contagem**

Fonte: Anselmo (2010, p. 29)

### 2.1.1. Determinar o Tipo de Contagem

A contagem de pontos de função pode ser de três tipos:

- Projeto de desenvolvimento: mede uma nova funcionalidade na sua primeira instalação e por isso, inclui quando necessário, funções de conversão de dados;
- Projeto de melhoria: mede funções incluídas, alteradas e removidas da aplicação pelo projeto, eventualmente também podem ser necessárias funções de conversão de dados;
- Uma Aplicação: mede as funcionalidade fornecidas pelo usuário na aplicação instalada.

A técnica utilizada na contagem é a mesma em cada caso, a diferença está no que é considerado em cada um.

### 2.1.2. Definir a Fronteira da Aplicação e Escopo da Contagem

O objetivo deste passo é delimitar a fronteira da aplicação a ser medida e com relação ao mundo exterior (aplicações externas ao domínio do projeto, usuários). Estabelecer esta demarcação é fundamental para o andamento do projeto, tendo em vista que esta será a base para as demais etapas.

O IFPUG estabelece as seguintes regras para a determinação da fronteira da aplicação:

- Dever ser feita com base no ponto de vista do usuário;

- Deve ser baseada na separação das funções conforme os processos de negócio;
- Em projetos de melhoria, deve estar de acordo com a fronteira do projeto ou da aplicação estabelecida anteriormente.

O escopo da contagem por sua vez, estabelece quais funcionalidades de dentro da fronteira da aplicação deverão ser consideradas, ou seja, as que serão afetadas pelo desenvolvimento ou melhoria.

### 2.1.3. Contar as Funções do Tipo de Dados

As funções do tipo de dados podem ser classificadas em Arquivos Lógicos Internos (ALI) e Arquivos de Interface Externa (AIE).

Os Arquivos Lógicos Internos (ALI) correspondem aos arquivos mantidos e utilizados por processos elementares da aplicação que está sendo contada. Já os Arquivos de Interface Externa (AIE) correspondem aos arquivos utilizados pela aplicação que está sendo contada, mas que são mantidos por outras aplicações.

Segundo Vazquez (2008, p. 73) ALI e AIE são:

- Grupos de dados ou informações de controle;
- Identificáveis pelo usuário;
- Logicamente relacionados.

Salientando que um ALI é mantido na fronteira da aplicação e um AIE é referenciado (lido) pela aplicação.

Cada ALI e AIE podem ser classificados, conforme o número de Tipos de Dados (TD) referenciados e o número de Tipos de Registros (TR). A complexidade atribuída poderá ser baixa, média ou alta de acordo com a Tabela 1 – FPA: Complexidade funcional dos ALI e AIE.

**Tabela 1–FPA: Complexidade funcional dos ALI e AIE**

	Tipos de Dados		
	< 20	20 – 50	> 50
Tipos de Registros			
1	Baixa	Baixa	Média
2 – 5	Baixa	Média	Alta
> 5	Média	Alta	Alta

Fonte: Vazquez (2008, p. 75)

Um TD pode ser identificado por ser um campo único, reconhecido pelo usuário e um TR por ser um subgrupo de dados, reconhecido pelo usuário, componente de um ALI ou AIE.

Após classificar a complexidade dos ALI e AIE deve-se calcular a sua contribuição utilizando a Tabela 2 – FPA: Contribuição das funções do tipo de dados.

**Tabela 2–FPA: Contribuição das funções do tipo de dados**

<b>Tipo de Função</b>	<b>Baixa</b>	<b>Média</b>	<b>Alta</b>
Arquivo Lógico Interno	7 PF	10 PF	15 PF
Arquivo de Interface Externa	5 PF	7 PF	10 PF

Fonte: Vazquez (2008, p. 78)

#### 2.1.4. Contar Funções do Tipo de Transação

As funções do tipo de transação representam a funcionalidade fornecida ao usuário para atender às suas necessidades de processamento de dados pela aplicação. São classificadas em Entradas Externas (EE), Saídas Externas (SE) e Consultas Externas (CE).

Vazquez (2008, p. 100)

Conforme Anselmo (2008) a Tabela 3 – FPA: Objetivos de cada tipo de transação, foi criada pelo IFPUG para auxiliar na identificação de cada tipo de função transacional e resume os seus objetivos primários.

**Tabela 3–FPA: Objetivos de cada tipo de transação**

<b>Tipo de Função Transação</b>	<b>EE</b>	<b>SE</b>	<b>CE</b>
Alterar o comportamento da aplicação	IP	F	NP
Manter um ou mais ALIs	IP	F	NP
Apresentar informações	F	IP	IP

Onde:

IP = Intenção primária da função de transação;

F = Uma operação que pode ser feita pela função, mas não é sua intenção primária. Ela pode existir ou não;

NP = Não é permitida a função transacional.

A complexidade das funções transacionais é determinada pelo número de arquivos de

dados (ALI e AIE) e pelos itens de dados referenciados. A seguir serão descritas resumidamente as características de cada uma das funções.

#### **a. Entradas Externas (EE)**

São as transações que tem como objetivo a manutenção de arquivos ou a alteração do comportamento da aplicação.

Segundo Vazquez (2008, p. 101), são exemplos de EE:

- Transações que recebem dados externos utilizados na manutenção de ALIs;
- Janela que permite adicionar, excluir e alterar registros em arquivos contribui com três entradas externas;
- Processamento em lotes de atualização de bases cadastrais a partir de arquivos de movimento.

#### **b. Saídas Externas (SE)**

São as transações que tem como objetivo apresentar informações aos usuários, através do processamento de uma fórmula matemática ou um cálculo, criando dados derivados e/ou mantendo ALI e/ou alterando o comportamento do sistema.

Segundo Vazquez (2008, p. 102) são exemplos de SE:

- Relatórios com totalização de dados e que também atualizam arquivos;
- Consultas com cálculos ou apresentação de dados derivados;
- Arquivo de movimento gerado para outra aplicação;
- Informações em formato gráfico;
- Telas de *login*(com criptografia).

#### **c. Consultas Externas (CE)**

Correspondem a transações cujo objetivo é a apresentação de informações aos usuários, por meio da recuperação de dados ou informações de controle de ALIs e AIEs.

Segundo Vazquez (2008, p. 102) são exemplos de SE:

- Caixas de listagem;
- Telas de login (sem criptografia);
- Menus gerados dinamicamente com base em configuração da aplicação;
- Informações em formato gráfico.

#### d. Complexidade e Contribuição (EE,SE e CE)

Conforme Vazquez (2008), cada função do tipo de transação pode ser classificada, conforme o número de Tipos de Dados (TD) referenciados e o número de Arquivos Referenciados (AR), como baixa, média e alta de acordo com a Tabela 4 – FPA: Complexidade EE, SE e CE.

**Tabela 4– FPA: Complexidade EE, SE e CE**

Arquivos Referenciados	Tipos de Dados				
	SE e CE		< 6	6 – 19	> 19
		EE	< 5	5 – 15	> 15
	< 2	< 2	Baixa	Baixa	<b>Média</b>
	2 – 3	2	Baixa	<b>Média</b>	Alta
> 3	> 2	<b>Média</b>	Alta	Alta	

Fonte: Vazquez (2008, p. 108)

Um TD pode ser identificado por ser um campo único, reconhecido pelo usuário e um AR por ser um ALI ou AIE lido ou mantido pela função do tipo transação.

Após classificar a complexidade deve-se calcular a contribuição das funções do tipo transação utilizando a Tabela 5– FPA: Contribuição das funções do tipo de transação.

**Tabela 5–FPA: Contribuição das funções do tipo de transação**

Tipo de Função	Baixa	Média	Alta
Entrada Externa / Consulta Externa	3 PF	4 PF	6 PF
Saída Externa	4 PF	5 PF	7 PF

#### 2.1.5. Determinar a Contagem de Pontos de Função não Ajustados

Segundo Anselmo (2010) o cálculo de pontos de função não ajustados pode ser resumido conforme a Tabela 6 – FPA: Pontos de função não ajustados, onde o número de pontos de função é igual a soma da quantidade de funções multiplicada pelo fator de contribuição de sua complexidade.

**Tabela 6–FPA: Pontos de função não ajustados**

<b>Tipo</b>	<b>Complexidade Funcional</b>	<b>Qtde. X</b>	<b>Total Complexidade</b>	<b>Valor</b>
<b>ALI</b>	Simple	7	ALI-S	ALI-S + ALI-M + ALI-C
	Médios	10	ALI-M	
	Complexos	15	ALI-C	
<b>AIE</b>	Simple	5	AIE-S	AIE-S + AIE-M + AIE-C
	Médios	7	AIE-M	
	Complexos	10	AIE-C	
<b>EE</b>	Simple	3	EE-S	EE-S + EE-M + EE-C
	Médios	4	EE-M	
	Complexos	6	EE-C	
<b>SE</b>	Simple	4	SE-S	SE-S+ SE+M + SE-C
	Médios	5	SE-M	
	Complexos	7	SE-C	
<b>CE</b>	Simple	3	CE-S	CE-S + CE-M + CE-C
	Médios	4	CE-M	
	Complexos	6	CE-C	

#### **2.1.6. Determinar o valor do fator de ajuste**

Segundo Anselmo (2010), baseado no manual da IFPUG o fator de ajuste (FA) é baseado em quatorze características gerais de sistema que determinam a funcionalidade geral da aplicação que está sendo contada, são estas:

1. Comunicação de Dados;
2. Processamento de Dados Distribuído (Funções Distribuídas);
3. Performance;
4. Configuração do equipamento;
5. Volume de Transações;
6. Entrada de Dados On-Line;
7. Interface com o usuário;

8. Atualização On-Line;
9. Processamento Complexo;
10. Reusabilidade;
11. Facilidade de Implantação;
12. Facilidade Operacional;
13. Múltiplos Locais;
14. Facilidade de mudanças.

Para a determinação do valor do fator de ajuste é atribuindo um grau de influência, que varia em uma escala de zero a cinco, a cada característica, conforme abaixo:

- 0 - Nenhuma influência;
- 1 - Influência mínima;
- 2 - Influência moderada;
- 3 - Influência média;
- 4 - Influência significativa;
- 5 - Influência forte.

O somatório do grau de influência (GI) obtido pela classificação das características deve ser aplicado na formula a seguir para obter o fator de ajuste (FA):

$$FA = (GI * 0,01) + 0,65$$

**Equação 1 – FPA: Fatore de ajuste**

Vale ressaltar que os valores fixos presentes nas formulas, como por exemplo a variação de 0,65 até 1,35 (35%), resultam dos estudos feitos com base em projetos passados e análises empíricas sob estes dados.

#### **2.1.7. Determinar o Número de Pontos de Função Ajustados**

A seguir serão apresentados os cálculos de pontos de função ajustados segundo Vazquez (2008), o qual utilizou as fórmulas descritas no manual do IFPUG utilizando os mesmo nomes de variáveis.

- **Projeto de Desenvolvimento**

$$DFP = (UFP + CPF) * VAF$$

**Equação 2 – FPA: Projeto de desenvolvimento**

Onde:

DFP = Número de pontos de função do projeto de desenvolvimento;

UFP = Número de pontos de função não ajustados;

CFP = Número de pontos de função não ajustados das funções de conversão;

VAF=Valor do fator de ajuste.

#### **e. Projeto de Melhoria**

$$\mathbf{EFP = [(ADD + CHGA + CFP) * VAFA] + (DEL * VAFB)}$$

#### **Equação 3 – FPA: Projeto de melhoria**

Onde:

EFP=Número pontos de função do projeto de melhoria;

ADD = Número de pontos de função não ajustados das funções incluídas pelo projeto de melhoria;

CHGA= Número de pontos de função não ajustados das funções modificadas. Reflete as funções depois da modificação;

CFP=Número de pontos de função não ajustados adicionados pela conversão;

VAFA = Valor do fator de ajuste da aplicação depois do projeto de melhoria;

DEL= Número de pontos de função não ajustados das funções excluídas pelo projeto de melhoria;

VAFB= Valor do fator de ajuste da aplicação antes do projeto de melhoria.

#### **f. Aplicação**

$$\mathbf{AFP = ADD \times VAF}$$

#### **Equação 4 – FPA: Aplicação**

Onde:

AFP=Número de pontos de função ajustados da aplicação;

ADD =Pontos de função não ajustados das funções instaladas;

VAF= Valor do fator de ajuste da aplicação.

## **2.2. Pontos por Casos de Uso**

A técnica de modelagem de casos de uso tem sido amplamente utilizada para

descrever e capturar os requisitos funcionais de software. Uma vez mapeado este cenário faz sentido pensar em realizar a estimativa de tamanho e recursos baseada sobre tal modelagem. (DAMODARAN, 2003)

Foi pensando neste contexto que Gustav Karner em 1993, propôs a metodologia dos Pontos por Caso de Uso (*Use Case Point* - UCP), com o intuito de estimar recursos para os projetos já na fase de levantamento dos casos de uso.

Esta métrica utiliza a produtividade média da organização para produzir um UCP aliado as medições feitas com base no digrama de casos de uso, como: contagem de atores e casos de uso e identificação de sua complexidade, determinação da complexidade de fatores técnicos e ambientais e cálculos de UCP ajustados.

O processo de medição desta metodologia consiste resumidamente em seis passos:

- Contar os atores e identificar a sua complexidade;
- Contar os casos de uso e identificar o grau de complexidade;
- Somar o total de atores com o total de casos de uso para obter o UCP não ajustado;
- Determinar a complexidade do fator técnico;
- Determinar a complexidade do fator ambiental;
- Calcular o UCP ajustado.

A seguir é apresentada cada etapa do processo conforme descrito por Anselmo (2010).

### 2.2.1. Contagem dos Atores

A identificação dos atores auxilia no estabelecimento das fronteiras, na compreensão da finalidade e na extensão do sistema.

**Tabela 7–UCP: Tipos de ator x peso**

<b>Tipo de Ator</b>	<b>Descrição</b>	<b>Peso</b>
<b>Simple</b>	Quando ator representa outros sistemas, que se comunicam com o Software através de APIs de comunicação (SOAP, serviço WEB, entre outros).	1
<b>Médio</b>	Quando ator representa um sistema externo acessado através da rede onde não existe um protocolo/interface de comunicação definida.	2
<b>Complexo</b>	Quando o ator interage através de uma interface gráfica (GUI).	3

Fonte: Anselmo (2010, p. 92)

A classificação dos atores envolvidos em cada Caso de Uso é feita conforme Tabela 7, onde o ator é classificado conforme o papel que representa para a aplicação.

Para determinar o peso total dos atores deverá ser multiplicado o número de cada tipo de ator pelo seu peso, para posterior soma destes valores. O valor encontrado a partir deste cálculo é o UAW (*Unadjusted Actor Weight*).

### 2.2.2. Contagem dos Casos de Uso

Existem duas formas de classificar a complexidade dos casos de uso, considerando o número de transações envolvidas no seu processamento ou considerando o número de classes (entidades) envolvidas no processo do caso de uso.

Na Tabela 8 – UCP: Tipos de casos de uso x peso por transações, são apresentadas as regras para a classificação dos caso de uso utilizando o primeiro conceito, dividindo-se em três níveis de complexidade, de acordo com o número de transações envolvidas no seu processamento.

**Tabela 8–UCP: Tipos de casos de uso x peso por transações**

<b>Tipo de Caso de Uso</b>	<b>Descrição</b>	<b>Peso</b>
<b>Simple</b>	Quantidade de passos no fluxo for no máximo 3 caminhos.	5
<b>Médio</b>	Quantidade de passos no fluxo estiver entre 4 e 7 caminhos.	10
<b>Complexo</b>	Quantidade de passos no fluxo for maior que 7 caminhos.	15

**Fonte: Anselmo (2010, p. 93)**

Levando em consideração ao invés do número de transações o número de classes (entidades) envolvidas no processo, pode ser utilizada a Tabela 9 – UCP: Tipos de casos de uso x peso por classes (entidades).

O peso total dos atores do sistema UUCW (*Unadjusted Use Case Weight*) é calculado pela soma dos produtos do número de casos de uso de cada tipo pelo respectivo peso.

**Tabela 9–UCP: Tipos de casos de uso x peso por classes (entidades)**

<b>Tipo de Caso de Uso</b>	<b>Descrição</b>	<b>Peso</b>
<b>Simple</b>	Necessitar de até 5 classes de análise.	5
<b>Médio</b>	Necessitar de 5 a 10 classes de análise.	10
<b>Complexo</b>	Necessitar de mais de 10 classes de análise.	15

Fonte: Anselmo (2010, p. 93)

### **2.2.3. Equação para Obter o Valor dos Pontos por Caso de Uso não Ajustados**

Para obter o valor dos casos de uso não ajustados é necessário somar o total de atores com o total de Casos de Uso. O valor encontrado a partir deste cálculo, demonstrado abaixo, é o UUCP (*Unadjusted Use Case Point*).

$$\text{UUCP} = \text{UAW} + \text{UUCW}$$

### **Equação 5 – UCP: Valor dos pontos por caso de uso não ajustados**

### **2.2.4. Cálculo dos Fatores de Ajuste**

Semelhante a análise de pontos de função, o fator de ajuste do UCP é constituído por duas partes, ambas geram multiplicadores que serão aplicados ao UUCP. Vale ressaltar que assim como os demais modelos de estimativa, estes fatores e seus pesos utilizados são resultado do conhecimento empírico adquirido através da análise de dados históricos nos estudos feitos pelos autores das métricas.

A primeira parte consiste nos Fatores de Complexidade Técnica (*Technical Complexity Factor- TCF*), onde são considerados treze fatores cada um com determinado peso. A Tabela 10 – UCP: Fatores de complexidade técnica, apresenta cada um dos fatores e seus respectivos pesos.

Para realizar o cálculo do TCF deve-se:

- Classificar cada fator em uma escala de zero a cinco, sendo que quanto mais baixo o valor atribuído menor a influência do fator;
- Multiplicar o valor atribuído para cada fator ao respectivo peso;
- Aplicar a equação a seguir:

$$\text{TCF} = 0.6 + (0.01 \times (\text{TFactor}))$$

**Equação 6 – UCP: Fatores de complexidade técnica**

Onde: TFactor = Somatório do fator 1 ao 13.

**Tabela 10–UCP: Fatores de complexidade técnica**

<b>Fator</b>	<b>Requisito</b>	<b>Peso</b>
1	Sistema é distribuído	2
2	Performance	1
3	Eficiência para o usuário final	1
4	Processamento interno complexo	1
5	Código deve ser reusável	1
6	Fácil para instalar	0,5
7	Fácil para usar	0,5
8	Portátil	2
9	Fácil de mudar	1
10	Concorrente	1
11	Seguro	1
12	Acessível por terceiros	1
13	Requer treinamento especial	1

Fonte: Anselmo (2010, 94)

A segunda parte do cálculo dos fatores de ajuste leva em consideração o Cálculo dos Fatores Ambientais (*Environmental ComplexityFactor* - ECF), onde são considerados oito fatores conforme descrito na Tabela 11 – UCP:

Fatores de complexidade ambiental. Para realizar o cálculo do ECF deve-se: classificar cada fator em uma escala de zero a cinco, sendo que quanto mais baixo o valor atribuído menor a influência do fator, multiplicar o valor atribuído para cada fator ao respectivo peso e aplicar a equação a seguir, onde: EFactor = Somatório dos fatores de um a oito.

$$\text{ECF} = 1.4 + (-0.03 \times (\text{EFactor}))$$

**Equação 7 – UCP: Fator de complexidade ambiental**

**Tabela 11 – UCP: Fatores de complexidade ambiental**

<b>Fator</b>	<b>Requisito</b>	<b>Peso</b>
1	Familiaridade da Equipe com o Projeto com os processos de desenvolvimento de Software	1,5
2	Experiência na Aplicação	0,5
3	Equipe com Experiência em OO (Orientação a Objetos)	1
4	Capacidade de Análise	0,5
5	Motivação	1
6	Requisitos Estáveis	2
7	Recursos em Meio Período ( <i>Part-Time</i> )	-1
8	Dificuldade na Linguagem de programação	-1

Fonte: Anselmo (2010, 95)

### 2.2.5. Equação para Obter o Valor dos Pontos por Casos de Uso Ajustados

Para obter o valor dos casos de uso ajustados é necessário multiplicar o valor dos UUCP pelos valores dos fatores de ajuste ambientais e técnicos. O valor encontrado a partir deste cálculo, demonstrado abaixo, é o valor do UCP:

$$\text{UCP} = \text{UUCP} \times \text{TCF} \times \text{ECF}$$

#### Equação 8 – UCP: Valor dos pontos por casos de uso

A partir daí para realizar o cálculo de estimativa de horas de programação pode-se aplicar a equação abaixo:

$$\text{H} = \text{UCP} * \text{pessoas/hora por unidade de UCP}$$

#### Equação 9 – UCP: Estimativa de horas de programação

## 2.3. COCOMO

O COCOMO (*Constructive Cost Model*) busca medir os seguintes fatores, necessários para o desenvolvimento de software, tendo como premissa que se tenha a dimensão do mesmo: esforço, prazo, tamanho da equipe e custo.

Segundo Sommerville (2007) o primeiro modelo COCOMO 81, proposto em 1981 por

Boehm, é dividido em três níveis: Básico, Intermediário e Avançado, ambos os níveis são calculados através do tamanho estimado.

O nível básico considera somente o tamanho estimado em linha de códigos, o intermediário considera também um conjunto de direcionadores de custo e avançado considera, além destas características, a avaliação de impacto dos atributos do software e da equipe, podendo ser aplicado em diferentes fases do projeto.

As equações e parametrizações do COCOMO foram obtidas inicialmente através do estudo de 63 projetos, além disso, Boehm considerou a opinião subjetiva de gerentes de software e os resultados de outros modelos de estimativa. Posteriormente os parâmetros iniciais foram refinados e calibrados com projetos adicionais.

O modelo COCOMO 81 foi apresentado pensando no processo de desenvolvimento cascata, utilizando linguagens de programação imperativas padronizadas (como C ou FORTRAN). Deste então, ocorreram mudanças radicais no desenvolvimento de software, como a prototipação e o desenvolvimento incremental. Sommerville (2007)

Com o objetivo de obter um modelo que contemplasse as novas necessidades dos projetos de software com conceitos mais atualizados, em 1994 começou a ser desenvolvido o COCOMO II por iniciativa da USC (*University of Southern California*). Além da USC o projeto teve a colaboração de algumas empresas como membros afiliados.

Como resultado do projeto foi publicado um novo manual o qual propunha três modelos de estimativa: *Application Composition*, *Early Design* e *Post-Architecture*. Nos subcapítulos a seguir será feita a descrição destes modelos e para isto será utilizada como base a Versão 2.1 deste manual (*COCOMO II Model Definition Manual*) publicado em 2000.

### **2.3.1. Modelo Post-Architecture**

O modelo *Post-Architecture* é o mais completo e conseqüentemente o que precisa de mais informações disponíveis para a sua aplicação, portanto, conforme Pressman (2006), é usado durante a construção do software.

O cálculo do esforço de desenvolvimento deste modelo, conforme descrito no manual do COCOMO II, é composto pela medição de tamanho, ajustado através de dezessete direcionadores de custo e da aplicação de cinco fatores de escala.

### a. Medição de Tamanho

O tamanho da aplicação deve ser estimado em milhares de linhas de código (KSLOC–K *Source Lines of Code*), porém, pode-se utilizar a contagem de pontos de função não ajustados e posteriormente convertê-los em KSLOC.

**Tabela 12–COCOMO: Taxas de conversão de UFP para SLOC**

Linguagem	SLOC/UFP	Linguagem	SLOC/UFP
Access	38	Jovial	107
Ada 83	71	Lisp	64
Ada 95	49	MachineCode	640
Al Shell	49	Modula 2	80
APL	32	Pascal	91
Assembly – Basic	320	PERL	27
Assembly – Macro	213	PowerBuilder	16
Basic – ANSI	64	Prolog	64
Basic – Compiled	91	Query – Default	13
Basic – Visual	32	ReportGenerator	80
C	128	SecondGenerationLanguage	107
Cobol (ANSI 95)	55	Simulation – Default	46
Database – Default	40	Spreadsheet	6
Fifth Generation Language	4	Third Generation Language	80
First Generation Language	320	Unix Shell Scripts	107
Forth	64	USR_1	1
Fortran 77	107	USR_2	1
Fortran 95	71	USR_3	1
FourthGenerationLanguage	20	USR_4	1
High LevelLanguage	64	USR_5	1
HTML 3.0	15	Visual Basic 5.0	29
Java	53	Visual C++	34

Fonte: Model Definition Manual (2000, p. 6)

Primeiramente deve ser aplicada a estimativa de pontos de função não ajustados (UFP

– *Unadjusted Function Points*), conforme já verificado. A seguir para realizar a conversão devem ser aplicados, de acordo com a linguagem de programação, os valores obtidos pelo UFP conforme Tabela 12 – COCOMO: Taxas de conversão de UFP para SLOC.

#### **b. Direcionadores de Custo (*Cost Drivers*)**

Os dezessete direcionadores de custo são utilizados como forma de tornar a estimativa de acordo com as características do ambiente de desenvolvimento, levando em consideração fatores dos contextos de produto, plataforma, pessoal e projeto.

Os direcionadores referente ao produto são:

- Confiabilidade requerida do software (*Required Software Reliability- RELY*): para estimar o nível de confiabilidade que o sistema precisa oferecer deve ser observado o impacto que uma falha pode causar, sendo este impacto baixo ou alto o valor do RELY seguirá o mesmo padrão;
- Tamanho da base de dados (*Data Base Size- DATA*): este direcionador permite estimar a necessidade da aplicação de testes em escala para a verificação da performance dos programas;
- Complexidade do produto (*Product Complexity- CPLX*): A complexidade é dividida em cinco áreas e estimada de forma subjetiva, onde deve-se combinar as características das áreas conforme o produto que está sendo medido. As áreas consideradas são: operações de controle, operações computacionais, operações dependentes de algum tipo de máquina, operações de gerenciamento de dados e operações de gerenciamento de interface de usuário;
- Reusabilidade requerida (*Developed for Reusability - RUSE*): este direcionador leva em consideração a necessidade de utilização dos componentes desenvolvidos em outras aplicações, pois neste caso, deverá ser considerado um esforço maior para a construção e validação dos mesmos;
- Documentação associada às necessidades do ciclo de vida (*Documentation Match to Life-CycleNeeds- DOCU*): considera-se o esforço para documentação necessário de acordo com as etapas do ciclo de vida em que a mesma deverá ser construída e mantida.

Os direcionadores referentes a plataforma são:

- Restrição de tempo de execução (*Execution Time Constraint - TIME*): este direcionador refere-se a restrição de utilização do tempo de execução disponível;
- Restrição de armazenamento principal (*Main Storage Constrain- STOR*): indica a

restrição imposta de armazenamento principal ao sistema. Apesar do aumento do tempo de execução disponível, este direcionador é mantido em virtude do também contínuo aumento do consumo dos recursos disponíveis;

- Volatilidade da plataforma (*Platform Volatility* - PVOL): este direcionador considera a volatilidade das plataformas de hardware e sistema operacional, e/ou outros sistemas, utilizados como base no desenvolvimento da aplicação.

Os direcionadores referentes aos fatores pessoal são:

- Capacidade do analista (*Analyst Capability*- ACAP): as tendências atuais enfatizam a importância de analistas altamente capazes. Este direcionador considera os fatores mais relevantes com relação aos conhecimentos do profissional que irá realizar a análise do sistema, tais como a própria capacidade e eficiência de análise, design, comunicação e colaboração com a equipe;
- Capacidade dos programadores (*Programmer Capability* - PCAP): assim como o ACAP este direcionador considera fatores relevantes com relação aos conhecimentos da equipe de programação, como a eficiência, habilidade e meticulosidade na programação, assim como a experiência, capacidade de comunicação e cooperação;
- Rotatividade de pessoal (*Personnel Continuity*- PCON): este direcionador considera a rotatividade anual de pessoal;
- Experiência na aplicação (*Applications Experience* - APEX): considera a experiência da equipe envolvida no projeto de desenvolvimento na aplicação ou no tipo de aplicação que está sendo construído;
- Experiência na plataforma de desenvolvimento (*Platform Experience* - PLEX): este direcionador considera o grau de compreensão de plataformas mais robustas, incluindo interfaces gráficas de usuários, redes e capacidade de processamento distribuído;
- Experiência com as linguagens e ferramentas de desenvolvimento (*Language and Tool Experience* - LTEX): avalia a experiência da equipe envolvida no projeto de desenvolvimento com relação a linguagem e ferramentas utilizadas, salientando que tais ferramentas englobam as relativas a programação mas também as referentes a análise.

Os direcionadores referentes aos fatores de projeto são:

- Utilização de Ferramentas de Software (*Use of Software Tools* - TOOL): considera

as funcionalidades disponíveis nas ferramentas utilizadas para o desenvolvimento de software, desde as que oferecem apenas a edição e codificação, até o gerenciamento integrado do ciclo de vida;

- Desenvolvimento distribuído (*Multisite Development- SITE*): considera o desenvolvimento em ambientes fisicamente distribuídos, verificando fatores de distribuição geográfica e comunicações de apoio;
- Cronograma de desenvolvimento requerido (*Required Development Schedule - SCED*): considera as restrições de cronograma impostas a equipe de desenvolvimento, uma vez que cronogramas mais acelerados normalmente exigem um esforço maior de desenvolvimento nas fases finais do projeto.

### c. Fatores de Escala (*Scale Factors*)

Os fatores de escala visam balancear os gastos e economias dos projetos. Estes são divididos em cinco fatores, conforme o manual do COCOMO II, apresentados a seguir:

- Precedência (*Precedentedness- PREC*): verifica se as características do sistema a ser desenvolvido, são similares as de projetos anteriores;
- Flexibilidade de desenvolvimento (*Development Flexibility- FLEX*): indica o nível de flexibilidade permitida quanto aos requisitos estabelecidos;
- Arquitetura e resolução de risco (*Architecture/RiskResolution- RESL*): avalia os níveis de riscos envolvidos em aspectos como, cronograma, orçamento, incertezas na arquitetura.
- Coesão da equipe (*Team Cohesion- TEAM*): representa possíveis problemas referentes as diferentes culturas e objetivos a curto prazo dos envolvidos no projeto nas mais diversas áreas da empresa, deste desenvolvedores e responsáveis pelo suporte até consultores e clientes.
- Maturidade do processo (*Process Maturity- PMAT*): esta escala é obtida através do modelo *Capability Maturity Model*(CMM) do *Software Engineering Institute* (SEI).

Para o cálculo dos direcionadores de custo e fatores de escala apresentados, devemos classifica-los na escala representada nas tabelas 13 e 14 a seguir, que variam de muito baixo à extra alto, conforme calibração descrita no manual COCOMO II.

Para alguns direcionadores de custo as escalas “Muito Baixo” e “Extra Alto” não são aplicadas, por isso, estes aparecem em branco na Tabela 13. O multiplicador de custo é encontrado pela multiplicação do peso atribuído a cada multiplicador.

**Tabela 13–COCOMO: Direcionadores de custos**

<b>Direcionadores de Custo</b>	<b>Classificação – Multiplicadores de Esforço</b>					
	<b>Muito Baixo</b>	<b>Baixo</b>	<b>Nominal</b>	<b>Alto</b>	<b>Muito Alto</b>	<b>Extra Alto</b>
RELY	0,82	0,92	1,00	1,10	1,26	
DATA		0,90	1,00	1,14	1,28	
CPLX	0,73	0,87	1,00	1,17	1,34	1,74
RUSE		0,95	1,00	1,07	1,15	1,24
DOCU	0,81	0,91	1,00	1,11	1,23	
TIME			1,00	1,11	1,29	1,63
STOR			1,00	1,05	1,17	1,46
PVOL		0,87	1,00	1,15	1,30	
ACAP	1,42	1,19	1,00	0,85	0,71	
PCAP	1,34	1,15	1,00	0,88	0,76	
PCON	1,29	1,12	1,00	0,90	0,81	
APEX	1,22	1,10	1,00	0,88	0,81	
PLEX	1,19	1,09	1,00	0,91	0,85	
LTEX	1,20	1,09	1,00	0,91	0,84	
TOOL	1,17	1,09	1,00	0,90	0,78	
SITE	1,22	1,09	1,00	0,93	0,86	0,80
SCED	1,43	1,14	1,00	1,00	1,00	

**Fonte: Model Definition Manual (2000)**

Já o valor dos fatores de escala é dado pela soma dos pesos atribuídos a cada fator, neste caso quando um fator não é aplicado em alguma escala atribui-se ao mesmo zero, conforme pode ser observado na Tabela 14 – COCOMO: Fatores de Escala.

**Tabela 14–COCOMO: Fatores de escala**

Fatores de Escala	Classificação					
	Muito Baixo	Baixo	Nominal	Alto	Muito Alto	Extra Alto
PREC	6,20	4,96	3,72	2,48	1,24	0,00
FLEX	5,07	4,05	3,04	2,03	1,01	0,00
RESL	7,07	5,65	4,24	2,83	1,41	0,00
TEAM	5,48	4,38	3,29	2,19	1,10	0,00
PMAT	7,80	6,24	4,68	3,12	1,56	0,00

Fonte: Model Definition Manual (2000)

A seguir são apresentadas as equações para estimar o esforço no modelo *Post-Architecture*, representadas com as mesmas variáveis do manual COCOMO II, as quais utilizam as informações vistas anteriormente.

Primeiramente é necessário calcular o expoente referente aos fatores de escala, conforme equação 10, que faz parte da equação de estimativa do esforço do modelo *Post-Architectur*.

$$E = B + 0.01 \times \sum_{j=1}^5 SF_j$$

**Equação 10 – COCOMO: Expoente referente aos fatores de escala**

B = Escala base para a equação de esforço que pode ser calibrada – Padrão atual definido = 0,91;

SF = 5 Fatores de Escala.

$$PM = A \cdot \text{Size}^E \times \prod_{i=1}^{17} EM_i + PM_{\text{Auto}}$$

$$E = B + 0.01 \times \sum_{j=1}^5 SF_j$$

Adapted SI OCV (AT/ )

**Equação 11 – COCOMO: Esforço modelo *Post-Architecture***

A = Coeficiente de esforço que pode ser calibrado – Padrão atual definido por 2,94;

AT = Percentual de código referente a reengenharia de tradução automática;

ATPROD = Produtividade da tradução automática;

B = Escala base para a equação de esforço que pode ser calibrada – Padrão atual definido = 0,91;

E = Expoente referente aos fatores de escala calculado;

EM = Multiplicadores de esforço referentes aos 17 direcionadores de custo;

PM = Esforço pessoa/mês;

PM Auto = Esforço pessoa/mês em atividades de tradução automática;

SF = 5 Fatores de Escala.

### 2.3.2. Modelo Early Design

Este modelo utiliza a medição de tamanho e direcionadores de custo para realizar a estimativa, podendo ser aplicado após os requisitos terem sido estabilizados e a arquitetura básica ter sido estabelecida. PRESSMAN (2006)

O tamanho do software é também estimado em milhares de linhas de código (KSLOC), podendo ser utilizada a estimativa de Pontos de Função da mesma forma que é feito no modelo *Post-Architecture*.

Deve-se ajustar a estimativa inicial utilizando sete direcionadores de custos, estes são um modelo simplificado, pois alguns direcionadores derivam da combinação de direcionadores do modelo *Post-Architecture*, conforme pode ser observado na Tabela 15.

**Tabela 15– COCOMO: Combinação dos direcionadores de custo**

<b>Direcionadores de Custo</b>	
<b>Modelo Early Design</b>	<b>Combinação de Direcionadores Modelo Post-Architecture</b>
PERS	Capacidade do analista (ACAP), Capacidade dos programadores (PCAP), Rotatividade de pessoal (PCON)
RCPX	Confiabilidade requerida do software (RELY), Tamanho da base de dados (DATA), Complexidade do Produto(CPLX), Documentação associada às necessidades do ciclo de vida (DOCU)
RUSE	Reusabilidade requerida (RUSE)
PDIF	Restrição de tempo de execução (TIME), Restrição de armazenamento principal (STOR), Volatilidade da plataforma (PVOL)
PREX	Experiência na aplicação (APEX), Experiência na plataforma de desenvolvimento(PLEX), Experiência com as linguagens e ferramentas de desenvolvimento (LTEX)
FCIL	Utilização de ferramentas de software (TOOL), Desenvolvimento distribuído (SITE)
SCED	Cronograma de desenvolvimento requerido (SCED)

**Fonte: Model Definition Manual (2000, p.36)**

A seguir são apresentadas as equações do modelo *Early Design*, representadas com as

mesmas variáveis do manual COCOMO II, as quais utilizam as informações vistas acima.

$$PM = A \cdot \text{Size}^E \times \prod_{i=1}^7 EM_i + PM_{\text{Auto}}$$

### **Equação 12 - COCOMO: Esforço modelo *Early Design***

A = Coeficiente de esforço que pode ser calibrado – Padrão atual definido por 2,94;

E = Expoente referente aos fatores de escala calculado;

EM = Multiplicadores de esforço referentes aos 7 direcionadores de custo;

PM = Esforço pessoa/mês;

PM Auto = Esforço pessoa/mês em atividades de tradução automática;

SF = 5 Fatores de Escala.

### **2.3.3. Modelo Application Composition**

Este modelo é voltado ao desenvolvimento de aplicações de forma rápida, normalmente em ambientes de pequenas equipes. Neste contexto são utilizados conceitos de composição de componentes e prototipação.

Segundo Pressman (2006) este pode ser utilizado durante os primeiros estágios da engenharia de software.

Para obter o tamanho da aplicação é utilizada a contagem de Pontos de Objeto, uma medida indireta de software calculada através da contagem da quantidade de telas, relatórios e componentes necessários para construir a aplicação. PRESSMAN (2006)

A seguir é descrito resumidamente os passos para a realização do cálculo de pontos de objeto conforme Pressman (2006).

#### **a. Passo 1**

Identificação dos objetos e classificação conforme o tipo: telas, relatórios e componentes.

#### **b. Passo 2**

Atribuição do grau de complexidade, sendo este classificado em simples, médio ou difícil, de acordo com critérios estabelecidos por Boehm durante seus estudos, através da análise das bases históricas de projetos.

### c. Passo 3

Ponderar a quantidade de telas, relatórios e componentes obtidas através da complexidade atribuída, conforme Tabela 16.

**Tabela 16 – COCOMO: Sistema de classificação de pontos por objeto**

Tipo de Objeto	Peso da complexidade		
	Simple	Médio	Difícil
Tela	1	2	3
Relatório	2	5	8
Componente			10

Fonte: Pressman (2006, p. 535)

### d. Passo 4

Verificar se o desenvolvimento será baseado em componentes ou em reuso de software, pois caso seja necessário utilizar estes conceitos, deve-se ajustar a contagem conforme a fórmula a seguir, onde entende-se por NOP, novos pontos por objeto.

$$\text{NOP} = (\text{pontos por objeto}) \times [(100 - \% \text{ de reuso})/100]$$

**Equação 13 - COCOMO: Novos pontos por objeto**

### e. Passo 5

Aplicar a taxa de produtividade, obtendo assim o esforço estimado para o projeto.

$$\text{Esforço estimado} = \text{NOP}/\text{PROD}$$

**Equação 14 - COCOMO: Esforço modelo *Application Composition***

Sendo que a produtividade (PROD) pode ser obtida através da ponderação da Tabela 17– COCOMO: Taxas de produtividade para pontos por objeto.

**Tabela 17–COCOMO: Taxas de produtividade para pontos por objeto**

Experiência/capacidade do desenvolvedor	Muito baixa	Baixa	Normal	Alta	Muito Alta
Maturidade/capacidade do ambiente	Muito baixa	Baixa	Normal	Alta	Muito Alta
PROD	4	7	13	25	50

Fonte: Pressman (2006, p. 535)

### 2.3.4. Equação do Tempo de Desenvolvimento

A equação de tempo de desenvolvimento mostra a quantidade de meses necessária para o desenvolvimento da aplicação, levando em consideração o esforço calculado e as variáveis que podem ser calibradas. Abaixo é exibida a Equação 15 – COCOMO: Tempo de Desenvolvimento, conforme o manual COCOMO II.

$$TDEV = [C \times (PM_{NS})^F] \times \frac{SCED\%}{100}$$
$$F = (D + 0.2 \times [E - B])$$

#### Equação 15 - COCOMO: Tempo de Desenvolvimento

B = Escala base para a equação de esforço = 0,91;

C = Constante multiplicativa para tempo de desenvolvimento = 3,67;

D = Escala base que pode ser calibrada = 0,28;

E = Expoente referente aos fatores de escala;

PM<sub>NS</sub> = Esforço pessoa/mês estimado sem o direcionador de custo SCED;

SCED = Cronograma de desenvolvimento requerido;

TDEV = Tempo de desenvolvimento em meses.

### 2.3.5. Equipe Média e Custo do Projeto

A definição de tempo de trabalho por mês do COCOMO II para cada pessoa considera 152 horas, sendo este valor obtido através da experiência prática levando em consideração fatores como, licença médica, férias e feriados.

Com isso a equipe média do projeto é obtida através da divisão do esforço pelo tempo de desenvolvimento.

O custo do projeto então é calculado através do tempo de desenvolvimento, equipe média e do valor hora da equipe envolvida no projeto.

## 2.4. Considerações sobre as métricas

Conhecendo as características e o formato do cálculo utilizando pelo modelo FPA, observa-se que para a aplicação da métrica é necessário possuir conhecimento aprofundado no

processo de contagem, para assim poder equacionar a melhor forma de utilizá-lo.

Este método é regulamentado e padronizado por ISO o que lhe confere maior confiabilidade. Também é independente de tecnologia e pode ser utilizado aliado a qualquer forma de representação de requisitos.

Porém, para a sua aplicação é necessário possuir um conhecimento detalhado dos requisitos do projeto, o que acaba por privilegiar equipes com mais habilidades e mais experientes.

Apesar do FPA ser um modelo padronizado e regulamentado, a utilização dos fatores de ajuste é opcional por não ser prevista no padrão ISO/IEC de medição funcional. Sua aplicação baseia-se avaliação subjetiva de fatores, o que segundo Kitcheman (1997), pode causar variações na estimativa, mesmo considerando-se um mesmo projeto, quando estimado por diferentes pessoas da mesma organização.

A estimativa através de UCP é atrelada a utilização da modelagem UML, por este motivo, o ponto chave para utilização desta métrica é a aplicação da mesma.

Segundo Cohn (2006) é possível estabelecer uma média de tempo por UCP da organização, porém, para isso é importante que os casos de uso sejam descritos com o mesmo nível de detalhe. Com isso, Cohn ainda ressalta que o tamanho do aplicativo medido por UCP é independente da habilidade e experiência em medição da equipe que o implementa.

Da mesma forma que a análise de pontos de função, UCP pode ser considerada uma estimativa subjetiva e também pode apresentar diferenças quando executada por contadores distintos, pois utiliza a mesma lógica de fatores de ajuste já que seus conceitos derivam da análise de pontos de função.

Já a utilização de modelos paramétricos, como COCOMO II, está atrelada a calibração do modelo. Com isso, para sua implementação, independente do tipo de projeto, é necessária a calibração do modelo. Esta precisará ser feita com base em um histórico de projetos, para que sejam obtidos os coeficientes ajustados do modelo conforme o cenário em que será aplicado.

Conforme descrito por Aguiar (2002) os modelos paramétricos, são os recursos mais poderosos para a obtenção de estimativas nos projetos de software, porém, possuem um tratamento matemático mais complexo. Com isso a sua utilização deve ser precedida da preparação adequada juntamente de um estudo cuidadoso.

### **3. MAPEAMENTO DO PROCESSO DE DESENVOLVIMENTO**

Este trabalho pretende propor uma métrica de estimativa de tamanho de software, conforme já mencionado, aderente ao cenário de desenvolvimento de uma determinada empresa. Algumas características referentes a tal empresa serão ressaltadas a seguir.

A atualmente a empresa desenvolve em Oracle Forms 6i, uma linguagem de programação procedural, porém, pretende migrar para uma linguagem orientada a objetos.

Com o objetivo de alinhar os seus processos para a futura utilização da orientação a objetos, a empresa começou a utilizar o diagrama de casos de uso para o entendimento do escopo dos projetos.

A utilização da orientação a objetos juntamente com a modelagem UML é um ponto do planejamento estratégico da empresa para os próximos anos, visando a atualização tecnológica. Assim, a utilização do diagrama de casos de uso é o primeiro passo de uma preparação estrutural, com o intuito de possuir o processo de desenvolvimento orientado a casos de uso.

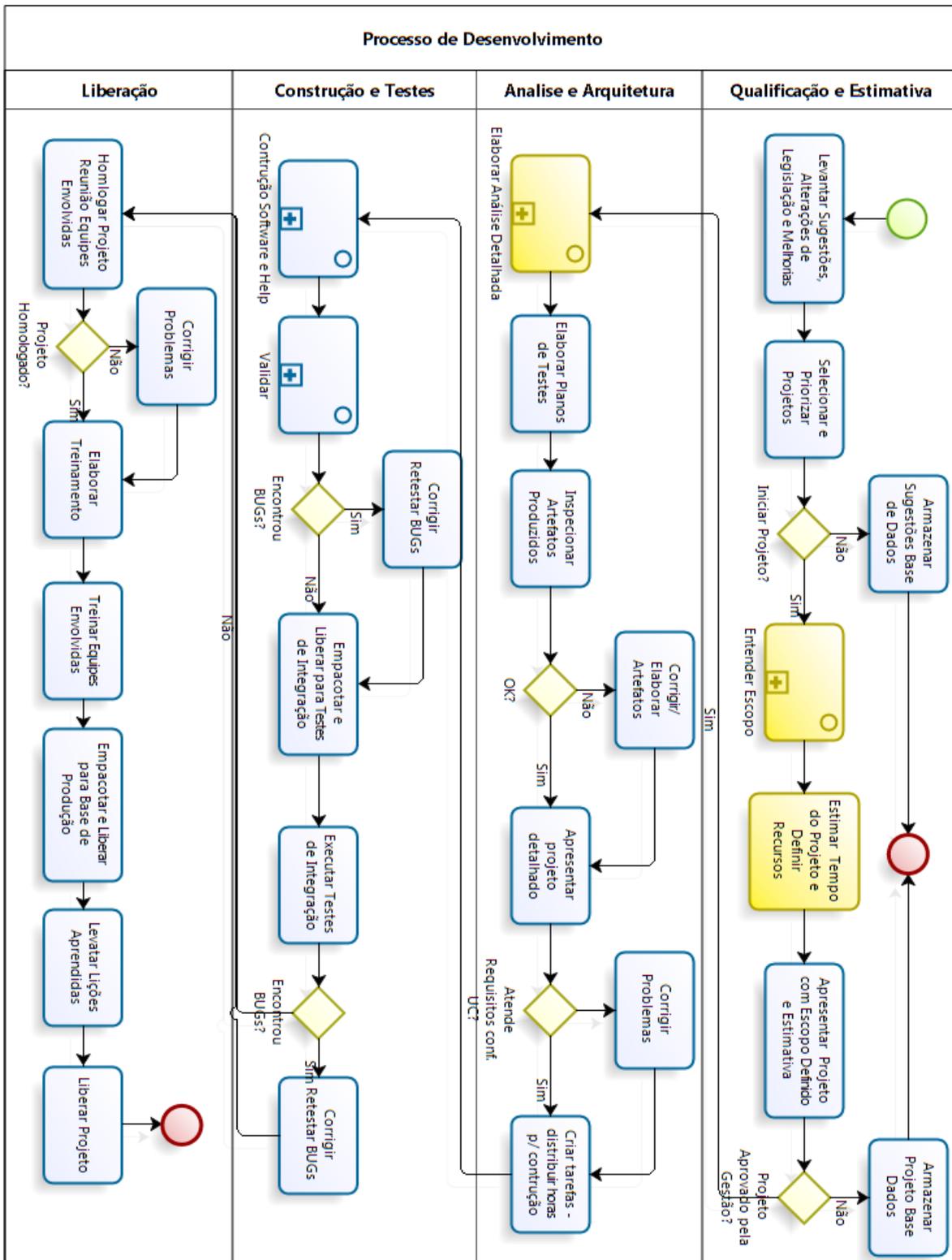
Por isso pretende-se avaliar o processo utilizado pela empresa atualmente, afim de auxiliar com os processos de medição de software nesta preparação para as mudanças de plataforma e modelagem, porém, tentado minimizar alterações no processo que inviabilizariam a aplicação imediata da métrica.

#### **3.1. Processo de Desenvolvimento Atual**

Para o mapeamento do processo foi utilizada a documentação já disponível na empresa adequando-a ao padrão de modelagem de processo BPMN (*Business Process Modeling Notation*). A Figura 3 apresenta o processo de desenvolvimento mapeado.

Com o objetivo de entender o cenário de desenvolvimento, foi realizado um levantamento do processo global, desde o levantamento do que será trabalhado nos projetos, até a liberação do mesmo.

O processo consiste em quatro etapas: qualificação/estimativa, análise/arquitetura, construção/testes e liberação. A seguir serão apresentados de forma resumida os processos executados em cada etapa.



**Figura 3 - Processo de desenvolvimento**

### 3.1.1. Qualificação e Estimativa

Esta é a etapa inicial do processo, onde são reunidas as necessidades de alterações e implementações dos produtos. Neste momento é formado um comitê que irá avaliar tais necessidades e indicar quais serão os projetos que serão desenvolvidos e sua priorização.

Em seguida deve ser entendido o escopo de cada projeto e realizada uma estimativa de tempo inicial, com o intuito de verificar a viabilidade de implementação e acomodar os diversos projetos concorrentes. Com isso é feito um planejamento da alocação de recursos humanos necessários para a construção das demais etapas.

Alguns sub-processos são mais relevantes e diretamente envolvidos na utilização de uma métrica, um destes é o levantamento do escopo, onde conforme pode ser observado, é o momento em que atualmente é feita a estimativa de tamanho do projeto.

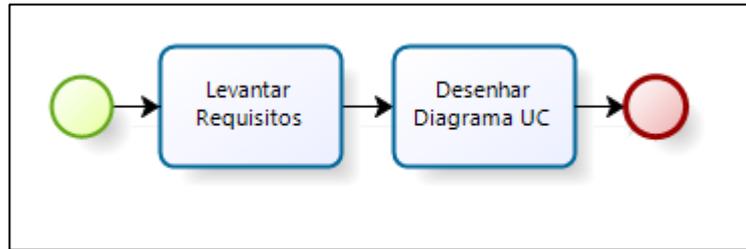
O processo de entendimento de escopo do projeto consiste, conforme sub-processo apresentado na Figura 4, no levantamento dos requisitos (funcionais e não funcionais) e o desenho do diagrama de casos de uso com base na descrição dos requisitos. Esta descrição gera uma parte do documento de especificação geral, o formato deste documento pode ser observado no anexo 1.

Para o levantamento dos requisitos são realizadas reuniões com os portadores do conhecimento de negócio envolvido no projeto, podendo este ser um consultor ou até mesmo um cliente.

Em seguida com base nos requisitos levantados e validados pelas partes, será realizada a identificação dos atores e casos de uso envolvidos no processo e por fim a construção do diagrama. No anexo 2 pode ser observado o formato deste artefato produzido em um projeto da empresa.

A estimativa do projeto ainda é feita em alguns casos de forma empírica, através do conhecimento do analista no sistema e área de negócio. Em outros casos, o que tende a se tornar o processo padrão para medição, utiliza-se a estimativa de Pontos por Casos de Uso com base no diagrama de casos de uso e no conhecimento do analista com relação aos impactos e estruturas necessárias para a implementação.

É importante observar que após a realização da estimativa, o tempo estimado e o escopo do projeto definido, estes serão apresentados às gerencias, que irão aprovar e determinar a continuidade ou não do mesmo.



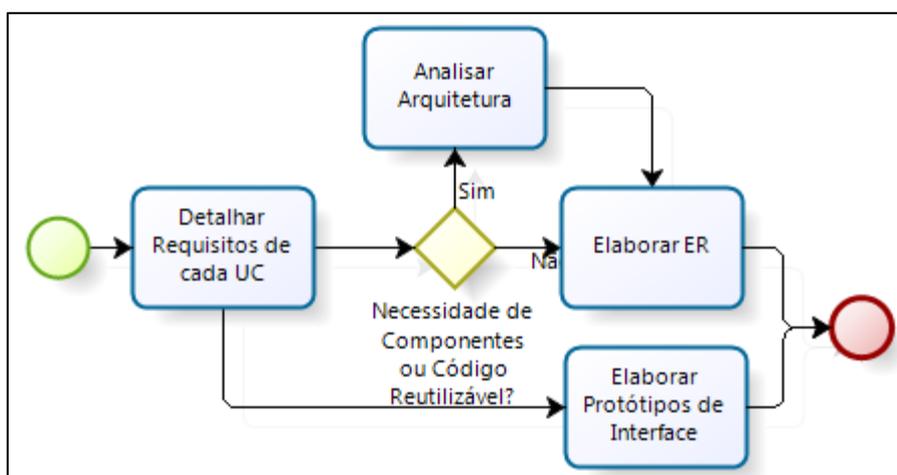
**Figura 4 - Sub-processo entender escopo**

### 3.1.2. Análise e arquitetura

Outro sub-processo relevante para a utilização de uma métrica e que deve ser observado é o de análise detalhada, pois é nesta etapa que o analista de sistema, conforme sub-processo apresentado na Figura 5, deve entender mais detalhadamente as necessidades do projeto.

A partir deste entendimento é gerado um documento de especificação geral, contendo o mapeamento das funcionalidades, os protótipos de interface das telas mais complexas e o desenho da estrutura física do banco de dados em formato de ER. Este documento também será utilizado posteriormente para a documentação, ou seja, geração dos helps do projeto.

Nota-se que não é utilizada nenhuma representação do modelo conceitual da estrutura de dados. É representado somente o modelo físico, onde são descritas diretamente as tabelas do banco de dados ao invés das entidades envolvidas no processo.



**Figura 5 - Sub-processo realizar análise detalhada**

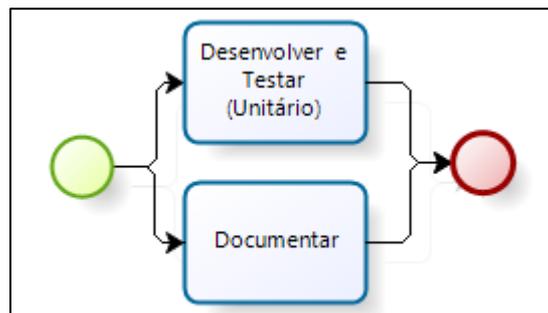
Com base no documento de especificação geral são construídos os cenários e casos de testes que serão aplicados após a construção do software. Estes casos de testes deverão

contemplar todos os requisitos da aplicação, incluindo conforme a necessidade testes de performance, de stress, entre outros.

Após a conclusão destas etapas serão auditados os artefatos produzidos e havendo necessidade, estes deverão ser corrigidos para que só depois disso seja dado início a próxima etapa. Esta auditoria será feita pelo papel de um analista par, que pode não ter acompanhado o processo de análise, mas terá a função de revisar os documentos gerados, verificando se estão completos e possibilitam o entendimento do projeto.

### 3.1.3. Construção e Testes

Esta etapa é composta basicamente por dois sub-processos, conforme figuras 6 e 7 a seguir, que consistem em construir, documentar e validar a aplicação.

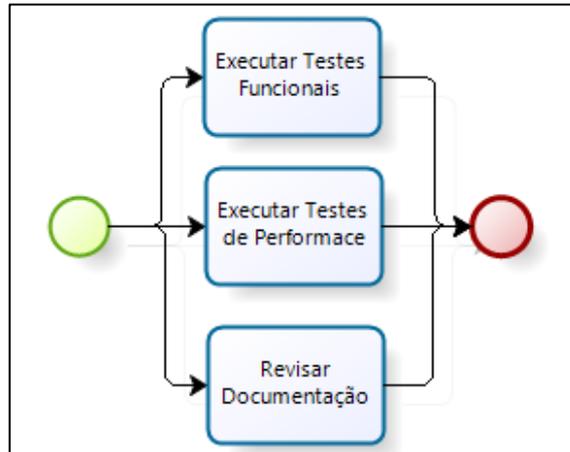


**Figura 6 - Construção do software e help**

O sub-processo de construção de software (Figura 6), consiste no desenvolvimento propriamente dito, com base na documentação gerada na fase anterior e nos padrões de implementação estabelecidos pela empresa, além da aplicação dos testes unitários.

A documentação por sua vez, consiste na elaboração da ajuda (*help*) ao usuário, disponível em cada interface e da documentação do processo de negócio implementado, ambas também com base na documentação disponibilizada na fase anterior.

O sub-processo de validação, representado na Figura 7, consiste na execução dos casos de testes, previamente elaborados e de testes voltados a performance da aplicação, que também foram construídos na fase anterior. Além disso, deve ser inspecionada a documentação gerada.



**Figura 7 - Validar a implementação**

### **3.1.4. Liberação**

A liberação do projeto se inicia com uma apresentação da aplicação implementada, às equipes envolvidas, desde as primeiras etapas do projeto. À estes participantes será exibido o funcionamento das implementações e com isso será finalizada a homologação do projeto.

Em seguida serão previstas as atividades de treinamento, tanto das funcionalidades, quanto da estrutura técnica desenvolvidas. Habilitando assim as áreas interessadas à implantação e manutenção.

Por fim é feita a liberação do projeto propriamente dita, ficando este disponível para a atualização pelos clientes. Encerrando assim o projeto, validando seus pontos positivos e negativos e levantando as lições aprendidas, afim de tomar ações para melhoria em novos projetos.

### **3.1.5. Considerações Finais**

Devido a necessidade da agilidade na estimativa nota-se que esta é realizada com base em pouca documentação, em uma etapa muito inicial do processo. Esta característica pode distorcer algumas evidências do projeto que somente serão percebidas mais tarde, talvez até quando o cronograma de desenvolvimento já estiver atrasado.

A aplicação da métrica da forma utilizada pela empresa pode ser questionada quanto a forma de atribuição de complexidade dos casos de uso, segundo Anselmo (2010) que exemplifica um processo de estimativa por UCP tomando como base o diagrama de casos de

uso e mas também os casos de uso descritivos.

Embasada somente no diagrama de casos de uso não há como estabelecer esta complexidade, pois este diagrama não contempla a descrição de nenhuma das formas (número de transações ou entidades) de calcular tal informação.

Outro aspecto que chama atenção é o fato da utilização do diagrama de casos de uso na estimativa, porém, a aplicação do documento de especificação geral no restante do processo, que conforme pode ser observado no anexo 1, não é estruturado de forma a dar sequencia na descrição dos casos de uso levantados no diagrama.

## 4. USO DE UMA MÉTRICA NO PROCESSO DE DESENVOLVIMENTO

Segundo Vazquez (2008), baseado no PMI (*Project Management Institute*) podem ser considerados três passos básicos na gerência de projetos, são eles:

- Planejamento: onde são definidos os objetivos e selecionado o melhor curso de execução;
- Execução: onde deve-se executar o plano definido, coordenando pessoas e demais recursos;
- Controle: onde deve ser feito o monitoramento e medição do progresso, tomando ações corretivas conforme necessário.

Ainda segundo Vazquez (2008), esta abordagem remete à definição de visibilidade, ou seja, o saber onde se quer chegar e como antes de iniciar a caminhada. As métricas de software deste modo, são objetos dos processos de planejamento e controle do projeto.

Com base neste enfoque a seguir serão avaliadas dentre as métricas abordadas, qual a indicada para o processo de desenvolvimento da empresa e, com isso, poderão ser identificadas as adaptações necessárias para sua aplicação.

### 4.1. Métrica a ser Aplicada

Para realizar a escolha da métrica que será aplicada no processo de desenvolvimento alvo, descrito no capítulo 3, será utilizado como base um comparativo construído por Anselmo (2010). Este abordava inicialmente análise de pontos de função e pontos por caso de uso, por isso foi adaptado para contemplar o modelo COCOMO II. Além disso, foram adicionados critérios relevantes para a identificação das características de cada métrica, conforme pode ser observado na Tabela 18.

**Tabela 18 - Comparativo métricas**

<b>Critério</b>	<b>PFA</b>	<b>UCP</b>	<b>COCOMO II</b>
<b>Ambiente em que foi baseada</b>	Análise estruturada de projetos.	Orientação a Objetos utilizando modelagem UML.	Modelo paramétrico, absorve conceitos de componentização, reutilização de código.
<b>Utilização</b>	Métrica mais antiga e utilizada no mundo.	Métrica relativamente nova, sua utilização vem aumentando.	Métrica mais antiga e utilizada no mundo.
<b>Padronização</b>	Padrão internacional.	Ainda não alcançou o nível de padronização.	Não possui um órgão de padronização, mas possui manual publicado e mantido pela USC.
<b>Notação Padrão</b>	Modelo funcional.	Modelo de casos de uso.	Modelo funcional (quando aplicado utilizando FPA para medição de tamanho).
<b>Regras e Padrões de Contagem</b>	Possui regras de contagem padronizadas.	Não existem padrões para descrever casos de uso.	Possui manual de contagem.
<b>Literatura Disponível</b>	Largamente discutida na literatura.	Tem aumentado a publicação de estudos na literatura.	Largamente discutida na literatura.
<b>Informações Históricas Disponíveis</b>	É suportada pelo IFPUG e possui bases históricas de contagens realizadas.	Ainda não possui históricos de produtividade.	Possui históricos de aplicação publicados e mantidos pela USC.
<b>Complexidade</b>	Necessidade de conhecimento detalhado dos requisitos do projeto e do modelo de cálculo.	Necessidade de conhecimento dos casos de uso envolvidos.	Utiliza equações matemáticas, calibragem de parametrização, além da contagem de tamanho como pré-requisito.
<b>Quando pode ser Aplicada</b>	Final das fases de análise e projeto necessário conhecimento prévio do modelo de dados.	Fase inicial do projeto e quando houver alterações de requisitos significativas, que modifiquem ou adicionem casos de uso.	Antes que a arquitetura completa do sistema tenha sido projetada ( <i>Early Design</i> ), após a definição completa da arquitetura do sistema ( <i>Post-Architecture</i> ) e durante os primeiros estágios da engenharia de software ( <i>Application Composition</i> ).

<b>Maturidade</b>	Alto nível de maturidade.	Em fase de amadurecimento.	Alto nível de maturidade.
<b>Subjetividade</b>	É subjetiva e possui diferença entre contadores.	É subjetiva e possui diferença entre contadores.	Pode ser considerada subjetiva se utilizar APF para medição de tamanho, no entanto, mesmo utilizando KLOC (Milhares de linhas de código) ou Contagem de Pontos de Objeto, poderá haver variação quanto ao modo de construção de software.
<b>Treinamento e Certificação</b>	Oferece treinamento e certificação.	Ainda não oferece treinamentos e certificação.	Não oferece certificação. Não foram identificados órgãos que ofereçam treinamento no Brasil.
<b>Softwares Disponíveis</b>	Vasta quantidade de softwares disponíveis, desde planilhas Excel a aplicações WEB.	Quantidade de softwares disponível crescente, já sendo possível encontrar o modelo incorporado a softwares de modelagem UML.	Vasta quantidade de softwares disponíveis, desde planilhas Excel a aplicações WEB.

Conforme pode ser observado durante o levantamento do processo de desenvolvimento, a estimativa do projeto é aplicada no início do processo, antes da análise detalhada dos requisitos. Este ponto torna a implementação das métricas APF e COCOMO II difícil, pois as mesmas precisam ser aplicadas após o detalhamento dos requisitos.

Utilização da APF ou COCOMO II embora não requeiram mudanças com a relação aos artefatos produzidos pelo processo de desenvolvimento, por não serem ligados a nenhuma notação padrão, necessitam de treinamentos e especialização no modelo de contagem. Isto porque são utilizadas terminologias próprias da métrica que não são familiares a equipe além da notável complexidade.

Além disso, no caso da especificação funcional, como pode ser observado no documento apresentado do processo de desenvolvimento, não são contemplados os detalhes necessários para a aplicação da medição de tamanho por pontos de função.

Já o UCP pode ser aplicado, segundo Anselmo (2010) com base no diagrama e descrição dos casos de uso, sendo esta comparada com uma nova medição a ser realizada com o término das fases de análise e projeto.

Observando a tendência da empresa em adotar a modelagem UML no processo de desenvolvimento, percebe-se que a escolha da métrica de pontos por caso de uso é a melhor alternativa para a empresa. A familiaridade da equipe com os conceitos da métrica, já que utilizam o diagrama de casos de uso, e com isso, a diminuição dos investimentos em treinamentos para a sua compreensão, também contam na sua indicação.

Apesar deste modelo não ser considerado consolidado como os demais, o próprio crescimento do método no cenário nacional, conforme indicado pela pesquisa apresentada no capítulo 2, é um fator que vem a agregar na sua escolha.

Outro ponto a ser observado é que embora hajam críticas severas com relação a esta abordagem, devido não haver um padrão regulamentado para a escrita de casos de uso, as demais métricas também podem ser consideradas subjetivas e apresentam variações entre as estimativas.

Por isso, acredita-se que aplicação de UCP para a estimativa no processo alvo é a mais coerente. Além disso, poderão ser observados durante o capítulo seguinte (onde serão sugeridas adaptações no processo) os pontos falhos já conhecidos da métrica, assim realizando consistências para reduzir o risco de falhas por estes problemas.

#### **4.2. Adaptações no Processo de Desenvolvimento**

As adaptações para o uso da métrica sugeridas para o processo de desenvolvimento alteram basicamente três sub-processos, são eles: entender o escopo, estimar o tempo do projeto e elaborar a análise detalhada, conforme pode ser observado em destaque na Figura 8.

O objetivo é poder aplicar a estimativa por UCP priorizando sua eficácia. Visando assim tornar a estimativa mais confiável e proporcionar, na medida em que aumente a base de conhecimento, a melhoria contínua do processo. Para isso são sugeridos três pontos de medição, em diferentes etapas do processo, são eles:

- Estimativa inicial: após o processo de entendimento do escopo do projeto;
- Primeira revisão da estimativa: durante a elaboração da análise detalhada;
- Segunda revisão da estimativa: ao final do processo de análise detalhada.

A seguir serão detalhados os processos conforme as adaptações sugeridas visando a aplicação dos três pontos de medição especificados acima.

Conforme observado durante o levantamento do processo, a empresa realiza hoje a

estimativa no início do projeto com base somente no diagrama de casos de uso construído, não sendo utilizado o modelo de casos de uso descritivo para auxiliar no entendimento.

É importante salientar que a estimativa é feita neste passo do processo, conforme pode ser observado na figura 8, pois a determinação da estimativa do tempo de um projeto é um fator decisivo para a aprovação ou não de sua implementação.

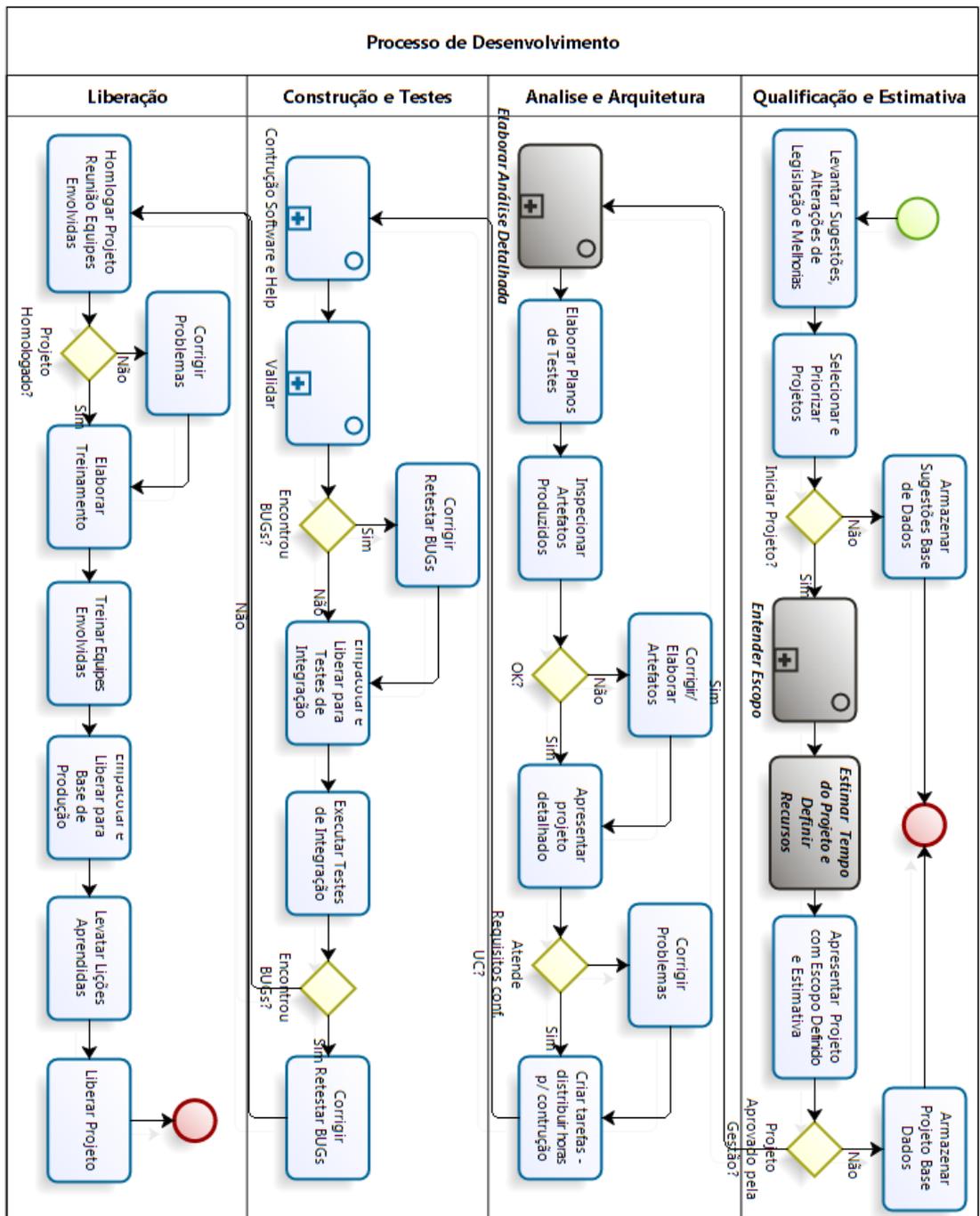
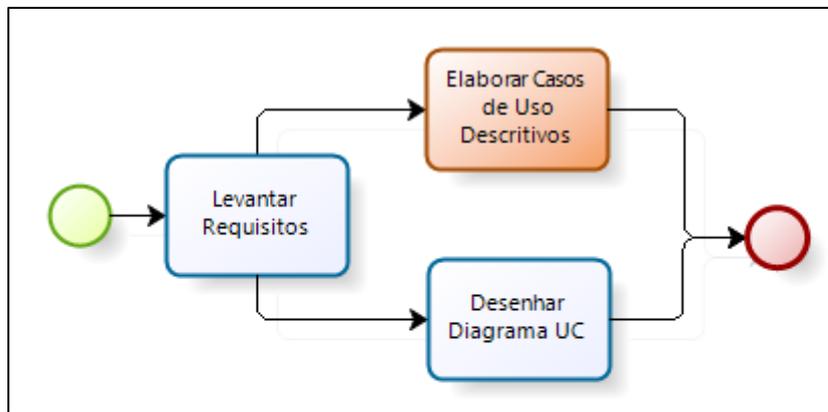


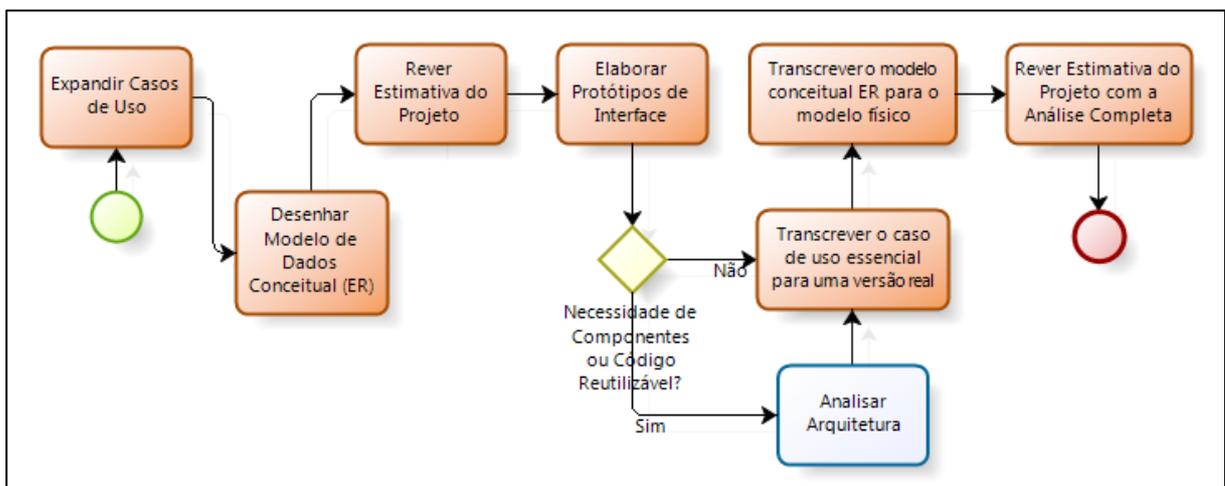
Figura 8 - Processo de desenvolvimento adaptado

A primeira adaptação sugerida quanto ao processo é a adoção do modelo descritivo de casos de uso, já na fase de levantamento do escopo do projeto, conforme modelado na figura 9, trazendo mais informações e padronização para a estimativa que será feita no processo seguinte.



**Figura 9 - Adaptação: Sub-processo entender o escopo**

Na etapa da análise detalhada, conforme pode ser observado na Figura 10, sugere-se a troca do atual documento de especificação geral pelos casos de uso expandidos onde serão detalhadas as descrições dos casos de uso, feitas em alto nível na etapa anterior. Assim, será mantida uma sequência na análise inicial feita, sem o risco de adicionar funcionalidades fora do contexto dos casos de uso descritos.



**Figura 10 - Adaptação: Sub-processo elaborar análise detalhada**

Segundo Wazlawick (2004) a expansão dos casos de uso corresponde ao aprofundamento da análise de requisitos, para isso deve ser feito um exame detalhado dos processo de negócio, descrevendo passo-a-passo como o caso de uso ocorre. Essa estrutura é

representada pelos fluxos principal e alternativos que fazem parte da estrutura do caso de uso expandido.

Para auxiliar na expansão dos casos de uso sugere-se o *template* utilizado pelo *Rational Unified Process* (RUP)<sup>2</sup> citado por Cockburn (2005) como um modelo atraente e fácil de utilizar.

Este modelo pode ser observado no anexo 3 e possui uma estrutura que contempla, a descrição dos casos de uso e seus atores, os fluxos de eventos básico e alternativo, os requisitos especiais e as pré-condições e pós-condições do caso de uso.

Sugere-se também a utilização do modelo conceitual para representar a estrutura de dados através do diagrama ER, antes da representação da estrutura física do banco de dados. Este modelo, segundo Heuser (2009), deve representar as necessidades do projeto em termos de armazenamento de dados independente do banco de dados que será utilizado.

Com estes artefatos produzidos pretende-se refazer a estimativa do projeto afim conferir a estimativa inicial, agora com mais informações sobre o projeto. Assim, pode-se ajustar as possíveis variações encontradas no cronograma estabelecido, antes do início do desenvolvimento do projeto.

Com a elaboração dos protótipos de interface e acoplando-os aos fluxos descrevendo as interações nas janelas e componentes, serão transcritos os casos de uso expandidos para os casos de uso reais. Assim, será possível ter as mesmas informações que antes constavam no documento de especificação geral (nome do documento de análise utilizado atualmente), agora utilizando os modelos UML.

Por fim, o modelo ER conceitual construído será transcrito no modelo lógico do banco de dados. Com isso, será finalizada a análise detalhada, mantendo as mesmas informações do processo original, porém, utilizando um formato diferente, assim antecipando a intensão da empresa em utilizar mais artefatos da modelagem UML.

Após a construção destes artefatos será finalizado o sub-processo de análise detalhada. Neste momento do processo sugere-se uma nova revisão da estimativa com o objetivo de obter uma avaliação com base em todos os artefatos produzidos. Esta será a estimativa final do projeto.

Além disso, as estimativas feitas nos três momentos poderão ser comparadas e assim poderá ser avaliada a variação entre as mesmas, inclusive no histórico de projetos. Será possível então, apontar os riscos envolvidos na estimativa inicial, que é feita com base em

---

<sup>2</sup> O RUP é um iterativo processo de desenvolvimento de software criado pela *Rational Software Corporation*.

menos informações, com o tempo estabelecendo um percentual de variação da estimativa inicial e final que normalmente ocorre nos projetos.

Sugere-se também a comparação da estimativa final de análise com as horas reais do projeto, após o término do mesmo. Será possível então, verificar a variação das horas estimadas e realizadas, podendo com isso serem tomadas ações de melhoria com relação ao processo de estimativa.

Com a implementação destas adaptações no processo de desenvolvimento a estimativa de pontos por caso de uso poderá ser aplicada, obtendo todas as informações necessárias para a estimativa através dos artefatos produzidos durante o sub-processo de análise de detalhada. Assim, os projetos poderão ser estimados de acordo com tal método sem que seja necessário produzir artefatos ou documentações auxiliares para isso.

## **5. FERRAMENTA DE ESTIMATIVA**

A ferramenta de estimativa será construída com base nas características da métrica escolhida e no processo de desenvolvimento mapeado, considerando as adaptações sugeridas.

Para a modelagem do software será utilizada a linguagem UML na versão 2.3, com os diagramas necessários para representar as características funcionais, estáticas e comportamentais do sistema, conforme Wazlawick (2004), para isto serão construídos os seguintes diagramas:

- Diagrama de Casos de Uso;
- Diagrama de Classes Conceitual;
- Diagrama de Sequência;
- Diagrama de Classes de Projetos.

Será utilizado também o modelo de casos de uso descritivo sugerido como aplicação no processo adaptado da empresa, no capítulo anterior deste trabalho. Este é baseado em um dos modelos propostos por Cockburn (2005), conforme já citado anteriormente. Este modelo junto com a elaboração dos protótipos de interface serão utilizados também na fase de análise.

A seguir serão apresentadas todas as fases da construção do protótipo da ferramenta de estimativa. Estas fases estarão divididas em: análise e projeto, representando a modelagem da ferramenta juntamente com os artefatos construídos; implementação e testes, onde será apresentado como foi elaborada a ferramenta com base na modelagem; requisitos para instalação, onde serão apresentadas as configurações necessárias para a utilização da ferramenta.

### **5.1. Fase de Análise**

O sistema será aplicado no processo de desenvolvimento estudado, com isso deverá ser aderente ao mesmo. Para isso deverá atender aos requisitos relacionados a seguir.

#### **a. Importar Informações dos Casos de Uso Descritivos**

Importar os casos de uso descritivos identificando atores e os passos dos fluxos (cadastrados via outro sistema). Com isso deverá ser possível calcular automaticamente a complexidade dos casos de uso com base na quantidade de transações (passos dos fluxos).

#### **b. Cadastrar e Configurar Projetos com Revisões de Estimativa**

Deve ser possível a cada projeto determinar a quantidade de revisões na estimativa que serão utilizadas nas diferentes fases do desenvolvimento. Além disso, deverá ser possível informar a forma de cálculo da complexidade dos casos de uso, se será utilizando as transações importadas dos casos de uso descritivos ou com base nas entidades envolvidas no processo. Poderão ser utilizadas as duas formas de estimar complexidade em um mesmo projeto (conforme sua configuração), definindo uma oficial para o cálculo após a análise das informações.

#### **c. Estimar o Projeto Utilizando UCP**

Será feita a estimativa e revisão de estimativa do projeto utilizando os conceitos apresentados neste trabalho de UCP. Para isso deverá ser considerada a atribuição da complexidade dos atores e casos de uso, definição dos fatores de ajuste, cálculo dos casos de uso não ajustados, cálculo dos casos de uso ajustados e tempo do projeto.

#### **d. Comparar Informações**

Deverá ser possível comparar as informações de diferentes estimativas do projeto.

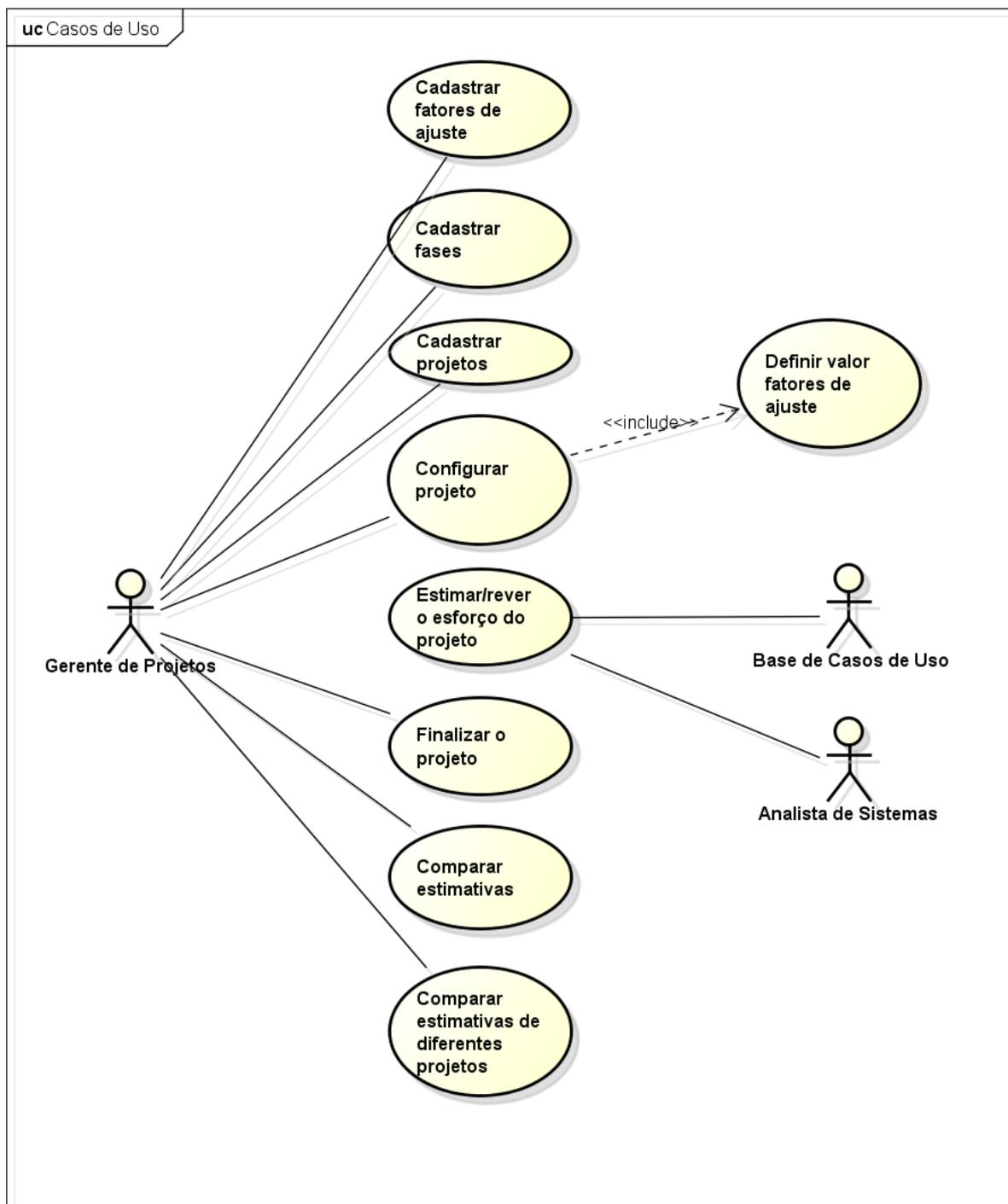
#### **e. Considerações Referentes a Base de Casos de Uso Descritivo**

Atualmente o ambiente de desenvolvimento estudado não possui uma ferramenta para o cadastro dos casos de uso descritivos, com isso, será proposta a importação dos dados conforme o formato modelado, porém, não será feita neste momento a modelagem e construção desta ferramenta. Esta poderá ser construída posteriormente obedecendo o padrão de transferência de dados estabelecido.

### **5.1.1. Diagrama de Casos de Uso**

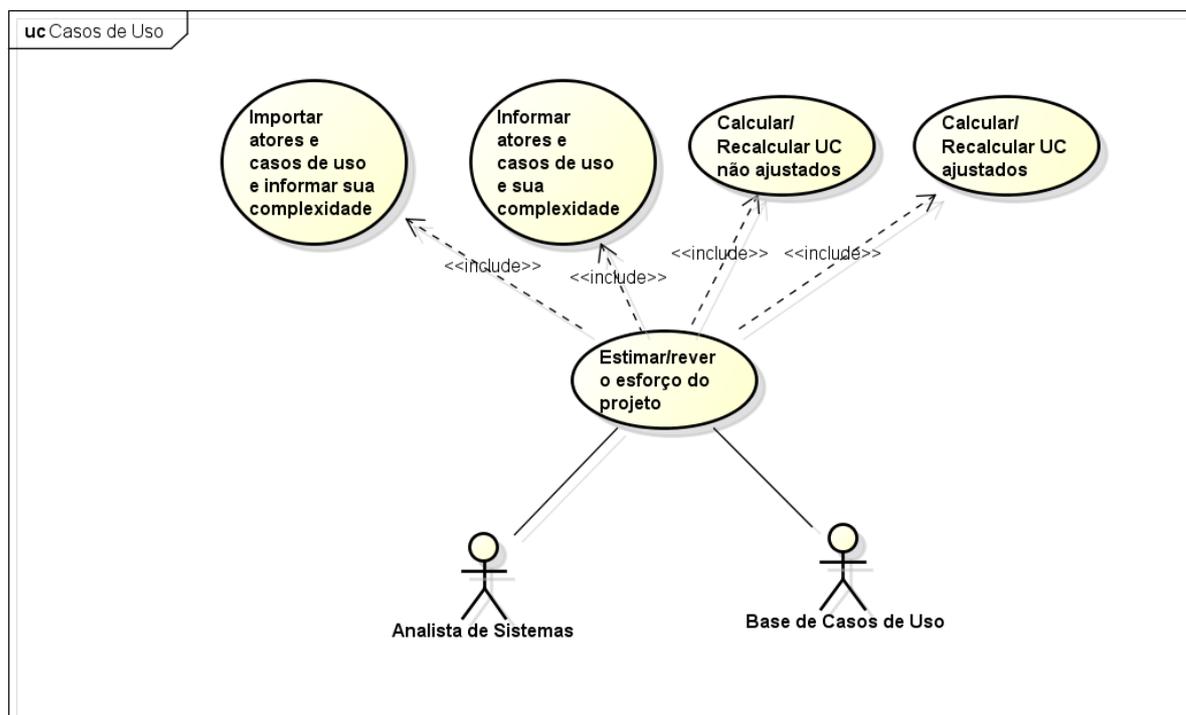
Segundo Larman (2007) o diagrama de casos de uso objetiva ilustrar os casos de uso e atores, assim como o relacionamento entre eles, ou seja, é uma ferramenta de comunicação que resume o comportamento de sistema e seus atores.

A figura a seguir ilustra o diagrama de casos de uso representando a comunicação entre os atores e casos de uso da ferramenta que será construída.



**Figura 11 - Diagrama de casos de uso**

Para melhor compreensão do caso de uso “Estimar/rever o esforço do projeto”, este foi expandido conforme apresentado na figura a seguir.



**Figura 12 - Caso de uso estimar/rever esforço do projeto expandido**

### 5.1.2. Descrição dos Casos de Uso

O modelo de casos de uso segundo Bezerra (2003) é uma representação das funcionalidades do sistema. Este é um modelo de fundamental importância na medida que direciona as demais fases posteriores. Este modelo é composto por casos de uso, atores e da representação do relacionamento entre os mesmos.

A seguir serão descritas, utilizando o modelo de casos de uso sugerido, as funcionalidades da ferramenta que será implementada.

#### a. Cadastrar Fatores de Ajuste

- |  |
|--|
| <p>1. Nome do Caso de uso: Cadastrar Fatores de Ajuste</p> <p>1.1. Descrição Resumida<br/>Este caso de uso permite cadastrar, alterar e excluir os fatores de ajuste considerados no ambiente de desenvolvimento.</p> <p>1.2. Atores</p> <p>1.2.1. Gerente de Projetos.</p>            |
| <p>2. Fluxos de Eventos</p> <p>2.1. Fluxo Básico</p> <p>2.1.1. O Gerente de Projetos seleciona a opção de cadastrar fatores de ajuste;</p> <p>2.1.2. Informa o número, a descrição e peso do fator de ajuste e salva.</p> <p>2.2. Fluxos Alternativos – Edição do fator de ajuste:</p> |

<p>2.2.1. O Gerente de Projetos seleciona o fator de ajuste que será modificado;</p> <p>2.2.2. Edita as informações referentes ao número, descrição e peso do fator de ajuste.</p> <p>2.3. Fluxos Alternativos – Exclusão do fator de ajuste:</p> <p>2.3.1. O Gerente de Projetos seleciona o fator de ajuste que será excluído;</p> <p>2.3.2. O gerente de projetos exclui o fator de ajuste selecionado.</p>
3. Requisitos Especiais
4. Pré-condições
5. Pós-condições
5.1. Fatores de ajuste cadastrados disponíveis para a estimativa do projeto.

### **b. Cadastrar Fases**

<p>6. Nome do Caso de uso: Cadastrar Fases</p> <p>6.1. Descrição Resumida Este caso de uso permite cadastrar, alterar e excluir as fases e informar em quais destas fases serão realizadas as estimativas de tempo do projeto.</p> <p>6.2. Atores 6.2.1. Gerente de Projetos.</p>
<p>7. Fluxos de Eventos</p> <p>7.1. Fluxo Básico 7.1.1. O Gerente de Projetos seleciona a opção de cadastrar as fases do projeto; 7.1.2. Informa a sequência o nome da fase, a descrição e se será realizada a estimativa nesta fase do projeto e salva.</p> <p>7.2. Fluxos Alternativos – Edição da Fase: 7.2.1. O Gerente de Projetos seleciona a fase que será modificada; 7.2.2. Edita as informações referentes ao nome, descrição, estimativa da fase.</p> <p>7.3. Fluxos Alternativos – Exclusão da Fase: 7.3.1. O Gerente de Projetos seleciona a fase que será excluída; 7.3.2. O Gerente de Projetos exclui a fase selecionada.</p>
<p>8. Requisitos Especiais 8.1. Não permitir excluir as fases dos projetos se existirem estimativas de projetos utilizando a mesma.</p>
9. Pré-condições
<p>10. Pós-condições 10.1. Fases do projeto cadastradas disponíveis para a estimativa do projeto.</p>

### **c. Cadastrar Projetos**

<p>11. Nome do Caso de uso: Cadastrar Projetos</p> <p>11.1. Descrição Resumida Este caso de uso permite o cadastro de projetos de desenvolvimento para realizar a medição de esforço do mesmo. Também deve permitir editar e excluir projetos cadastrados.</p> <p>11.2. Atores 11.2.1. Gerente de Projetos.</p>
---

<p>12. Fluxos de Eventos</p> <p>12.1. Fluxo Básico</p> <p>12.1.1. O Gerente de Projetos seleciona a opção de criar um novo projeto;</p> <p>12.1.2. Informa o nome do projeto, a descrição, data de início, o valor de pessoas/hora por UCP, se este irá utilizar importação de casos de uso, como será feito o calculo da complexidade dos casos de uso e salva o projeto incluído.</p> <p>12.2. Fluxos Alternativos – Edição do Projeto</p> <p>12.2.1. O Gerente de Projetos seleciona o projeto que será editado;</p> <p>12.2.2. Edita as informações referentes ao nome, descrição, data de início do projeto, o valor de pessoas/hora por UCP, se este irá utilizar importação de casos de uso e como será feito o calculo da complexidade dos casos de uso.</p> <p>12.3. Fluxos Alternativos – Exclusão do Projeto</p> <p>12.3.1. O Gerente de Projetos seleciona o projeto que será excluído;</p> <p>12.3.2. O Gerente de Projetos exclui o projeto selecionado.</p>
<p>13. Requisitos Especiais</p> <p>13.1. Não permitir alterar informações de projetos já finalizados.</p>
<p>14. Pré-condições</p>
<p>15. Pós-condições</p> <p>15.1. Projeto cadastrado disponível para configuração ou estimativa.</p>

#### **d. Configurar Projeto**

<p>1. Nome do Caso de uso: Configurar Projeto</p> <p>1.1. Descrição Resumida</p> <p>Este caso de uso permite configurar o peso dos casos de uso por entidade e/ou transações (conforme definição do projeto) e dos atores. Além disso neste momento deverá ser atribuído o valor dos fatores de ajuste para o projeto.</p> <p>1.2. Atores</p> <p>1.2.1. Gerente de Projetos.</p>
<p>2. Fluxos de Eventos</p> <p>2.1. Fluxo Básico</p> <p>2.1.1. O Gerente de Projetos seleciona o projeto que irá configurar;</p> <p>2.1.2. O Gerente de Projetos configura a complexidade e o peso dos casos de uso por transação para o projeto;</p> <p>2.1.3. O Gerente de Projetos configura o peso dos atores para o projeto;</p> <p>2.1.4. O Gerente de Projetos configura o valor dos fatores de ajuste para o projeto.</p> <p>2.2. Fluxos Alternativos</p> <p>2.2.1. No passo 2 o projeto está configurado para utilizar como base de complexidade dos casos de uso entidade.</p> <p>2.2.2. O gerente configura a complexidade e o peso dos casos de uso por entidade e retorna ao fluxo principal no passo 3.</p>
<p>3. Requisitos Especiais</p> <p>3.1. Não permitir alterar informações de projetos já finalizados.</p> <p>3.2. Carregar como padrão as informações de configuração de casos de uso, atores informadas no último projeto realizado, e dos fatores de ajuste cadastrados.</p>
<p>4. Pré-condições</p>

4.1. Projeto cadastrado.
5. Pós-condições
5.1. Projeto configurado disponível para estimativa.

#### **e. Importar Atores e Casos de Uso e Informar a sua Complexidade**

<p>1. Nome do Caso de uso: Importar Atores e Casos de Uso e Informar sua Complexidade</p> <p>1.1. Descrição Resumida</p> <p>Este caso de uso permite o Analista de Sistemas importar os atores e casos de usos informados nos modelos descritivos de casos de uso. Para os atores deverá ser definida manualmente a complexidade. Para os casos de uso na importação de dados será calculada automaticamente a complexidade conforme as transações informadas nos fluxos dos casos de uso descritivos.</p> <p>1.2. Atores</p> <p>1.2.1. Analista de Sistema;</p> <p>1.2.2. Base de Casos de Uso;</p>
<p>2. Fluxos de Eventos</p> <p>2.1. Fluxo Básico</p> <p>2.1.1. O Analista de Sistemas seleciona na Base de Casos de Uso os casos de uso descritivos que deseja importar e o Sistema de Estimativas cadastra os atores e casos de uso no projeto;</p> <p>2.1.2. O Sistema de Estimativas calcula a complexidade dos casos com base nos fluxos dos casos de uso descritivos;</p> <p>2.1.3. O Analista de Sistemas informa a complexidade dos atores e salva os casos de uso e atores cadastrados.</p> <p>2.2. Fluxos Alternativos</p> <p>2.2.1. Cadastro de mais de duas formas de atribuição de complexidade para casos de uso:</p> <p>2.2.1.1. No passo 2 o projeto foi configurado para utilizar as duas formas de estimativa, o Analista de Sistemas informa a complexidade dos casos de uso de acordo com as entidades envolvidas no processo do caso de uso, continua o fluxo no passo 3.</p> <p>2.2.2. Alteração da complexidade dos casos de uso e atores:</p> <p>2.2.2.1. No passo 1 o Analista de Sistema altera a complexidade dos atores e casos de uso que deseja e salva as alterações realizadas</p>
<p>3. Requisitos Especiais</p> <p>3.1. Quando existirem estimativas posteriores validar a consistência de informações da base e da última estimativa, possibilitando a importação dos dados da última estimativa, caso o usuário opte por manter os dados iguais.</p>
<p>4. Pré-condições</p> <p>4.1. Projeto cadastrado e configurado.</p>
<p>5. Pós-condições</p> <p>5.1. Atores e casos de uso cadastrados e com complexidade definida.</p>

#### **f. Informar Atores e Casos de Uso e sua Complexidade**

<p>1. Nome do Caso de uso: Informar Atores e Casos de Uso e sua Complexidade</p> <p>1.1. Descrição Resumida</p> <p>Este caso de uso permite o Analista de Sistemas informar os atores e casos de</p>
--

<p>usos manualmente. Após informar novos atores e casos de uso, podendo isto ser feito após o processo de importação, deverá ser definido a complexidade dos atores e casos de uso. Caso na configuração do projeto ambas as formas de atribuição de complexidade para o caso de uso, esta deverá ser informada com base nas entidades envolvidas e transações. Poderão ser utilizadas as duas formas de estimar complexidade em um mesmo projeto (conforme sua configuração), definindo uma delas para ser utilizada no cálculo após a análise das informações.</p> <p>1.2. Atores</p> <p>1.2.1. Analista de Sistema;</p>
<p>2. Fluxos de Eventos</p> <p>2.1. Fluxo Básico</p> <p>2.1.1. O Analista de Sistemas cadastra os novos atores e casos de uso;</p> <p>2.1.2. O Analista de Sistemas informa a complexidade dos atores e casos de uso incluídos de com a forma configurada no projeto e salva os casos de uso e atores cadastrados.</p> <p>2.2. Fluxos Alternativos</p> <p>2.2.1. Cadastro de mais de duas formas de atribuição de complexidade para casos de uso:</p> <p>2.2.1.1. No passo 2 o projeto foi configurado para utilizar as duas formas de estimativa, o Analista de Sistemas informa a complexidade dos atores de acordo com a segunda forma definida.</p> <p>2.2.2. Alteração da complexidade dos casos de uso e atores:</p> <p>2.2.2.1. No passo 1 o Analista de Sistema altera a complexidade dos atores e casos de uso que deseja e salva as alterações realizadas.</p>
<p>3. Requisitos Especiais</p> <p>3.1. Quando existirem estimativas posteriores copiar os dados da última estimativa.</p>
<p>4. Pré-condições</p> <p>4.1. Projeto cadastrado e configurado.</p>
<p>5. Pós-condições</p> <p>5.1. Atores e casos de uso cadastrados e com complexidade definida.</p>

#### **g. Calcular/Recalcular UC não Ajustados**

<p>1. Nome do Caso de uso: Calcular/Recalcular UC não Ajustados</p> <p>1.1. Descrição Resumida</p> <p>Este caso de uso permite o Analista de Sistemas calcular os UC não ajustados de acordo a complexidade atribuída aos casos de uso e atores cadastrados no projeto.</p> <p>1.2. Atores</p> <p>1.2.1. Analista de Sistemas;</p>
<p>2. Fluxos de Eventos</p> <p>2.1. Fluxo Básico</p> <p>2.1.1. O Analista de Sistemas seleciona o cálculo/recálculo dos UC não ajustados;</p> <p>2.1.2. O Sistema de Estimativa calcula o peso dos atores (UAW) através da soma dos produtos do número de atores de cada tipo pelo respectivo peso (conforme Tabela 7 que pode ser visualizada na página 27);</p> <p>2.1.3. O Sistema de Estimativa calcula o peso dos casos de uso (UUCW) através da soma dos produtos do número de casos de uso de cada tipo pelo</p>

<p>respectivo peso (conforme tabelas 8 e 9 que podem ser visualizadas na página 28);</p> <p>2.1.4. O Sistema de Estimativa calcula o valor dos UC não ajustados através da fórmula <math>UUCP = UAW + UUCW</math>;</p> <p>2.1.5. O Analista de Sistemas verifica e salva o valor calculado.</p> <p>2.2. Fluxos Alternativos</p>
3. Requisitos Especiais
4. Pré-condições
4.1. Atores e casos de uso cadastrados e com complexidade definida.
5. Pós-condições
5.1. UC não ajustados calculados.

#### **h. Calcular/Recalcular UC Ajustados**

<p>1. Nome do Caso de uso: Calcular/Recalcular UC Ajustados</p> <p>1.1. Descrição Resumida</p> <p>Este caso de uso permite o Analista de Sistemas calcular os UC ajustados de com base no cálculo de UC não ajustados e na complexidade dos fatores de ajuste.</p> <p>1.2. Atores</p> <p>1.2.1. Analista de Sistemas;</p>
<p>2. Fluxos de Eventos</p> <p>2.1. Fluxo Básico</p> <p>2.1.1. O Analista de Sistemas seleciona o cálculo/recálculo dos UC ajustados;</p> <p>2.1.2. O Sistema de Estimativa calcula o valor dos fatores de complexidade técnica (TCF) através da fórmula <math>TCF = 0.6 + (0.01 \times (TFactor))</math>, onde TFactor é o somatório dos produtos do valor dos fatores pelo seu respectivo peso (conforme Tabela 10 que pode ser visualizada na página 29);</p> <p>2.1.3. O Sistema de Estimativa calcula o valor dos fatores de complexidade ambiental (ECF) através da fórmula <math>ECF = 1.4 + (-0.03 \times (EFactor))</math>, onde TFactor é o somatório dos produtos do valor dos fatores pelo seu respectivo peso (conforme Tabela 11 que pode ser visualizada na página 30);</p> <p>2.1.4. O Sistema de Estimativa calcula o valor dos casos de uso ajustados multiplicando o valor dos UUCP pelos valores dos fatores de ajuste ambientais e técnicos, conforme a fórmula <math>UCP = UUCP \times TCF \times ECF</math>;</p> <p>2.1.5. O Sistema de Estimativa calcula o tempo do projeto multiplicando o valor dos UCP pelo valor de pessoas/hora por UCP do projeto;</p> <p>2.2. Fluxos Alternativos</p>
3. Requisitos Especiais
4. Pré-condições
4.1. UC não ajustados calculados e complexidade dos fatores técnicos e ambientais definida.
5. Pós-condições
5.1. UC ajustados calculados.

## **i. Estimar/Rever Esforço do Projeto**

<p>1. Nome do Caso de uso: Estimar/Rever Esforço do Projeto</p> <p>1.1. Descrição Resumida</p> <p>1.2. Atores</p> <p>1.2.1. Analista de Sistemas</p> <p>1.2.2. Base de Casos de uso</p>
<p>2. Fluxos de Eventos</p> <p>2.1. Fluxo Básico</p> <p>2.1.1. O Analista de Sistemas seleciona o projeto que irá cadastrar a estimativa;</p> <p>2.1.2. O Sistema de Estimativa verifica a próxima fase do processo que esta configurada para realizar estimativa.</p> <p>2.1.3. O Analista de Sistemas cadastra a nova estimativa informando sua descrição e a data;</p> <p>2.1.4. O Analista de Sistemas executa os casos de uso “Importar atores e casos de uso e definir sua complexidade” e/ou “Informar atores e casos de uso e definir sua complexidade”, conforme configuração do projeto;</p> <p>2.1.5. O Analista de Sistemas executa o caso de uso “Calcular/Recalcular os UC não ajustados”;</p> <p>2.1.6. O Analista de Sistemas executa o caso de uso “Definir o valor dos fatores de ajuste”;</p> <p>2.1.7. O Analista de Sistemas executa o caso de uso “Calcular/Recalcular os UC ajustados”.</p> <p>2.2. Fluxos Alternativos</p> <p>2.2.1. Alteração configuração da estimativa:</p> <p>2.2.1.1. No passo 1 o Analista de Sistemas seleciona o projeto que irá alterar a estimativa;</p> <p>2.2.1.2. No passo 2 o Sistema de Estimativas carrega as fases que possuem estimativas;</p> <p>2.2.1.3. No passo 3 o Analista de Sistemas escolhe a fase que deseja editar a configuração da estimativa, continua o fluxo no passo 4.</p>
<p>3. Requisitos Especiais</p> <p>3.1. Não permitir alterar informações de projetos já finalizados.</p>
<p>4. Pré-condições</p> <p>4.1. Projeto cadastrado e configurado;</p>
<p>5. Pós-condições</p> <p>5.1. Processo de estimar ou rever estimativa completo.</p>

## **j. Finalizar o Projeto**

<p>1. Nome do Caso de uso: Finalizar o Projeto</p> <p>1.1. Descrição Resumida</p> <p>Este caso de uso permite finalizar o projeto de estimativa informando o número de horas reais do projeto.</p> <p>1.2. Atores</p> <p>1.2.1. Gerente de Projetos</p>
<p>2. Fluxos de Eventos</p> <p>2.1. Fluxo Básico</p> <p>2.1.1. O Gerente de Projetos seleciona a opção de finalizar projetos;</p>

<ul style="list-style-type: none"> <li>2.1.2. O Gerente de Projetos seleciona o projeto que irá finalizar;</li> <li>2.1.3. O Gerente de Projetos informa a data final e o número de horas reais do projeto e salva o projeto finalizado.</li> </ul>
2.2. Fluxos Alternativos
3. Requisitos Especiais
4. Pré-condições
4.1. Projeto estimado e revisto conforme configuração.
5. Pós-condições
5.1. Projeto finalizado.

### **k. Comparar Estimativas**

<ul style="list-style-type: none"> <li>1. Nome do Caso de uso: Comparar Estimativas</li> <li>1.1. Descrição Resumida <ul style="list-style-type: none"> <li>Este caso de uso permite visualizar e comparar as estimativas de um projeto e as horas reais.</li> </ul> </li> <li>1.2. Atores <ul style="list-style-type: none"> <li>1.2.1. Gerente de Projetos</li> </ul> </li> </ul>
<ul style="list-style-type: none"> <li>2. Fluxos de Eventos</li> <li>2.1. Fluxo Básico <ul style="list-style-type: none"> <li>2.1.1. O Gerente de Projetos seleciona a opção de comparar estimativas;</li> <li>2.1.2. O Gerente de Projetos seleciona o projeto que irá verificar;</li> <li>2.1.3. O Gerente de Projetos seleciona as fases que irá verificar;</li> <li>2.1.4. São exibidos os dados ref. ao UAW, UUCW, UUCP, TCF, ECF, UCP, tempo estimado, valor de pessoas/hora por UCP e tempo real do projeto.</li> </ul> </li> <li>2.2. Fluxos Alternativos</li> </ul>
<ul style="list-style-type: none"> <li>3. Requisitos Especiais</li> <li>3.1. Permitir consultar os dados das estimativas de cada fase para possibilitar a identificação das divergências entre as estimativas.</li> </ul>
<ul style="list-style-type: none"> <li>4. Pré-condições</li> <li>4.1. Projeto estimado e revisto conforme configuração.</li> </ul>
<ul style="list-style-type: none"> <li>5. Pós-condições</li> <li>5.1. Comparação das estimativas.</li> </ul>

### **5.1.3. Protótipos de Interface**

Para representar a camada de interface do sistema foram desenvolvidos os protótipos de interface referentes aos casos de uso descritos acima. Estes protótipos podem ser observados nos anexos 4 à 12.

### **5.1.4. Diagrama de Classes Conceitual**

O diagrama de classes é, segundo Bezerra (2003), o diagrama UML utilizado para

representar o aspecto estrutural estático de um sistema orientado a objetos, permitindo a compreensão de como o sistema deverá ser estruturado internamente para atender as funcionalidades externas. Deste modo, foi utilizado este diagrama para expressar tal representação da ferramenta que será desenvolvida, conforme observado na anexo 13.

#### **5.1.5. Diagrama de Atividades**

Com o objetivo de apresentar o fluxo entre as atividades do sistema, foi construído a diagrama de atividades da ferramenta de estimativa. Este diagrama, que pode ser observado no anexo 14 do trabalho, representa os passos da execução de uma estimativa e poderá auxiliar não só no desenvolvimento mas também na execução dos testes de sistema.

### **5.2. Fase de Projeto**

Segundo Bezerra (2003) uma vez especificados na fase de análise os aspectos lógicos de negócio independentes da implementação do sistema, na fase de projeto o foco deve estar na forma em que o sistema funcionará para atender aos requisitos, levando agora em consideração os recursos tecnológicos disponíveis.

Para isto, nesta fase serão construídos os diagramas de interação, representando os aspectos comportamentais de projeto, e o diagrama de classes de projeto representando as características estáticas, considerando negócio, arquitetura e tecnologia.

#### **5.2.1. Diagramas de Interação**

Segundo Fowler (2005) os diagramas de interação representam a colaboração entre os objetos para a execução de um cenário. Um dos diagramas de interação mais comuns é o diagrama de sequência.

O diagrama de sequência mostra a interação, troca de mensagens, entre os objetos em uma linha de vida, e é considerado também segundo Fowler um diagrama interessante pelo fato de quase não ser necessário explicar a sua notação.

Por isso, para representar a troca de mensagens entre os objetos em cada caso de uso será utilizado este diagrama. Em casos de uso com interação de poucos objetos foi

representada além da interação dos objetos da camada de negócio, a dos objetos da camada de aplicação e acesso aos dados, conforme pode ser observado no anexo 15 que representa o diagrama de sequência do caso de uso “Cadastrar Fases”.

Já em casos de uso com maior número de objetos e conseqüentemente maior complexidade, para facilitar o entendimento foi representada somente a interação dos objetos da camada de negócio, conforme pode ser observado no anexo 16 que representa o diagrama de sequência do casos de uso “Estimar Esforço do Projeto”.

Neste diagrama podemos observar a troca de mensagens entre os objetos, acionando todos os passos necessários para a obtenção da estimativa do esforço do projeto.

Os demais diagramas de sequência referentes aos principais casos de uso podem ser observados nos anexos de 17 à 20.

### **5.2.2. Diagrama de classes de projeto**

A partir do modelo de classes conceitual, o qual representa o domínio do negócio, foi construído o modelo de classes de projeto considerando as características tecnológicas da ferramenta de estimativa. Esta ferramenta será desenvolvida utilizando o conceito de separação de camadas (apresentação, lógica de negócio e acesso aos dados), e uma linguagem orientada a objetos.

O diagrama de classes de projeto representando esta separação de camadas bem como a comunicação entre as mesmas, pode ser observado no anexo 21.

A partir do digrama de classe foi realizado o modelo relacional de banco de dados representado através do diagrama ER (entidade relacionamento), este pode ser visualizado no anexo 22 deste trabalho.

## **5.3. Implementação**

Para o desenvolvimento da ferramenta de estimativa será utilizada a linguagem de programação C# com o *framework* Microsoft Visual Studio. Aplicação será *desktop* e será desenvolvida aplicando o conceito de três camadas: apresentação, lógica de negócio e acesso aos dados.

A escolha da linguagem e plataforma de desenvolvimento foi feita com base na expectativa de migração de tecnologia da empresa para a mesma. Além disso, já existem

algumas aplicações na empresa desenvolvidas utilizando tal linguagem.

O banco de dados utilizado para a implementação foi o Oracle 11g na versão XE (*Express Edition*), por se tratar de uma versão gratuita e pelo fato de manter o padrão de banco de dados utilizado pela empresa, que atualmente tem todos os seus produtos na versão 11g do banco de dados Oracle.

Para realizar o acesso e integração do banco de dados Oracle com o Visual Studio foi utilizado o *Oracle Data Provider for .NET* incluso no ODAC (*Oracle Data Access Components*), um pacote de componentes para conexão com o banco de dados disponibilizado pela Oracle.

Foi criada na aplicação uma classe de conexão com o Oracle que utiliza o *Oracle Data Provider for .NET*, onde são feitas todas as configurações de conexão com o banco. Deste modo as classes DAO de acesso aos dados utilizam a classe de conexão para realizar as operações no banco de dados.

Para a construção das interfaces da ferramenta foi utilizado um formulário padrão, para com isso manter os mesmos aspectos visuais em todos os formulários da ferramenta. Deste modo cada interface implementa a herança deste padrão.

Com base na modelagem do diagrama de atividades foi realizada a implementação da camada de negócio da aplicação, onde para o detalhamento de cada caso de uso foram utilizados os casos de uso descritivos e os digramas de sequência. Deste modo, foi possível visualizar o fluxo do processo a ser implementado através do diagrama de atividades e as funcionalidades específicas de cada etapa através dos demais artefatos.

#### **5.4. Testes**

Segundo Sommerville (2007) o processo de validação e verificação de um sistema deve demonstrar que este atende as funcionalidades especificadas e as necessidades reais das partes interessadas.

Para realizar esta validação do sistema, levando em consideração a dimensão do software e sua implementação e testes serem executadas por uma só pessoa, foram utilizados testes unitários durante o desenvolvimento e testes de sistema após o término.

A cada componente desenvolvido foram aplicados testes unitários com base nos casos de uso descritivos, visando garantir que estes estavam de acordo com as características e funcionalidades descritas.

Após o término do desenvolvimento para validar as funcionalidades do sistema foi utilizado o diagrama de atividades para estabelecer o fluxo dos testes de integração entre os componentes desenvolvidos.

### **5.5. Requisitos para Instalação**

Para que seja possível a instalação da ferramenta é necessário possuir um servidor com no mínimo 2 GB de memória e 100 MB de disco rígido, além disso deve possuir sistema operacional Windows XP ou superior e o banco de dados Oracle 11g instalado.

Atendidos estes pré-requisitos é necessário executar um script para a criação do esquema de dados e em seguida copiar o arquivo “EstimativaUCP.exe”.

## **6. SIMULAÇÃO DE UTILIZAÇÃO DA FERRAMENTA**

Para aplicar o modelo de estimativa sugerido com a ferramenta construída será utilizado um projeto já concluído da empresa que foi o cenário alvo do trabalho. Para esta simulação serão apresentados os objetivos e requisitos deste projeto, a forma com que foi estimado e forma com que foi especificado.

Em seguida será aplicado o modelo de estimativa proposto. Esta simulação irá considerar somente uma das estimativas sugeridas para o processo de desenvolvimento, ou seja, não serão aplicadas as revisões de estimativa. Será utilizado como base para estimativa os casos de uso descritivos, que serão construídos.

A estimativa será feita desta forma pois, por este projeto já ter sido finalizado não será possível aplicar todo o modelo proposto, sendo que os artefatos construídos não serão de fato utilizados no desenvolvimento.

A utilização de um projeto já finalizado possibilitará a comparação entre as horas estimadas de forma empírica, as horas estimadas conforme o modelo proposto e as horas reais do projeto.

Além disso, não foi possível aplicar a simulação em um projeto durante o seu desenvolvimento, pelo fato da empresa não possuir nenhum projeto de pequena ou média escala a ser iniciado até o final do ano. A maioria dos projetos desenvolvidos ao decorrer dos últimos meses na empresa são projetos de grande dimensão, que ultrapassam duas mil horas de desenvolvimento e, por isso, ficaria inviável a aplicação da simulação do método de estimativa nestes projetos.

### **6.1. Características do Projeto**

O projeto que será utilizado é o de requisição de materiais, cujo objetivo foi criar um processo para tratar solicitações de itens pela fábrica e sua respectiva entrega pelo almoxarifado diretamente no setor onde o material é necessário.

Contempla a requisição de materiais a partir de ordens de fabricação e avulsas. Quando a requisição for realizada a partir de ordens de fabricação, o local de entrega dos itens será o almoxarifado que está na ordem de fabricação. Quando a requisição for avulsa, o usuário da fábrica/expedição responsável por gerar a solicitação deve informar o local de entrega do item solicitado.

Após a geração da solicitação, o usuário responsável pelo almoxarifado emite um relatório onde constam as informações dos itens que devem ser separados e os locais onde estes devem ser entregues. Este relatório permite selecionar o almoxarifado desejado, facilitando assim, o trabalho do almoxarife que irá listar somente os itens que são de sua responsabilidade. Assim que os itens estiverem separados, o usuário da expedição realiza transferência do estoque da expedição para a fábrica informando as quantidades realmente separadas e após, entrega os materiais no local determinado na solicitação.

Caso sejam necessárias alterações nas solicitações, deve ser possível corrigi-las cancelando ou alterando as quantidades (quando estas não estiverem sido atendidas).

## **6.2. Simulação de Estimativa do Projeto Utilizando a Ferramenta**

Para simular a estimativa no modelo proposto é necessário possuir a informação dos casos de uso referentes ao projeto. Deste modo foram construídos os casos de uso descritivos com base nos requisitos especificados. Após concluídos estes casos de uso foram validados pela analista de sistemas que implementou o projeto. Os casos de uso construídos podem ser observados nos anexos de 24 a 28.

É importante lembrar que no modelo proposto foi previsto a existência de um sistema paralelo a ferramenta de estimativa para o gerenciamento dos casos de uso descritivos. Como esta ferramenta ainda não existe os casos de uso descritivos foram incluídos manualmente nas tabelas de importação que farão a integração entre os sistemas. Estes casos de uso serão posteriormente importados para a estimativa.

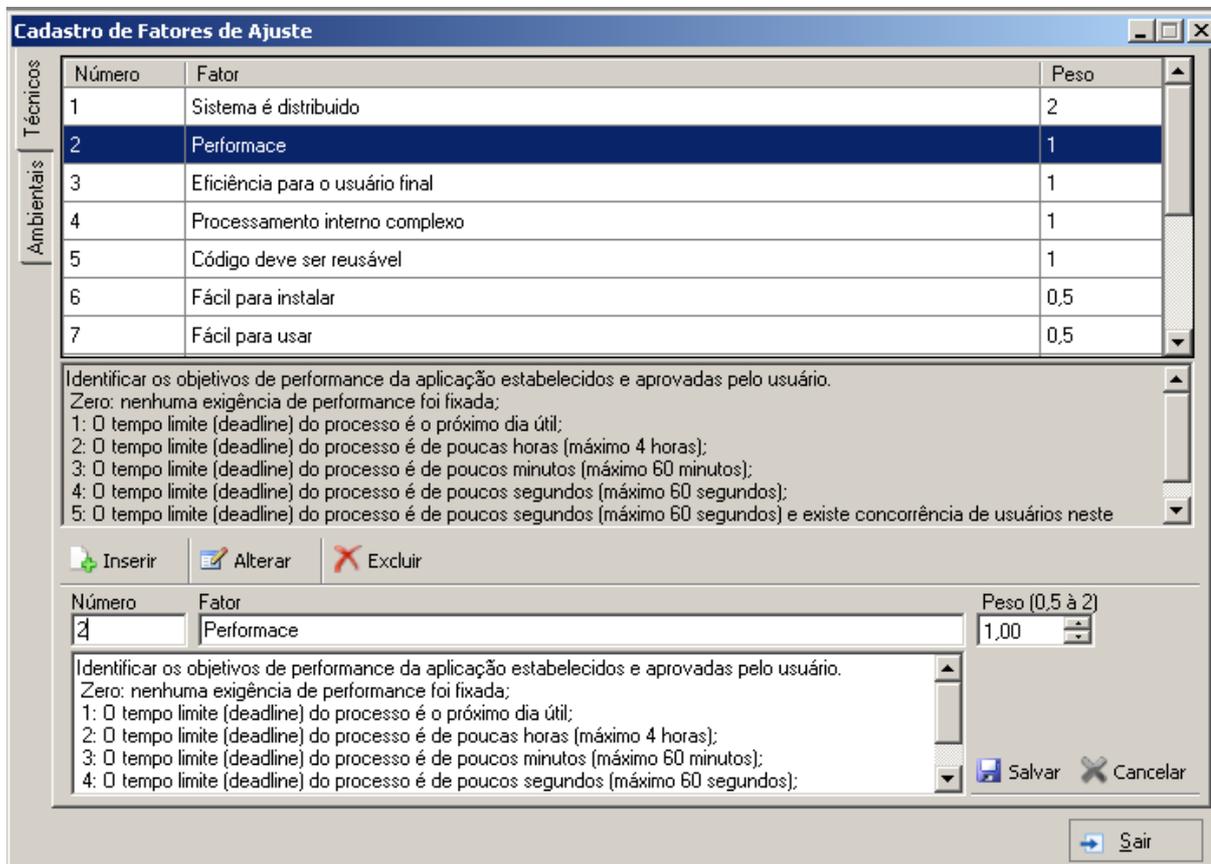
A seguir será apresentada a sequência de atividades realizadas para a estimativa de tempo do projeto de requisição de materiais:

### **a. Cadastro dos fatores de ajuste e definição dos pesos:**

Antes de realizar a primeira estimativa utilizando o protótipo de sistema de estimativa construído, é necessário configurar as informações conforme o processo de desenvolvimento em que será aplicado. Uma destas informações são os fatores de ajuste.

Para o cadastro, por se tratar do início da utilização do método de estimativa, serão utilizados os fatores de ajuste e pesos configurados conforme a sugestão da literatura. Estes fatores técnicos e ambientais e seus respectivos pesos podem ser observados nas tabelas 10 e 11 respectivamente, que encontram-se na seção 2 referente ao estudo das métricas.

Através deste cadastro é possível configurar também uma descrição da maneira de atribuição de valor para cada fator de ajuste, com o intuito de auxiliar na sua classificação durante os projetos, conforme pode ser observado na figura 13. Esta descrição deve esclarecer a forma e impacto do valor avaliado ao fator de ajuste.



**Figura 13 - Cadastro de fatores de ajuste**

**b. Cadastro das fases do processo de desenvolvimento:**

Para a configuração das fases do processo de desenvolvimento foram cadastradas as principais fases da estimativa sugerida à empresa, juntamente com os processos macro de desenvolvimento. Este cadastro, que está sendo exibido na figura 14, pode ser alterado conforme a necessidade da empresa e deve guiar o desenvolvimento de todos os projetos, auxiliando na tarefa garantir que estes seguirão o processo definido.

**Cadastro de Fases**

Seq.	Fase	Descrição	Realiza Estimativa
1	Entender o Escopo	Levantamento inicial de requisitos e UC descritivos/diagrama.	Não
2	Estimar Tempo (p/ Orçamento)	Estimativa de tempo para aprovação do projeto.	Sim
3	Análise Detalhada - Revisão I	Revisão da estimativa realizada durante a elaboração da análise deta...	Sim
4	Análise Detalhada - Revisão II	Revisão da estimativa realizada ao final da elaboração da análise det...	Sim
5	Construção e Testes	Desenvolvimento e validação do projeto.	Não
6	Liberação do Projeto	Liberação do projeto.	Não

Seq. Fase Descrição  
 1 Entender o Escopo Levantamento inicial de requisitos e UC descritivos/diagrama.

Realiza Estimativa

**Figura 14 - Cadastro de fases**

### c. Cadastro do projeto

Após realizar as configurações básicas do ambiente é possível cadastrar o projeto propriamente dito, conforme pode ser observado na figura 15. Para este projeto, por a estimativa ser com base nos casos de uso descritivos, a forma de atribuição da complexidade dos casos de uso será por transação. O número de horas por ponto de caso de uso definido pela empresa é oito, porém, é importante observar que este número deve ser verificado a medida que a empresa possuir uma base de dados históricos.

O projeto irá utilizar a importação de casos de uso do sistema de casos de uso descritivos, o qual conforme mencionado anteriormente ainda não existe. Por isto, para esta simulação foram incluídos os dados manualmente nas tabelas de integração. Posteriormente estes dados serão alimentados por tal sistema.

Projeto	Descrição	Base Complex. UC	Pessoas Hrs. UCP	Dt. Início	Imp. UC
1	Requisição de materias.	Transações	8	01/01/2012	Sim

Inserir   Alterar   Excluir

Projeto: 1   Descrição: Requisição de materias.

Base Complexidade UC: Transações   Pessoa/hora por UCP: 8   Dt. Início: 01/01/2012    Utiliza Importação de UC

Salvar   Cancelar   Sair

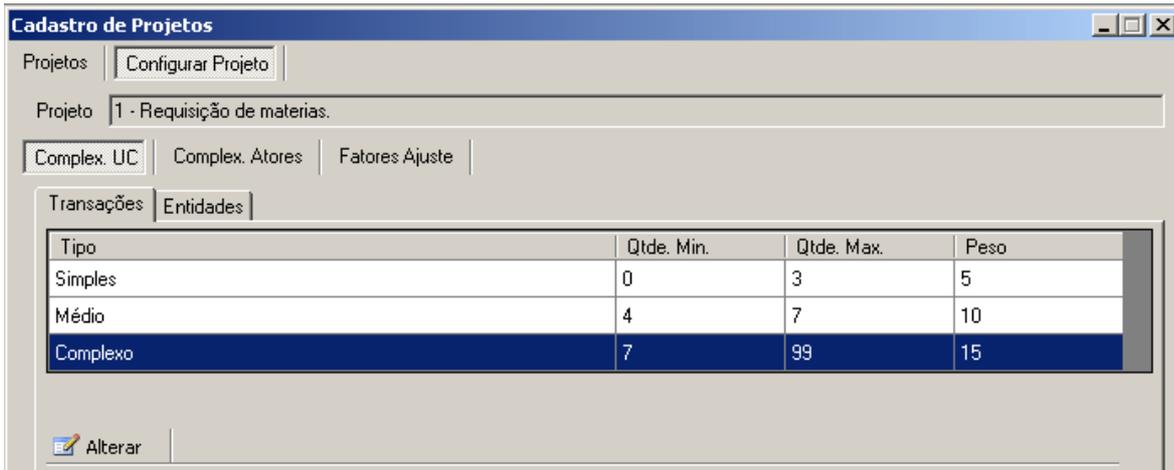
**Figura 15 - Cadastro de Projetos**

#### **d. Configuração do projeto**

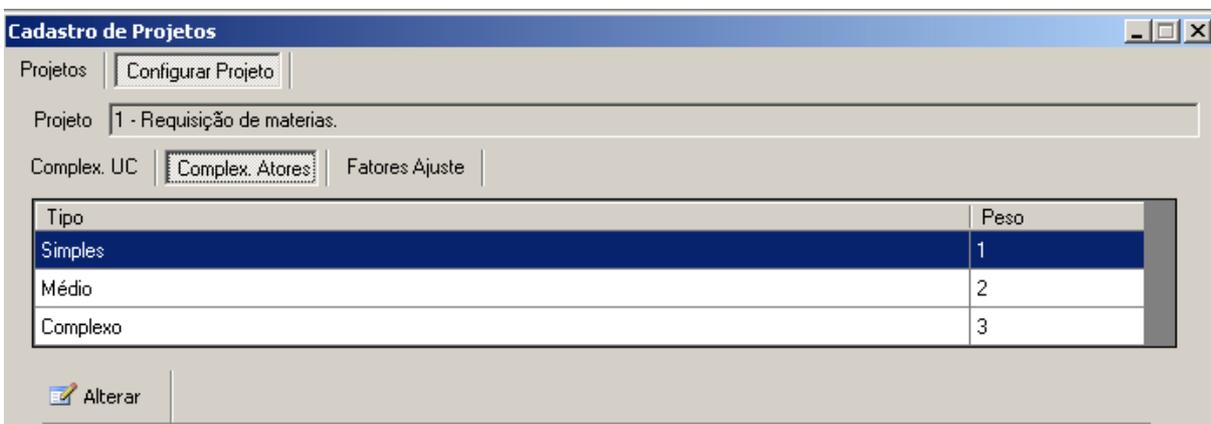
O próximo passo antes de realizar a estimativa é configurar o projeto. A configuração dos casos de uso e atores do projeto de requisição de materiais pode ser observada nas figuras 16 e 17.

Para a simulação, os valores da complexidade dos atores e casos de uso atribuídos foram os sugeridos na literatura conforme as tabelas 8 e 9 do trabalho. O peso da complexidade dos atores simples, médios e complexos variam de um a três respectivamente.

Já o valor da complexidade dos casos de uso pode ser atribuída conforme o número de transações ou entidades envolvidas no mesmo. Em ambos os casos, os pesos dos UC simples, médio e complexo são cinco, dez e quinze respectivamente, o que muda de uma forma para a outra é o número de entidades ou transações que indicam a classificação da complexidade.



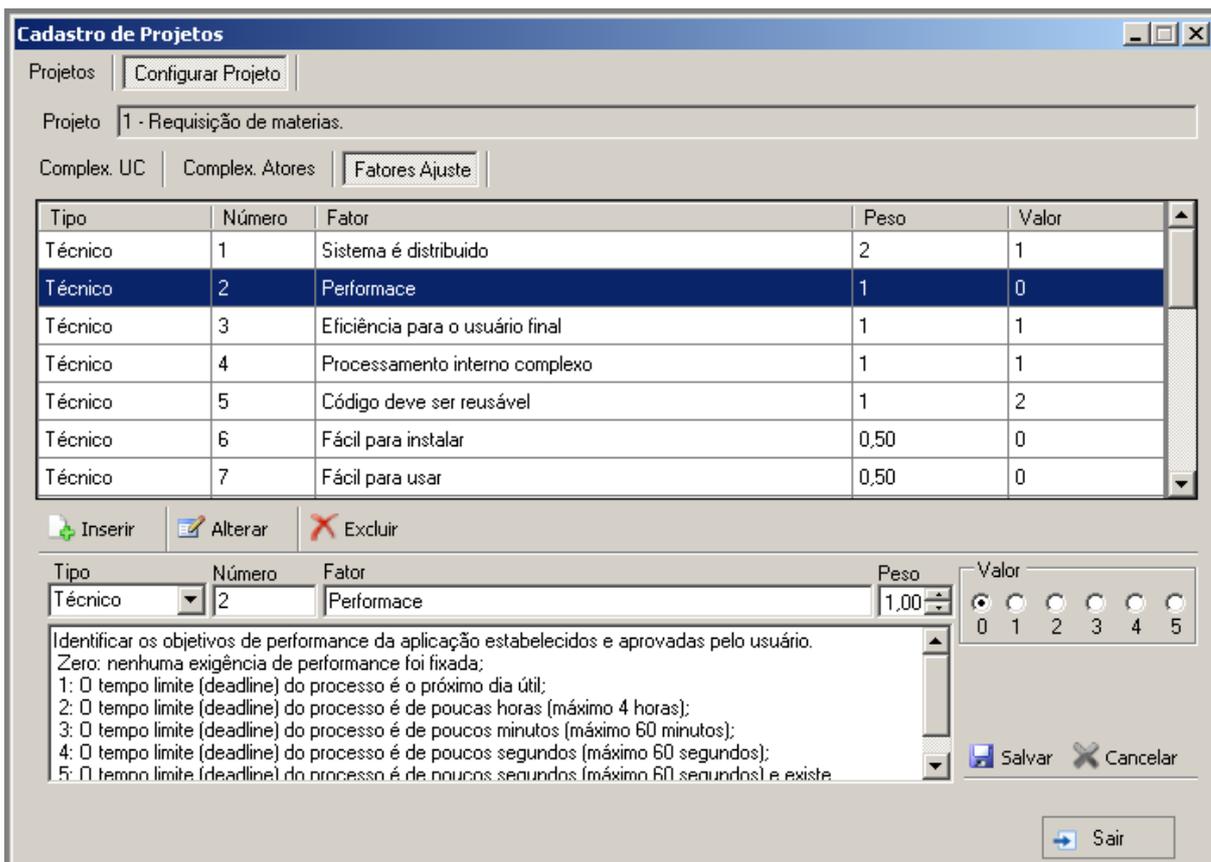
**Figura 16 - Configurar projeto: Complexidade UC**



**Figura 17 - Configurar projeto: Complexidade atores**

A definição dos valores dos fatores de ajuste do projeto foi feita com o auxílio da analista de sistemas que desenvolveu o projeto, relatando as características técnicas e ambientais do momento em que o projeto foi desenvolvido. Esta configuração dos fatores de ajuste do projeto pode ser observada na figura 18.

Tal configuração deve indicar o nível de influência de cada fator de ajuste no cenário de desenvolvimento do projeto, por exemplo, para o segundo fator de ajuste técnico que trata da influência da performance nas operações, foi atribuído valor zero, pois nenhuma exigência com relação a este fator de foi fixada para o desenvolvimento.

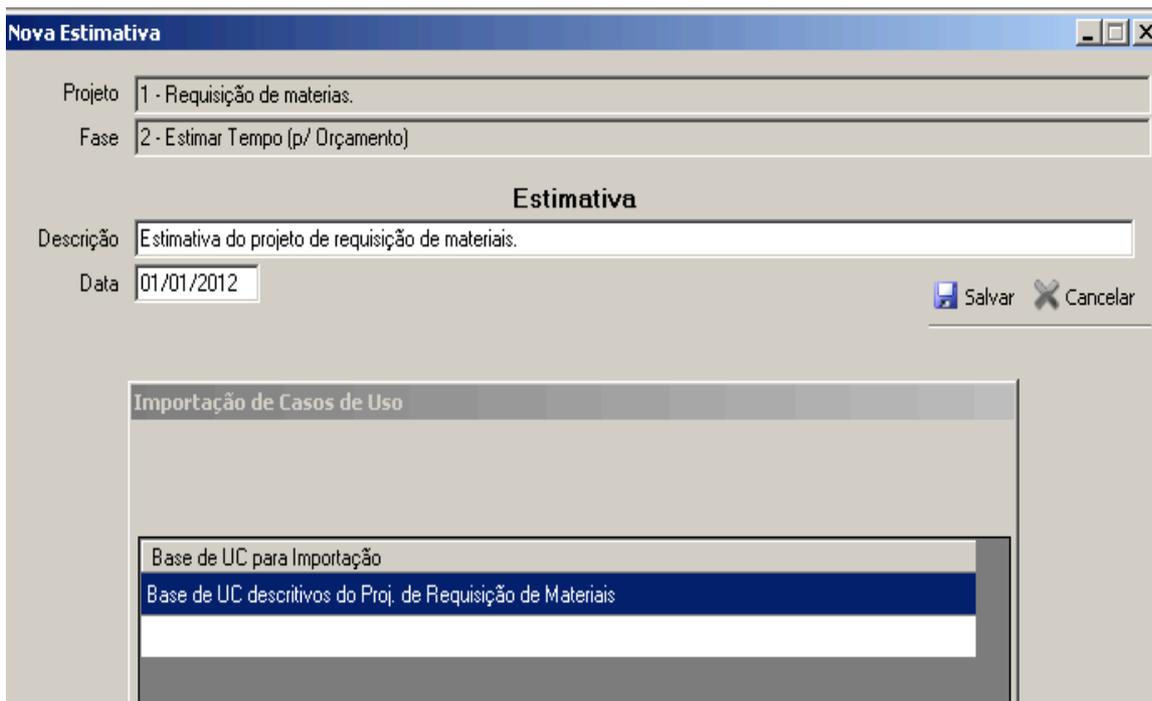


**Figura 18 - Configurar projeto: Fatores de ajuste**

**e. Criação da estimativa:**

Após o projeto ter sido configurado foi criada a estimativa de tempo do projeto de requisição de materiais. Para isso, conforme havia sido configurado no projeto foi utilizada a importação de dados da base de casos de uso descritivos, como pode ser observado na figura 19.

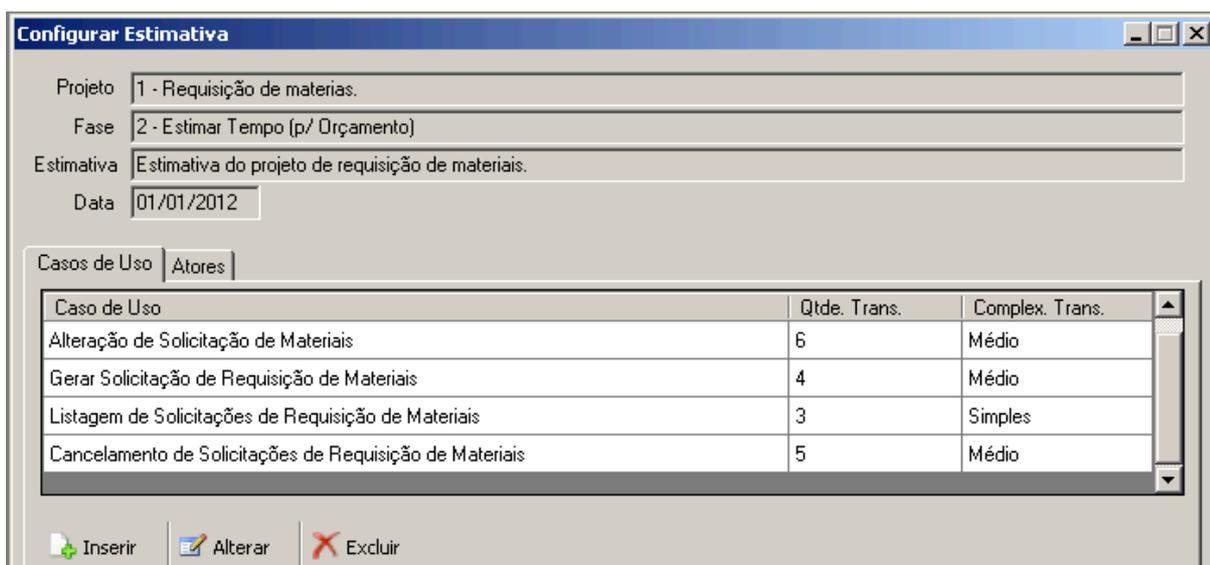
Através desta importação não haverá necessidade de repetir o cadastro de atores e casos de uso, uma vez que estes teriam sido realizados através do sistema de casos de uso descritivos.



**Figura 19 - Cadastro da estimativa do projeto**

**f. Configuração da estimativa:**

Os dados informados nos casos de uso descritivos foram importados para a estimativa criada. Com base nos passos dos fluxos de cada caso de uso, foi atribuída automaticamente a complexidade dos casos de uso importados, conforme a figura 20.



**Figura 20 - Complexidade dos casos de uso**

O passo seguinte para a estimativa é definir a complexidade dos atores do projeto. Como os dois atores interagem diretamente com interfaces gráficas, os mesmos foram

classificados como complexos, conforme pode ser observado na figura 21.

Ator	Complexidade
Planejador fábrica.	Complexo
Usuário da Expedição	Complexo

**Figura 21 - Complexidade dos atores**

#### **g. Cálculo dos casos de uso não ajustados**

Com isso, foi possível calcular o valor dos casos de uso não ajustados, levando em consideração os casos de uso e atores informados para o projeto, conforme pode ser verificado nas figura 22.

O cálculo feito pela ferramenta de estimativa implementa as fórmulas explicadas anteriormente neste trabalho, que consistem em somar o peso dos casos de uso ao peso dos atores. Na tabela 19 exibida a seguir, pode ser observada a estrutura deste cálculo, aplicada aos dados do projeto de requisição de materiais.

Através do cálculo dos casos de uso não ajustados foi possível verificar que foram obtidos cinquenta e um pontos, o que multiplicado pelas horas por caso de uso resultou no tempo estimado do projeto de quatrocentos e oito horas.

**Tabela 19 - Cálculo dos casos de uso não ajustados**

Casos de Uso			
Caso de Uso	Qtde.	Tipo	Peso
Gerar solicitação de requisição de materiais	4	Médio	10
Listagem de solicitações de requisição de materiais	3	Simples	5
Requisição de materiais a partir de uma solicitação	6	Médio	10
Alteração de solicitação de materiais	6	Médio	10
Cancelamento de Solicitações de materiais	5	Médio	10
<b>Total</b>			<b>45</b>
Atores			
Ator		Tipo	Peso
Planejador Fábrica		Complexo	3
Usuário da Expedição		Complexo	3
<b>Total</b>			<b>6</b>
<b>UUCP = 45 + 6 = 51</b>			
<b>Hrs. Pessoa por UCP = 8</b>			
<b>UCP = 51 * 8 = 408</b>			

The screenshot shows a window titled "Calcular Estimativa" with the following fields and values:

- Projeto: 1 - Requisição de materias.
- Fase: 2 - Estimar Tempo (p/ Orçamento)
- Estimativa: Estimativa do projeto de requisição de materiais.
- Data: 01/01/2012

Below these fields are two tabs: "UUCP" (selected) and "UCP". Under the "UUCP" tab, the section is titled "UC não Ajustados (UUCP)".

Fields in the "UUCP" section:

- Calcular o peso dos UC através da complexidade: Transações (dropdown menu)
- Peso dos Atores (UAW): 6
- Pesos dos Casos de Uso (UUCW): 45
- Valor UC não ajustados (UUCP): 51
- Tempo do Projeto: 408

A "Calcular/Recalcular" button is located at the bottom right of the section.

**Figura 22 - Casos de uso não ajustados**

#### **h. Cálculo dos casos de uso ajustados**

O cálculo dos casos de uso ajustados consiste na aplicação dos fatores de ajuste, conforme os valores definidos para o projeto, aos pontos obtidos no cálculo dos casos de uso não ajustados.

Com a aplicação dos fatores de ajuste, será feita a união do esforço de construção dos casos de uso com as características técnicas e ambientais definidas. Na tabela 20 pode ser observada a aplicação das fórmulas correspondentes ao cálculo utilizando os dados especificados no projeto. A seguir na figura 23 pode ser observado este mesmo cálculo realizado pela ferramenta de estimativa, sendo obtidos em torno de quarenta e cinco pontos, o que significa um esforço em torno de trezentos e sessenta e quatro horas de desenvolvimento.

**Tabela 20 - Cálculo dos casos de uso ajustados**

<b>Fatores de Ajuste</b>			
<b>Fator</b>	<b>Peso</b>	<b>Valor Avaliado (0 - 5)</b>	<b>Valor</b>
<b>Fatores Técnicos</b>			
Sistema é distribuído	2,0	1	2
Performance	1,0	0	0
Eficiência para o usuário final	1,0	1	1
Processamento interno complexo	1,0	1	1
Código deve ser reusável	1,0	2	2
Fácil para instalar	0,5	0	0
Fácil para usar	0,5	0	0
Portátil	1,5	0	0
Fácil de mudar	1,0	2	2
Concorrente	1,0	4	4
Seguro	1,0	0	0
Acessível por terceiros	1,0	0	0
Requer treinamento especial	1,0	4	4
<b>Total</b>			<b>16</b>
<b>TCF = 0,6 + (0,01 * (16)) = 0,76</b>			
<b>Fatores Ambientais</b>			
Familiaridade da equipe	1,5	2	3
Experiência na Aplicação	0,5	3	1,5
Equipe com Experiência em OO	1,0	0	0
Capacidade de Análise	0,5	2	1
Motivação	1,0	3	3
Requisitos Estáveis	2,0	2	4
Recursos em Meio Período (Part-Time)	-1,0	3	-3
Dificuldade na linguagem de programação	-1,0	2	-2
<b>Total</b>			<b>7,5</b>
<b>ECF = 1,4 + (-0,03 * (7,5)) = 1,175</b>			
<b>UCP = 51 * 0,76 * 1,175 = 45,54</b>			
<b>Hrs. Pessoa por UCP = 8</b>			
<b>Tempo UUCP = 45,54 * 8 = 364,34</b>			

UC Ajustados (UCP)	
Valor dos Fatores de Complexidade Técnica (TCF)	0,76
Valor dos Fatores de Complexidade Ambiental (ECF)	1,175
Valor dos UC Ajustados (UCP)	45,543
Tempo do Projeto	364,344

**Figura 23 - Casos de uso ajustados**

### 6.3. Comparação das Estimativas

Conforme já havia sido comentado durante o mapeamento do processo de desenvolvimento, apesar da empresa ter desenhado um processo de estimativa de tempo dos projetos utilizando UCP, muitos ainda não estão sendo estimados desta forma, sendo o projeto de requisição de materiais um exemplo disto.

Deste modo, as horas deste projeto foram estimadas informalmente, ou seja, sem seguir nenhuma metodologia, utilizando como base apenas os requisitos apontados nas características do projeto e o conhecimento prévio da analista de sistemas no módulo de suprimentos.

Para determinar o tempo do projeto a analista de sistemas criou, informalmente, um documento com as principais implementações e o respectivo número de horas estimado. Uma cópia deste documento, sem as informações referentes a empresa, pode ser observado no anexo 23 do trabalho.

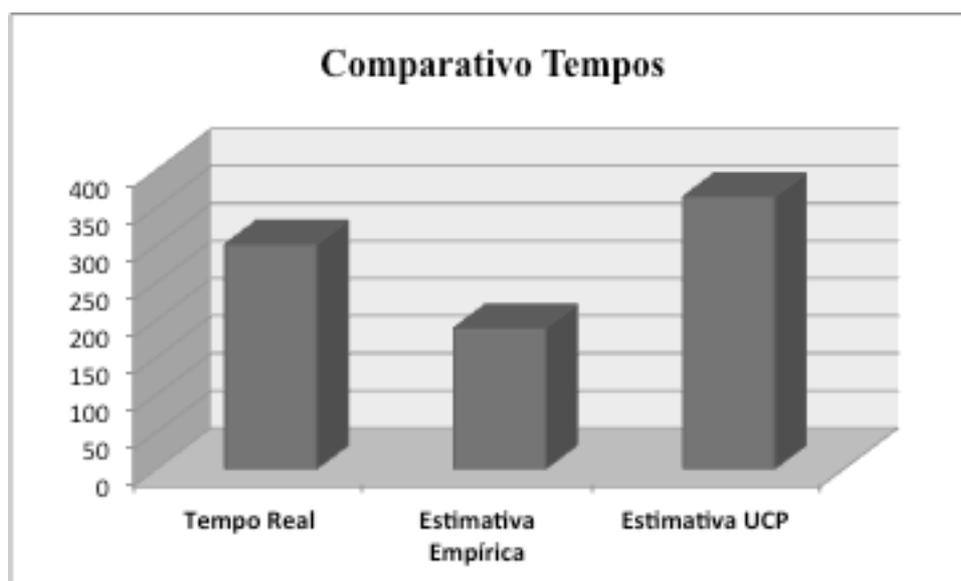
Com base neste documento, onde o tempo total previsto para o projeto é de cento e oitenta e nove horas, pode ser observado o baixo nível de detalhamento das informações no momento da estimativa, causado normalmente pelo tempo reservado para tal atividade, que

neste caso foi de seis horas e trinta minutos. Além disso, não foram consideradas nesta estimativa horas para a construção da documentação do projeto, a qual foi criada posteriormente durante o decorrer do mesmo.

Conforme visto no subcapítulo anterior através da aplicação do modelo de estimativa proposto, o número de horas calculas para a realização do projeto foi de trezentos e sessenta e quatro horas.

O tempo em que o projeto foi realizado segundo os registros da empresa foi de trezentas e uma horas, porém, neste tempo não foram contabilizados o tempo dos treinamentos técnicos e de negócio ministrados para as equipes de desenvolvimento e consultoria ao final do projeto.

Com isso, é possível verificar que o tempo estimado através do método proposto, com base nos casos de uso descritivos, foi mais assertivo do que a estimativa feita de forma empírica, conforme representado na figura 24.



**Figura 24 - Comparativo de tempo das estimativas**

É importante observar que foi possível, mesmo que com a utilização dados históricos, aplicar o método de estimativa proposto utilizando o protótipo de ferramenta construído. Através da análise dos dados deste projeto no ambiente real de desenvolvimento pode-se constatar a aderência do modelo. No entanto, em um processo de mudança da forma de estimativa será necessário contar com o empenho da empresa e da equipe na utilização do novo modelo, uma vez que isto implicará na mudança da maneira com que são construídos alguns artefatos de análise atualmente.

## 7. CONCLUSÃO

Durante os estudos feitos sobre as métricas de estimativa de software puderam ser observadas as características e o formato de cálculo de cada métrica bem como relacioná-las com as particularidades do processo de desenvolvimento mapeado. Foi possível então, indicar a métrica mais aderente ao processo e as adaptações necessárias para a sua aplicação.

Com base nisto, foram definidos os requisitos da ferramenta e elaborada a modelagem correspondente a fase de análise e projeto. Através dos artefatos construídos durante esta modelagem foi possível desenvolver e validar o protótipo de ferramenta de estimativa, quanto as características e funcionalidades implementadas.

Através da simulação de uma estimativa de projeto utilizando dados históricos do ambiente alvo do trabalho, foi possível constatar a aderência da ferramenta construída com o método de estimativa proposto, bem como com as características do processo de desenvolvimento.

Com o levantamento de dados referentes ao projeto utilizado na simulação da estimativa foi possível observar também, a dificuldade de compreensão da complexidade do projeto durante a estimativa feita de forma empírica pela equipe de desenvolvimento da empresa.

O método de estimativa proposto permitirá o acompanhamento das estimativas e suas revisões o que pode auxiliar a empresa a identificar motivos de variações entre a estimativa inicial e suas revisões e com isso, melhorar gradativamente o modelo conforme forem sendo analisados os dados históricos dos projetos.

Nos objetivos deste trabalho constavam a proposta de um modelo para estimativa de projetos aderente a um processo de desenvolvimento específico e sua aplicação neste ambiente, através de um protótipo de ferramenta construído. Através da simulação realizada foi possível observar a comprovação desta aderência e uma melhora na assertividade da estimativa quando comparada a forma empírica.

Por fins de comparação de tempos e pelos demais motivos especificados, esta simulação foi realizada com dados reais, porém, históricos de um projeto já concluído. Por isso, uma sugestão de trabalho futuro seria a aplicação do modelo em outros projetos, criando assim uma base de dados históricos, podendo medir melhor a eficiência e com isso e evoluir a estimativa.

Como a proposta do trabalho foi a construção de um protótipo de ferramenta em que

fosse possível a aplicação do processo de estimativa sugerido, alguns aspectos do software precisam ser aprimorados para a sua utilização como um sistema de estimativa. Os principais aspectos, que poderiam ser implementados em trabalhos futuros são quanto a interface da ferramenta, *help* do sistema e o próprio sistema de gerenciamento de casos de uso juntamente com sua integração com a ferramenta de estimativa.

Além disso, outra sugestão de trabalho futuro seria a aplicação do modelo de estimativa em outros ambientes que possuam o processo de desenvolvimento orientado a casos de uso, uma vez que o processo de estimativa sugerido utiliza os artefatos produzidos pela modelagem UML, e por este motivo pode ser aplicado em outros cenários que possuam tais características.

Vale ressaltar, porém, que se tratando da aplicação do modelo em outros cenários sugere-se mapear as características e especificidades de tais processos, para adequar a configuração das estimativas e o número de revisões conforme a necessidade de cada ambiente.

Sendo a utilização deste modelo no ambiente alvo do trabalho ou em outros ambientes com características semelhantes, deve-se atentar para importância de manter o processo de estimativa em um aprendizado contínuo, pois somente desta forma o modelo poderá se tornar mais assertivo agregando eficácia ao processo de desenvolvimento.

## 8. REFERÊNCIAS

AGUIAR, Mauricio. Estimando os Projetos com COCOMO II. 2002. Disponível em: <[http://www.univasf.edu.br/~ricardo.aramos/disciplinas/ES\\_I\\_2010\\_2/COCOMO\\_II.pdf](http://www.univasf.edu.br/~ricardo.aramos/disciplinas/ES_I_2010_2/COCOMO_II.pdf)>. Acesso em: 17 jun. 2011.

ANSELMO, Fernando. Métricas para Desenvolvedores. Florianópolis: Visual Books, 2010.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS; ISO/IEC 20926: Software and systems engineering - Software measurement – IFPUG functional size measurement method, 2009.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS; ISO/IEC 14143-1: 2007 – Tecnologia da informação — Medição de software — Medição de tamanho funcional: Parte 1: Definição de conceitos (inglês).

BEZERRA, Eduardo. Princípios de análise e projeto de sistemas com UML. Rio de Janeiro, RJ: Elsevier, 2003.

BRASIL. Ministério da Ciência e Tecnologia. Secretaria de Política de Informática. Pesquisa de qualidade no setor de software brasileiro 2009. Ministério da Ciência e Tecnologia, 2010.

BFPUG, BrazilianFunction Point Users Grou. Disponível em: <http://www.bfpug.com.br>.

COCKBURN, Alistair. Escrevendo Casos de Uso Eficazes. Porto Alegre: Bookman, 2005.

COHN, Mike. EstimatingWith Use Case Points. 2006. Disponível em: <[http://www.bfpug.com.br/Artigos/UCP/Cohn-Estimating\\_with\\_UCP.pdf](http://www.bfpug.com.br/Artigos/UCP/Cohn-Estimating_with_UCP.pdf)>. Acesso em: 19 maio 2011.

DAMODARAN, Mel; Washington, Aqua Netta. EstimationUsing Use Case Points. 2003. Disponível em: <[http://www.bfpug.com.br/Artigos/UCP/Damodaran-Estimation\\_Using\\_Use\\_Case\\_Points.pdf](http://www.bfpug.com.br/Artigos/UCP/Damodaran-Estimation_Using_Use_Case_Points.pdf)>. Acesso em: 25 maio 2011.

FOWLER, Martin. UML essencial : um breve guia para a linguagem-padrão de modelagem de objetos. 3.ed. Porto Alegre: Bookman,2005-. . Disponível em : <<http://ucsvirtual.ucs.br/>>. Acesso em : 13 jul. 2011.

HEUSER, Carlos Alberto. Projeto de banco de dados. 6.ed. Porto Alegre: Bookman, 2009.

Kitchenham, Barbara. The ProblemwithFunction Points. In: IEEE Software, vol. 14, n. 2, p. 29-31, Mar./Apr. 1997. Disponível em: <<http://doi.ieeecomputersociety.org/10.1109/MS.1997.582972>>. Acesso em: 17 jun. 2011.

LARMAN, Craig. Utilizando UML e padrões: uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo. 3.ed. Porto Alegre: Bookman, 2007.

PRESSMAN, Roger S.- Engenharia de Software. 6ª Ed. São Paulo: McGraw-Hill, 2006.

SOMMERVILLE, Ian. Engenharia de software. 6.ed. São Paulo: Addison-Wesley, 2007.

VAZQUEZ, Carlos Eduardo; SIMÕES, Guilherme Siqueira; ALBERT, Renato Machado. Análise de pontos de função: medição, estimativas e gerenciamento de projetos de

software. 8.ed. rev. atual. e ampl. São Paulo, SP: Érica, 2008.

WAZLAWICK, Raul Sidnei. Análise e projeto de sistemas de informação orientados a objetos. Rio de Janeiro: Elsevier, 2004.

## 9. ANEXOS

ANEXO 1 – Formato do documento de especificação geral

ANEXO 2 – Exemplo de diagrama de casos de uso

ANEXO 3 – Modelo de caso de uso expandido sugerido

ANEXO 4 – Protótipos UC Cadastrar Fatores de Ajuste

ANEXO 5 – Protótipos UC Cadastrar Fases

ANEXO 6 – Protótipos UC Cadastrar Projetos

ANEXO 7 – Protótipos UC Configurar Projeto

ANEXO 8 – Protótipos UC Importar/Informar Atores e Casos de Uso e a sua Complexidade

ANEXO 9 – Protótipo UC Calcular/Recalcular UC não Ajustados

ANEXO 10 – Protótipo UC Calcular/Recalcular UC Ajustados

ANEXO 11 – Protótipo UC Finalizar o Projeto

ANEXO 12 – Protótipo UC Comparar Estimativas

ANEXO 13 – Diagrama de Classes Conceitual

ANEXO 14 – Diagrama de Atividades

ANEXO 15 – Diagrama de Sequência: Cadastrar Fases

ANEXO 16 – Diagrama de Sequência: Estimar Esforço do Projeto

ANEXO 17 – Diagrama de Sequência: Cadastrar Fatores de Ajuste

ANEXO 18 – Diagrama de Sequência: Cadastrar Projetos

ANEXO 19 – Diagrama de Sequência: Configurar Projeto

ANEXO 20 – Diagrama de Sequência: Finalizar Projeto

ANEXO 21 – Diagrama de Classes de Projeto

ANEXO 22 – Diagrama ER

ANEXO 23 – Documento de Estimativa do Projeto de Requisição de Materiais

ANEXO 24 – Caso de Uso: Gerar Solicitação de Requisição de Materiais

ANEXO 25 – Caso de Uso: Listagem de Solicitações de Requisição de Materiais

ANEXO 26 – Caso de Uso: Requisição de Materiais a partir de uma Solicitação

ANEXO 27 – Caso de Uso: Cancelamento de Solicitações de Requisição de Materiais

ANEXO 28 – Caso de Uso: Alteração de Solicitação de Materiais

## ANEXO 1–Formato do documento de especificação geral

### Sumário

1	Apresentação (GERENTE DO PROJETO) .....	3
2	Objetivos do Projeto (GERENTE DO PROJETO).....	3
3	Informações Relativas ao Negócio (ANALISTA DE NEGÓCIO) .....	3
4	Requisitos Funcionais (ANALISTA DE SISTEMA) .....	4
5	Requisitos Técnicos (ANALISTA DE SISTEMA) .....	4
6	Requisitos Não Contemplados .....	4
7	Funcionalidades (ANALISTA DE SISTEMA).....	5
7.1	Forms.....	5
7.1.1	Forms (1).....	5
7.2	Rotina de Integração .....	6
7.2.1	Rotina de Integração (1) .....	6
7.3	Tabela .....	7
7.3.1	Tabela (1) .....	7
7.4	Procedure .....	7
7.4.1	Procedure (1) .....	7
7.5	Trigger .....	8

#### 1 Apresentação (GERENTE DO PROJETO)

Projeto:	<i>Número e Nome do Projeto</i>		
Solicitante:	<i>Pessoa e entidade que requisitou o projeto.</i>		
Data de criação:			
Responsável:			
Classificação:	Módulo	Programa	Funcionalidade
	Melhoramento	Produto	
Versão de Liberação:	<i>Versão a ser liberado o presente projeto para o cliente final.</i>		

#### 2 Objetivos do Projeto (GERENTE DO PROJETO)

#### 3 Informações Relativas ao Negócio (ANALISTA DE NEGÓCIO)

#### 4 Requisitos Funcionais (ANALISTA DE SISTEMA)

#### 5 Requisitos Técnicos (ANALISTA DE SISTEMA)

#### 6 Requisitos Não Contemplados

#### 7 Funcionalidades (ANALISTA DE SISTEMA)



### **ANEXO 3– Modelo de caso de uso expandido sugerido**

<ul style="list-style-type: none"><li>▪ Nome do Caso de uso<ul style="list-style-type: none"><li>○ Descrição Resumida</li><li>○ Atores</li></ul></li></ul>
<ul style="list-style-type: none"><li>▪ Fluxos de Eventos<ul style="list-style-type: none"><li>○ Fluxo Básico</li><li>○ Fluxos Alternativos</li></ul></li></ul>
<ul style="list-style-type: none"><li>▪ Requisitos Especiais</li></ul>
<ul style="list-style-type: none"><li>▪ Pré-condições</li></ul>
<ul style="list-style-type: none"><li>▪ Pós-condições</li></ul>

## ANEXO 4 – Protótipos UC Cadastrar Fatores de Ajuste

**Cadastro de Fatores de Ajuste**

Número	Fator	Peso
1	Sistema é distribuído	2
2	Performace	1
3	Eficiência para o usuário final	1
4	Processamento interno complexo	1
5	Código deve ser reusável	1
6	Fácil para instalar	0,5
7	Fácil para usar	0,5

Identificar os objetivos de performance da aplicação estabelecidos e aprovadas pelo usuário.  
Zero: nenhuma exigência de performance foi fixada;  
1: O tempo limite (deadline) do processo é o próximo dia útil;  
2: O tempo limite (deadline) do processo é de poucas horas (máximo 4 horas);  
3: O tempo limite (deadline) do processo é de poucos minutos (máximo 60 minutos);  
4: O tempo limite (deadline) do processo é de poucos segundos (máximo 60 segundos);  
5: O tempo limite (deadline) do processo é de poucos segundos (máximo 60 segundos) e existe concorrência de usuários neste

Inserir Alterar Excluir

Número	Fator	Peso (0,5 à 2)
2	Performace	1,00

Identificar os objetivos de performance da aplicação estabelecidos e aprovadas pelo usuário.  
Zero: nenhuma exigência de performance foi fixada;  
1: O tempo limite (deadline) do processo é o próximo dia útil;  
2: O tempo limite (deadline) do processo é de poucas horas (máximo 4 horas);  
3: O tempo limite (deadline) do processo é de poucos minutos (máximo 60 minutos);  
4: O tempo limite (deadline) do processo é de poucos segundos (máximo 60 segundos);

Salvar Cancelar

Sair

## ANEXO 5 – Protótipos UC Cadastrar Fases

**Cadastro de Fases**

Seq.	Fase	Descrição	Realiza Estimativa
1	Entender o Escopo	Levantamento inicial de requisitos e UC descritivos/diagrama.	Não
2	Estimar Tempo (p/ Orçamento)	Estimativa de tempo para aprovação do projeto.	Sim
3	Análise Detalhada - Revisão I	Revisão da estimativa realizada durante a elaboração da análise deta...	Sim
4	Análise Detalhada - Revisão II	Revisão da estimativa realizada ao final da elaboração da análise det...	Sim
5	Construção e Testes	Desenvolvimento e validação do projeto.	Não
6	Liberação do Projeto	Liberação do projeto.	Não

 Inserir    Alterar    Excluir

Seq. Fase Descrição

1 Entender o Escopo Levantamento inicial de requisitos e UC descritivos/diagrama.

Realiza Estimativa

 Salvar    Cancelar

 Sair

## ANEXO 6 – Protótipos UC Cadastrar Projetos

**Cadastro de Projetos**

Projetos | Configurar Projeto

Projeto	Descrição	Base Complex. UC	Pessoas Hrs. UCP	Dt. Início	Imp. UC
10	Requisição de materias.	Ambas	10	01/01/2012	Sim

Inserir | Alterar | Excluir

Projeto: 10 | Descrição: Requisição de materias.

Base Complexidade UC: Ambas | Pessoa/hora por UCP: 10 | Dt. Início: 01/01/2012 |  Utiliza Importação de UC

Salvar | Cancelar

Sair

## ANEXO 7 – Protótipos UC Configurar Projeto

**Cadastro de Projetos**

Projetos | **Configurar Projeto**

Projeto | 10 - Requisição de materias.

Complex. UC | Complex. Atores | Fatores Ajuste

Transações | Entidades

Tipo	Qtde. Min.	Qtde. Max.	Peso
Simple			
Médio			
Complexo			

 Alterar

Tipo | Qtde. Min. | Qtde. Max. | Peso

Simple | [ ] | [ ] | [ ]

 Salvar  Cancelar

 Sair

**Cadastro de Projetos**

Projetos | **Configurar Projeto**

Projeto | 10 - Requisição de materias.

Complex. UC | **Complex. Atores** | Fatores Ajuste

Tipo	Peso
Simple	
Médio	
Complexo	

 Alterar

Tipo | Peso

Simple | [ ]

 Salvar  Cancelar

 Sair

**Cadastro de Projetos**

Projetos: **Configurar Projeto**

Projeto: **10 - Requisição de materias.**

Complex. UC: **Complex. Atores** | **Fatores Ajuste**

Tipo	Número	Fator	Peso	Valor
Técnico	1	Sistema é distribuído	2	
Técnico	2	Performace	1	
Técnico	3	Eficiência para o usuário final	1	
Técnico	4	Processamento interno complexo	1	
Técnico	5	Código deve ser reusável	1	
Técnico	6	Fácil para instalar	0,50	
Técnico	7	Fácil para usar	0,50	

Tipo: Técnico | Número: 2 | Fator: Performace | Peso: 1,00

Valor:  0  1  2  3  4  5

Identificar os objetivos de performance da aplicação estabelecidos e aprovadas pelo usuário.  
 Zero: nenhuma exigência de performance foi fixada;  
 1: O tempo limite (deadline) do processo é o próximo dia útil;  
 2: O tempo limite (deadline) do processo é de poucas horas (máximo 4 horas);  
 3: O tempo limite (deadline) do processo é de poucos minutos (máximo 60 minutos);  
 4: O tempo limite (deadline) do processo é de poucos segundos (máximo 60 segundos);  
 5: O tempo limite (deadline) do processo é de poucos segundos (máximo 60 segundos) e existe.

## ANEXO 8 – Protótipos UC Importar/Informar Atores e Casos de Uso e sua Complex.

**Estimativa** \_ \_ X

**Projetos Pendentes**  Exibir Projetos já Finalizados

Projeto	Descrição	Dt. Início	Dt. Fim	Imp. UC
10	Requisição de materias.	01/01/2012		Sim

Fases do Projeto

Seq.	Fase	Descrição	Realiza Estimativa	Estimativa Criada
1	Entender o Escopo	Levantamento inicial de requisitos e UC descritivos/...	Não	Não
2	Estimar Tempo (p/ Orçamento)	Estimativa de tempo para aprovação do projeto.	Sim	Não
3	Análise Detalhada - Revisão I	Revisão da estimativa realizada durante a elaboraçã...	Sim	Não
4	Análise Detalhada - Revisão II	Revisão da estimativa realizada ao final da elaboraç...	Sim	Não
5	Construção e Testes	Desenvolvimento e validação do projeto.	Não	Não
6	Liberação do Projeto	Liberação do projeto.	Não	Não

**Nova Estimativa** \_ \_ X

Projeto: 10 - Requisição de materias.

Fase: 2 - Estimar Tempo (p/ Orçamento)

**Estimativa**

Descrição: Estimativa 1

Data: 01/01/2012

**Importação de Casos de Uso**

Base de UC para Importação

BASE PROJETO 1

**Configurar Estimativa**

Projeto: 10 - Requisição de materias.  
 Fase: 2 - Estimar Tempo (p/ Orçamento)  
 Estimativa: Estimativa 1  
 Data: 01/01/2012

Casos de Uso | Atores

Caso de Uso	Qtde. Trans.	Complex. Trans.	Qtde. Ent.	Complex. Ent.

Inserir   
 Alterar   
 Excluir

Caso de Uso:

Transações: Qtde.  Complex.    
 Entidades: Qtde.  Complex.

Salvar   
 Cancelar

Sair

**Configurar Estimativa**

Projeto: 10 - Requisição de materias.  
 Fase: 2 - Estimar Tempo (p/ Orçamento)  
 Estimativa: Estimativa 1  
 Data: 01/01/2012

Casos de Uso | Atores

Ator	Complexidade

Inserir   
 Alterar   
 Excluir

Ator:    
 Complexidade:

Salvar   
 Cancelar

Sair

## ANEXO 9 – Protótipo UC Calcular/Recalcular UC não Ajustados

**Calcular Estimativa**

Projeto: 10 - Requisição de materias.  
Fase: 2 - Estimar Tempo (p/ Orçamento)  
Estimativa: Estimativa 1  
Data: 01/01/2012

UUCP | UCP

**UC não Ajustados (UUCP)**

Calcular o peso dos UC através da complexidade: Entidades

Peso dos Atores (UAW): 0

Pesos dos Casos de Uso (UUCW): 0

Valor UC não ajustados (UUCP): 0

Tempo do Projeto: 0

Calcular/Recalcular

Sair

## ANEXO 10 – Protótipo UC Calcular/Recalcular UC Ajustados

**Calcular Estimativa**

Projeto: 10 - Requisição de materias.

Fase: 2 - Estimar Tempo (p/ Orçamento)

Estimativa: Estimativa 1

Data: 01/01/2012

UUCP UCP

**UC Ajustados (UCP)**

Valor dos Fatores de Complexidade Técnica (TCF) 0

Valor dos Fatores de Complexidade Ambiental (ECF) 0

Valor dos UC Ajustados (UCP) 0

Tempo do Projeto 0

Calcular/Recalcular

Sair

## ANEXO 11 – Protótipo UC Finalizar o Projeto

Finalizar Projeto

Projetos Pendentes

Projeto	Descrição	Dt. Início	Imp. UC
10	Requisição de materias.	01/01/2012	Sim

Finalizar

Tempo Realizado  Dt. Término

Salvar Cancelar

Sair

## ANEXO 12 – Protótipo UC Comparar Estimativas

**Comparar Estimativas**

Exibir Somente Projetos Finalizados

Projeto	Descrição
10	Requisição de materias.

Sel.	Seq.	Fase	Descrição
<input checked="" type="checkbox"/>	2	Estimar Tempo (p/ Orçamento)	Estimativa de tempo para aprovação do projeto.
<input checked="" type="checkbox"/>	3	Análise Detalhada - Revisão I	Revisão da estimativa realizada durante a elaboração da análise detalhada.
<input type="checkbox"/>	4	Análise Detalhada - Revisão II	Revisão da estimativa realizada ao final da elaboração da análise detalhada.

Comparar Estimativas    Sair

**Comparativo**

Projeto

Tempo Realizado

---

Casos de Uso não Ajustados

Seq.	Fase	Peso UC	Peso Atores	UUCP	Tempo UUCP
------	------	---------	-------------	------	------------

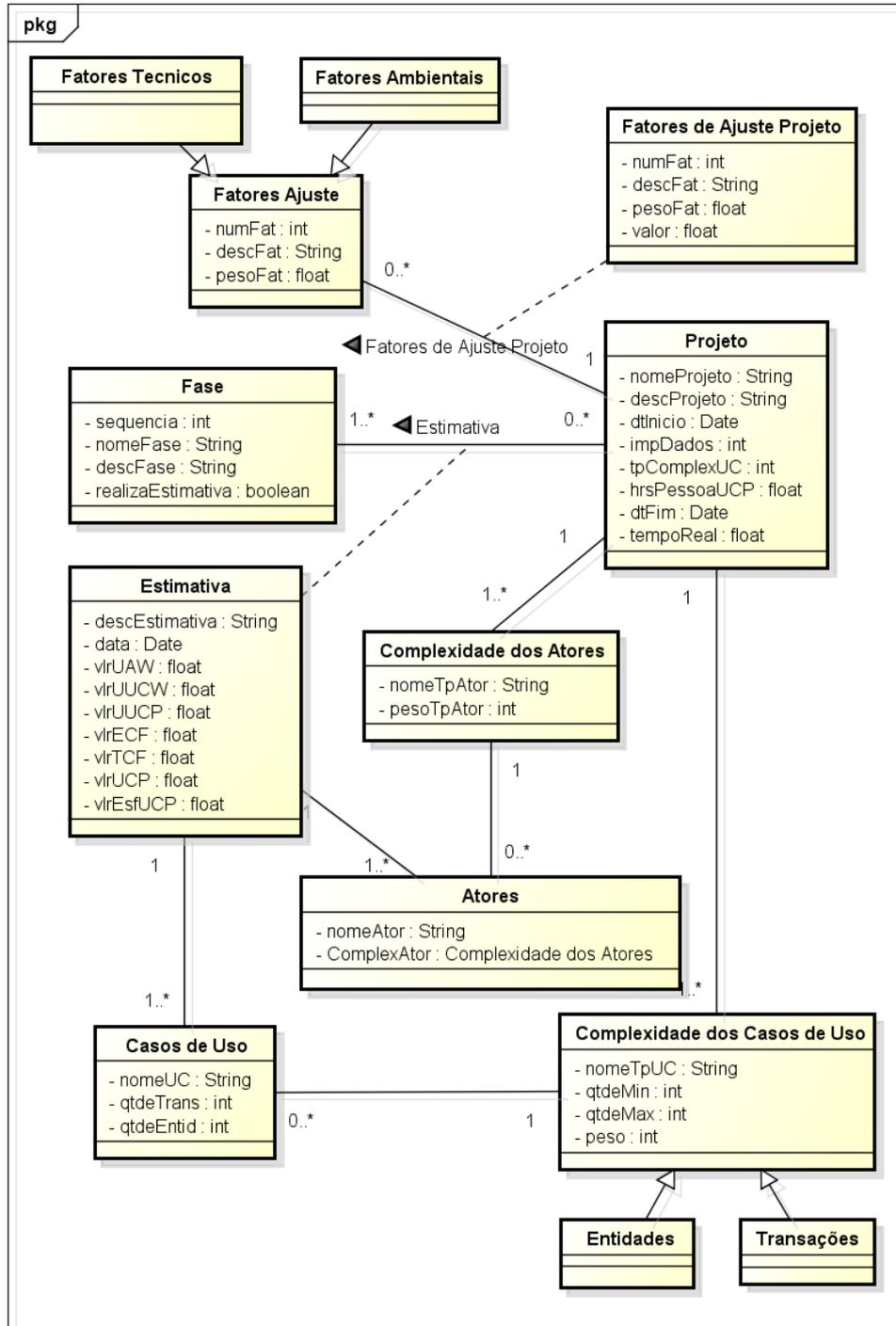
---

Casos de Uso Ajustados

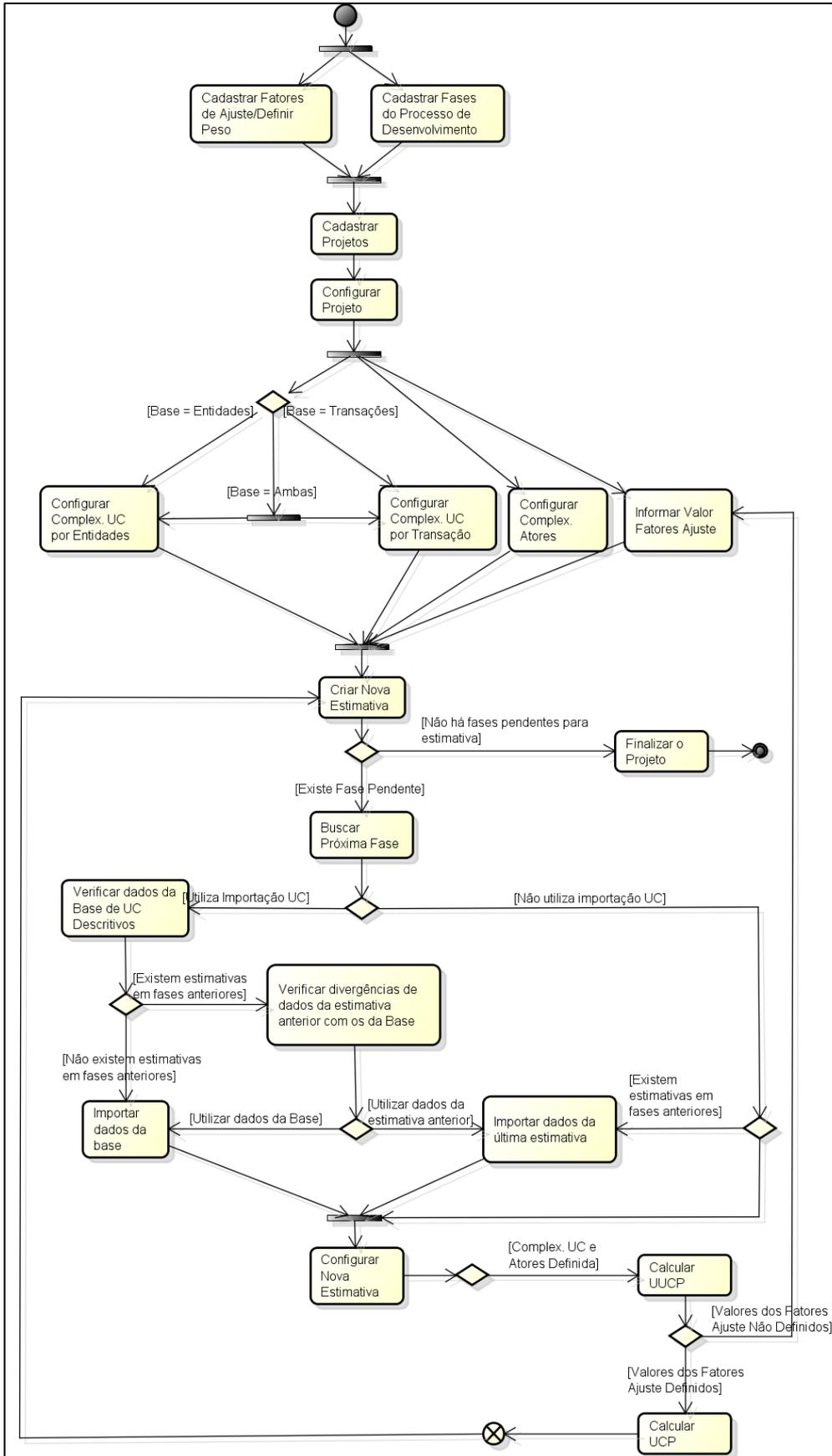
Seq.	Fase	Complex. Fat. Téc.	Complex. Fat. Amb.	UCP	Tempo UCP
------	------	--------------------	--------------------	-----	-----------

Consultar Estimativa    Sair

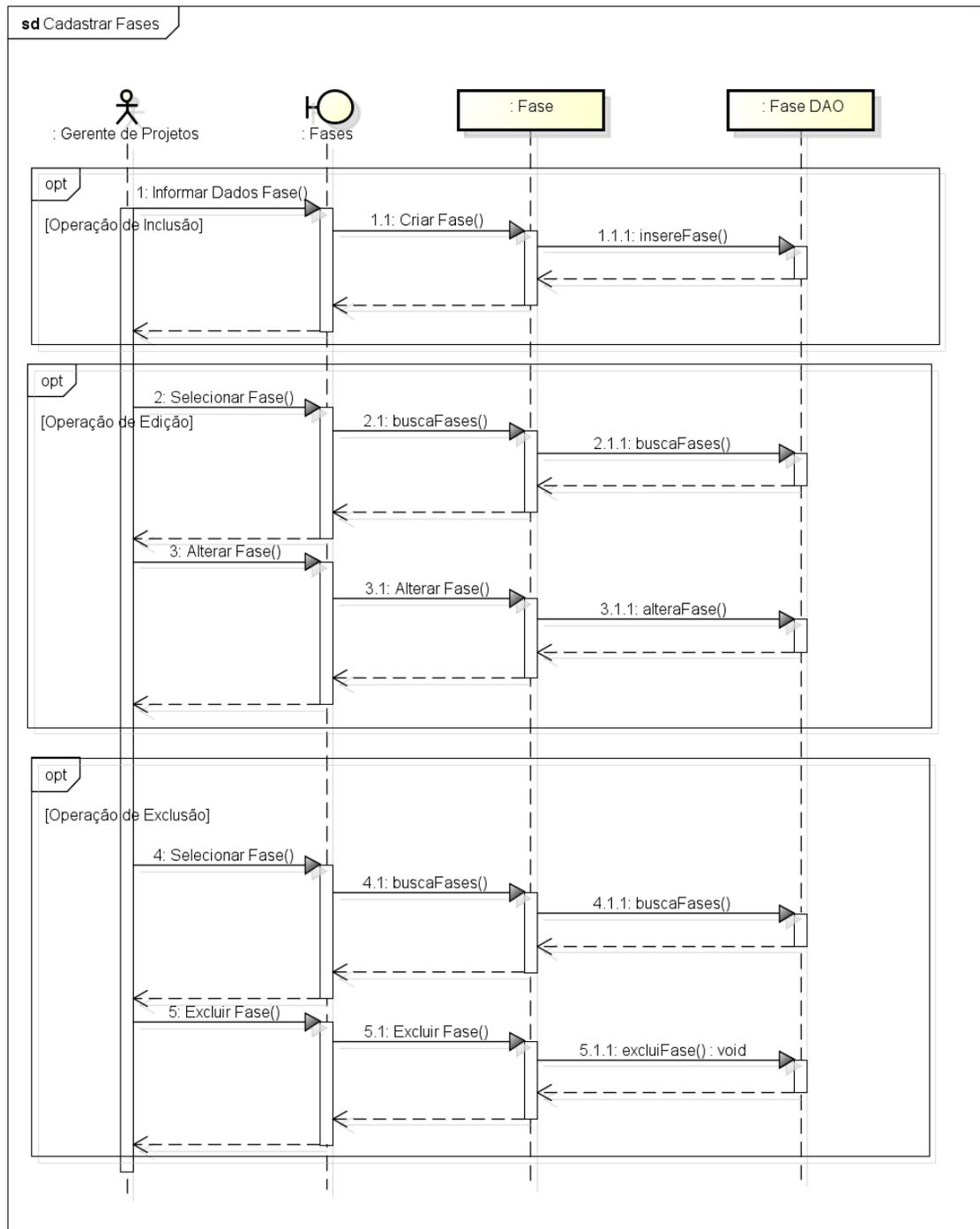
## ANEXO 13 – Diagrama de Classes Conceitual



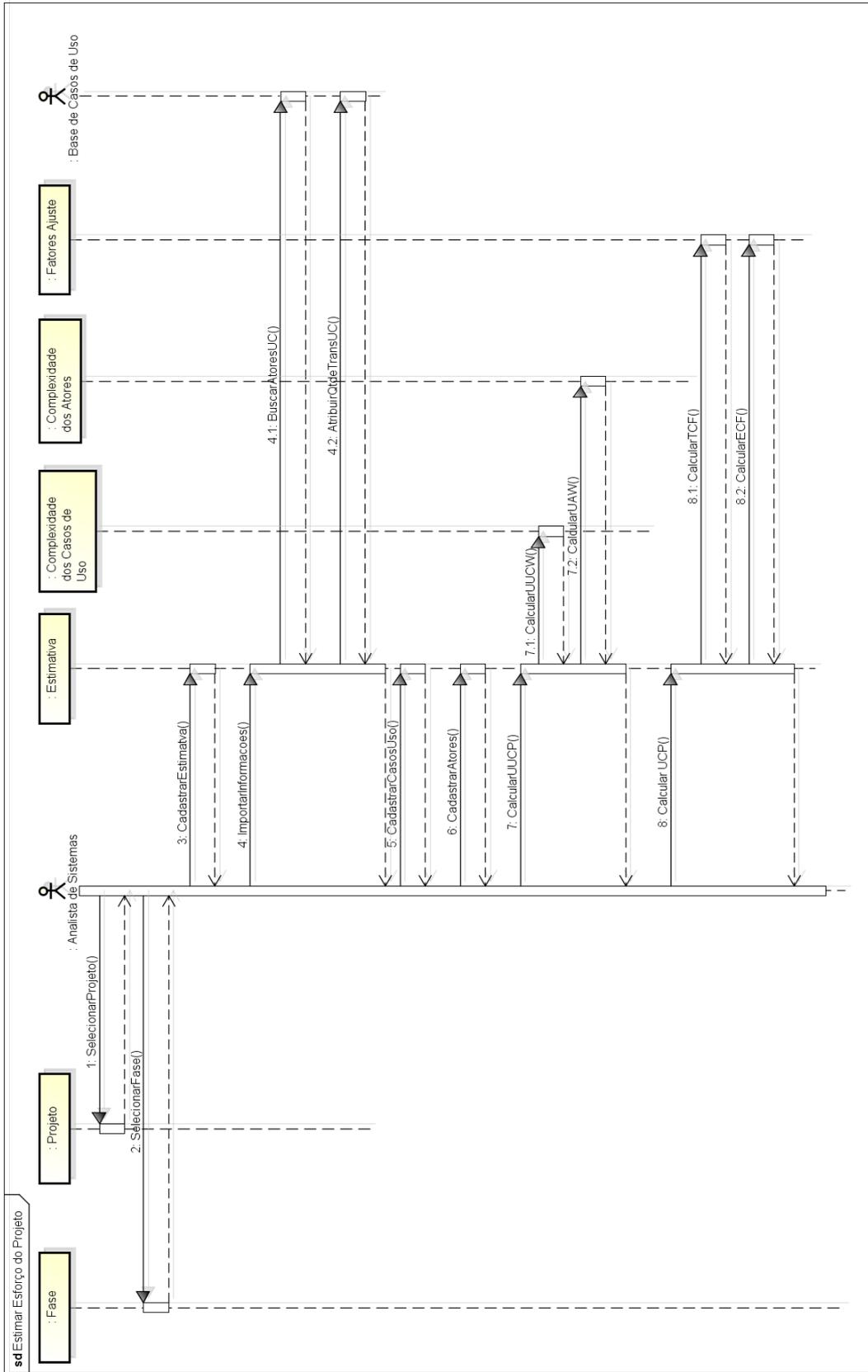
# ANEXO 14 - Diagrama de Atividades



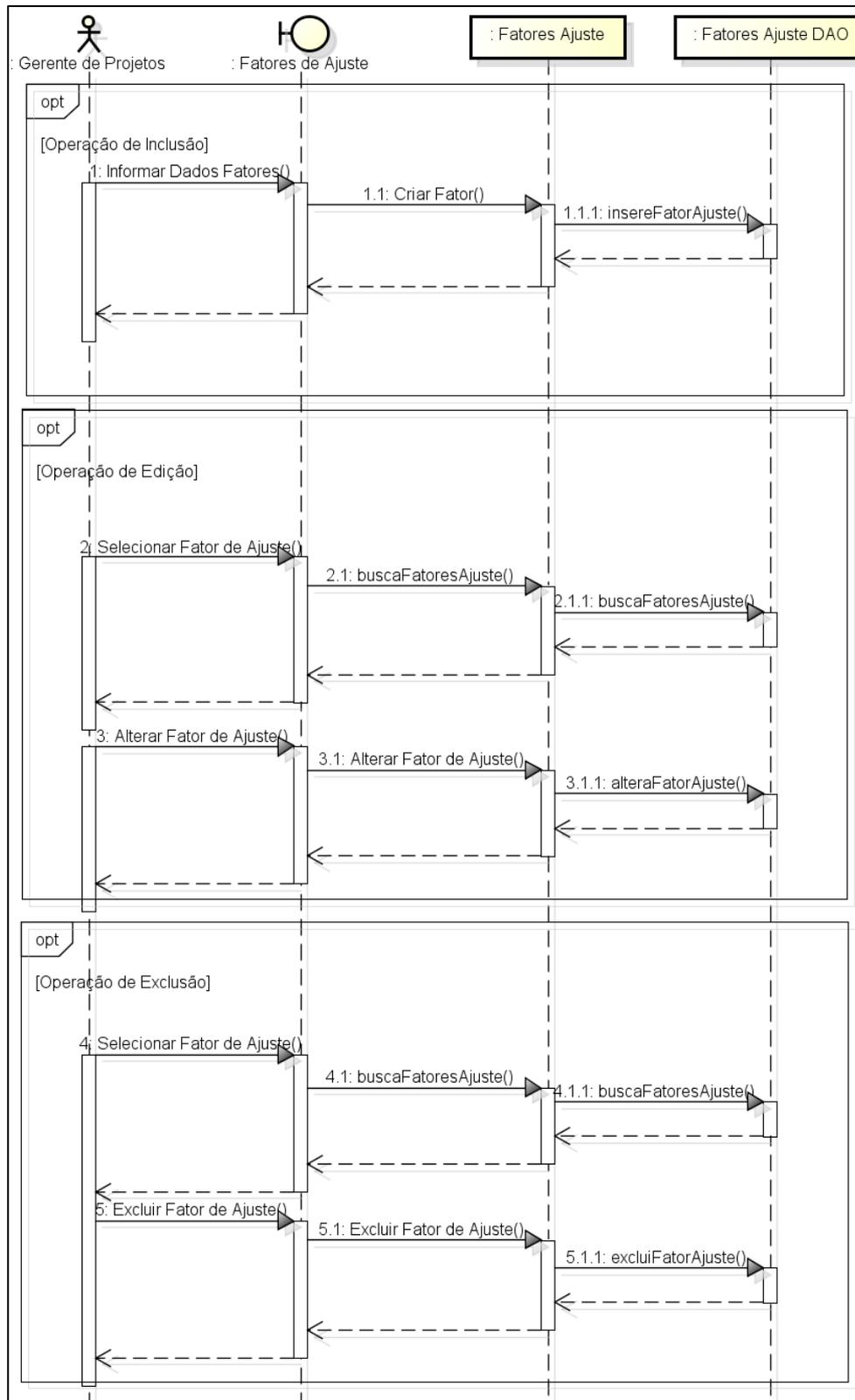
## ANEXO 15 - Diagrama de sequência: Cadastrar fases



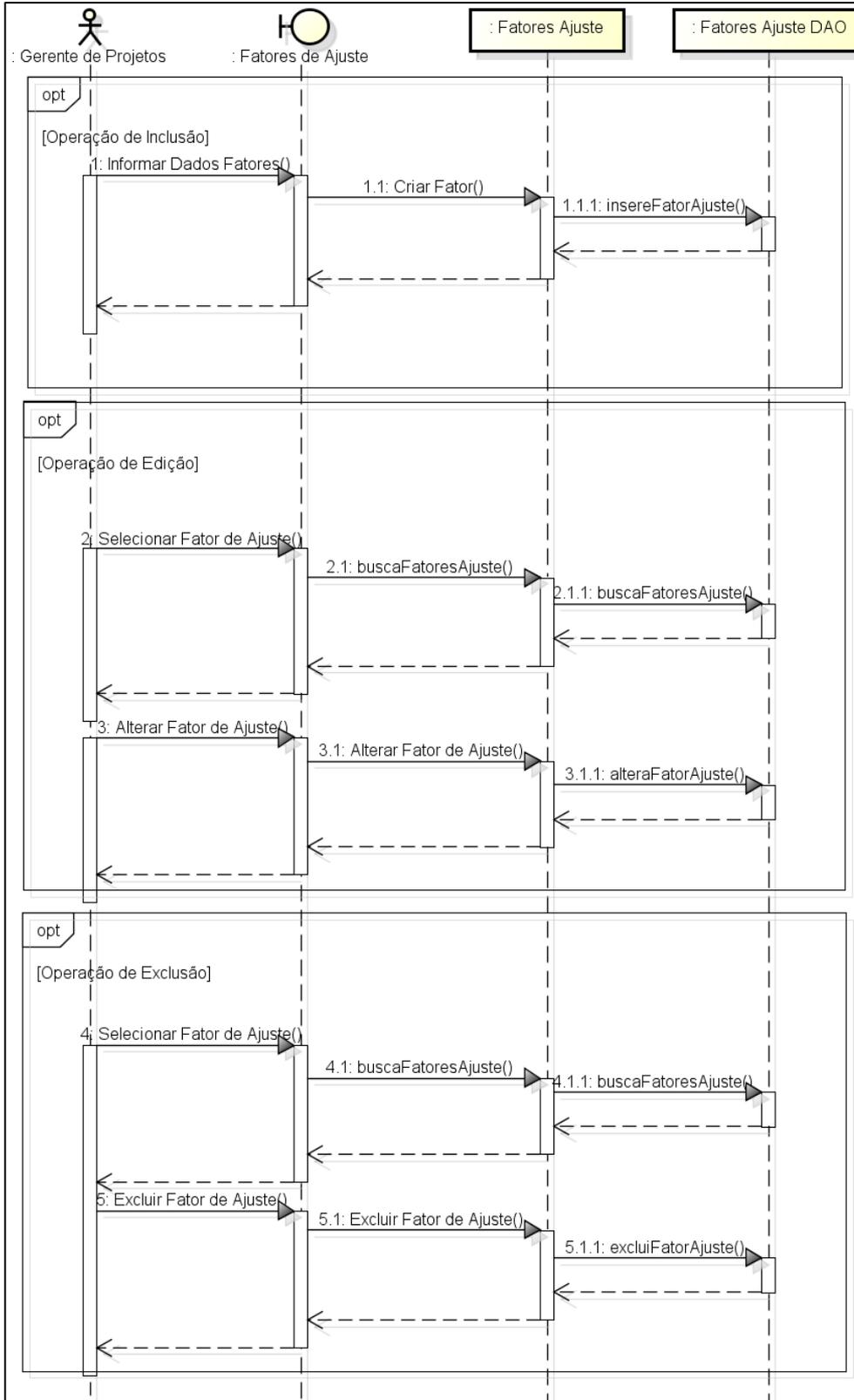
# ANEXO 16 – Diagrama de sequência: Estimar esforço do projeto



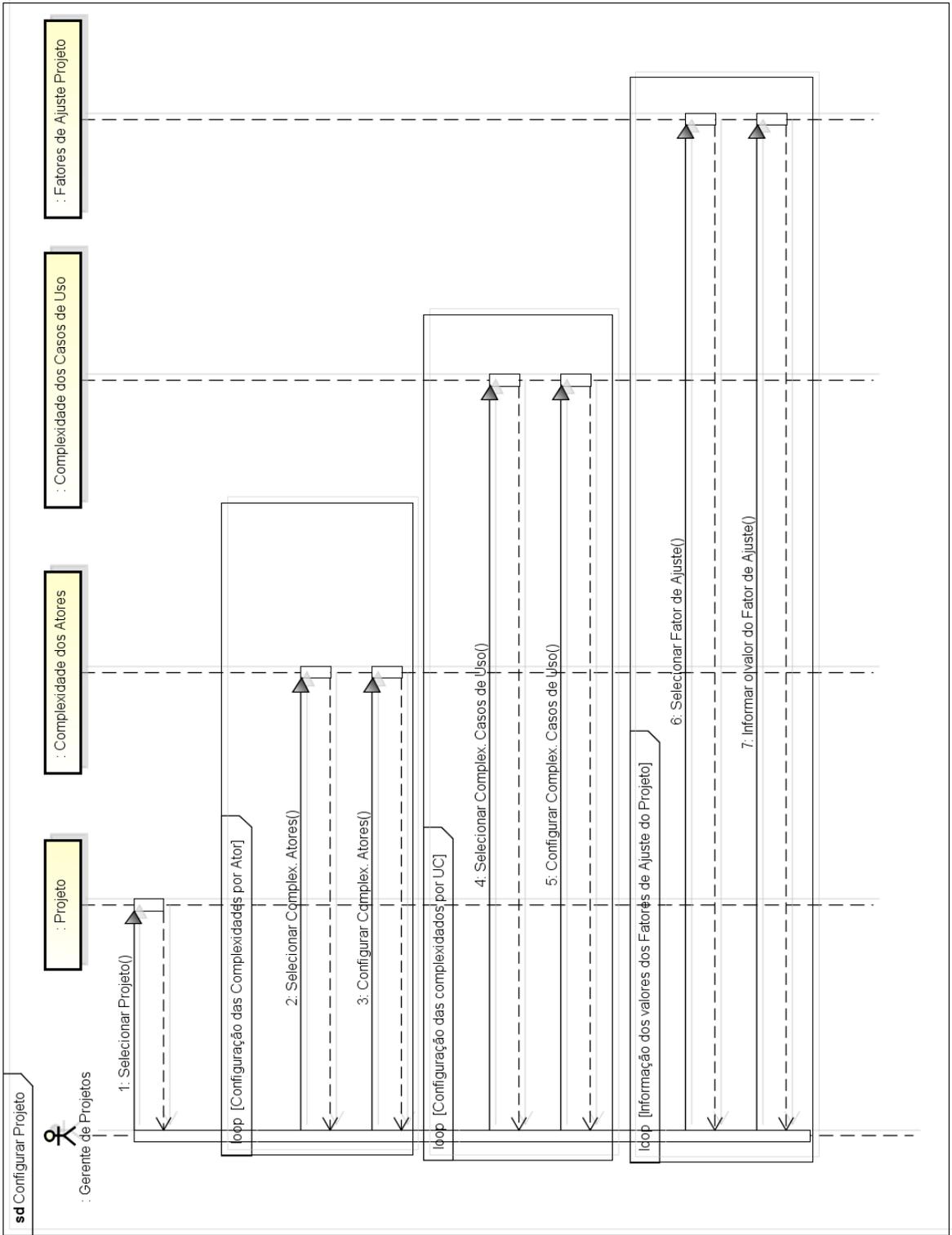
## ANEXO 17 - Diagrama de Sequência: Cadastrar Fatores de Ajuste



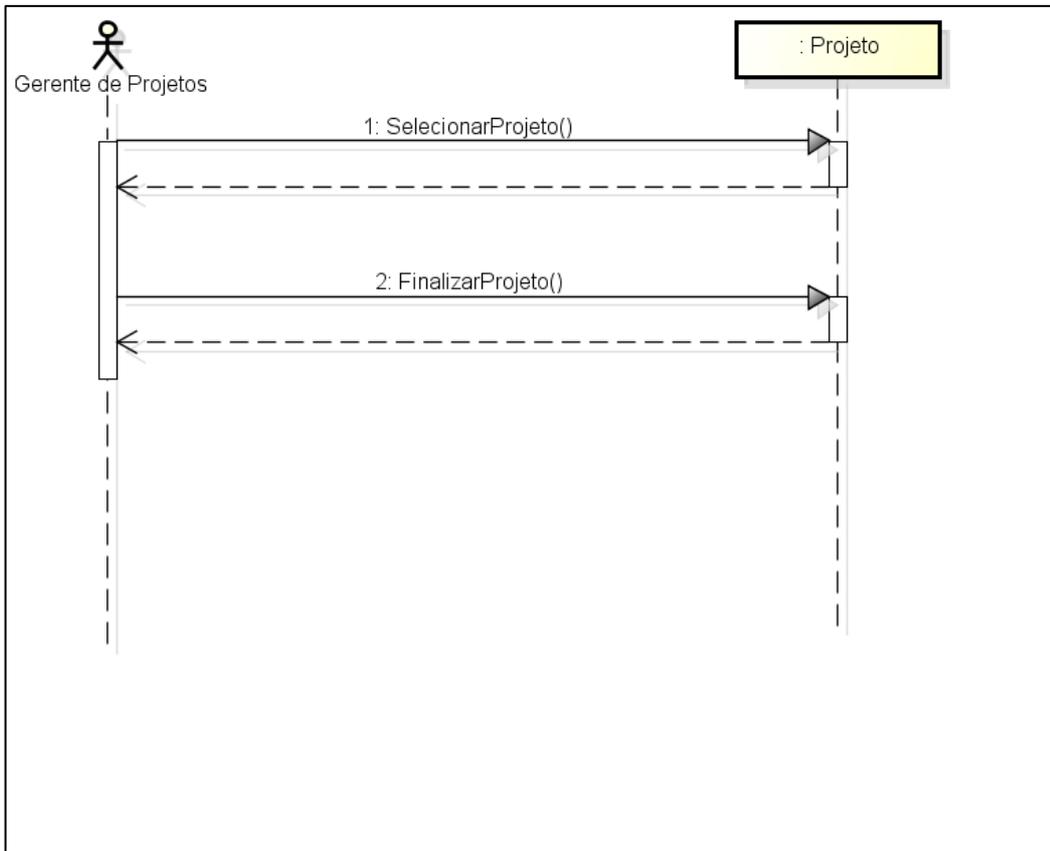
## ANEXO 18 – Diagrama de Sequência: Cadastrar Projetos



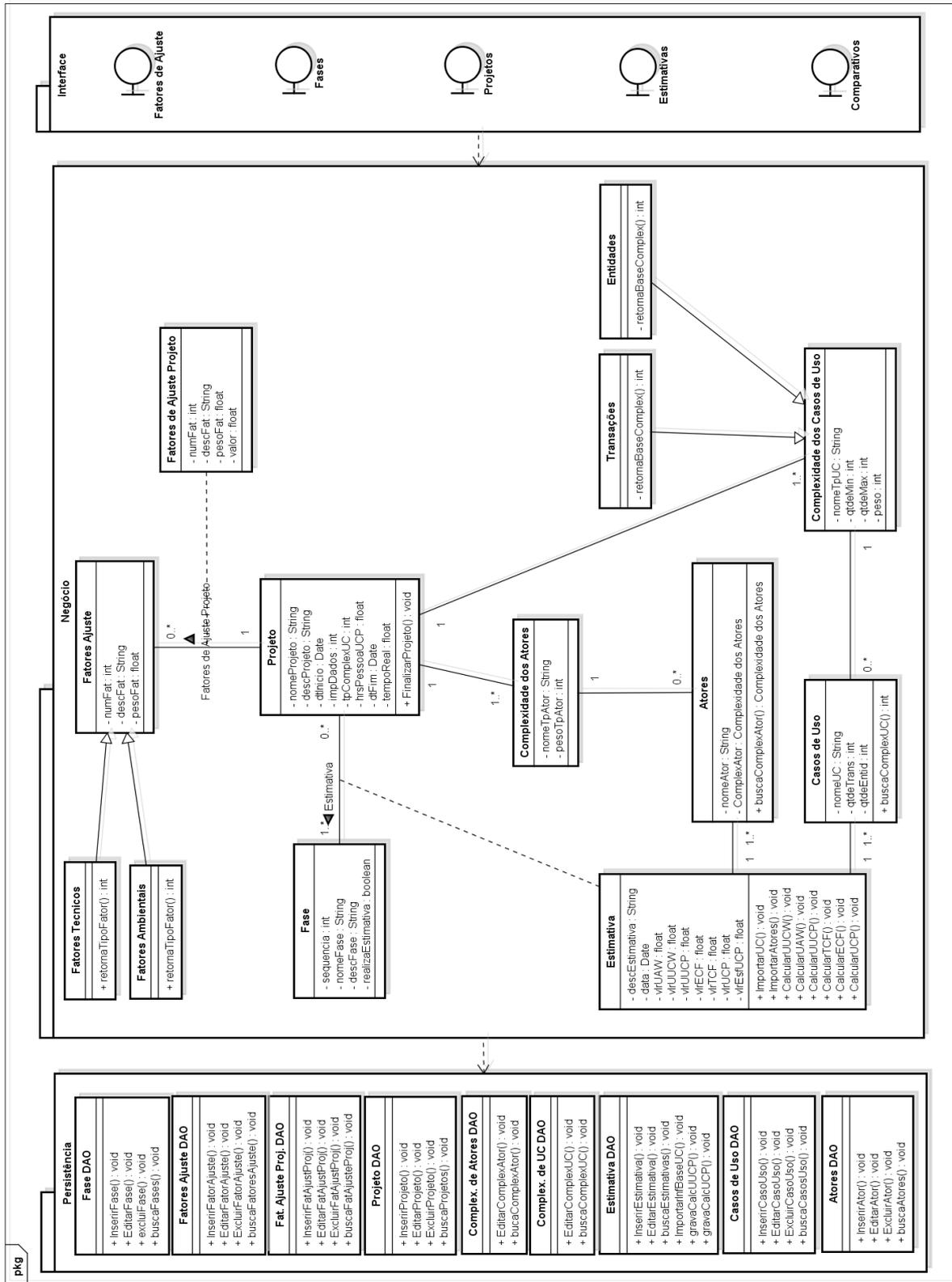
**ANEXO 19 – Diagrama de Sequência: Configurar Projeto**



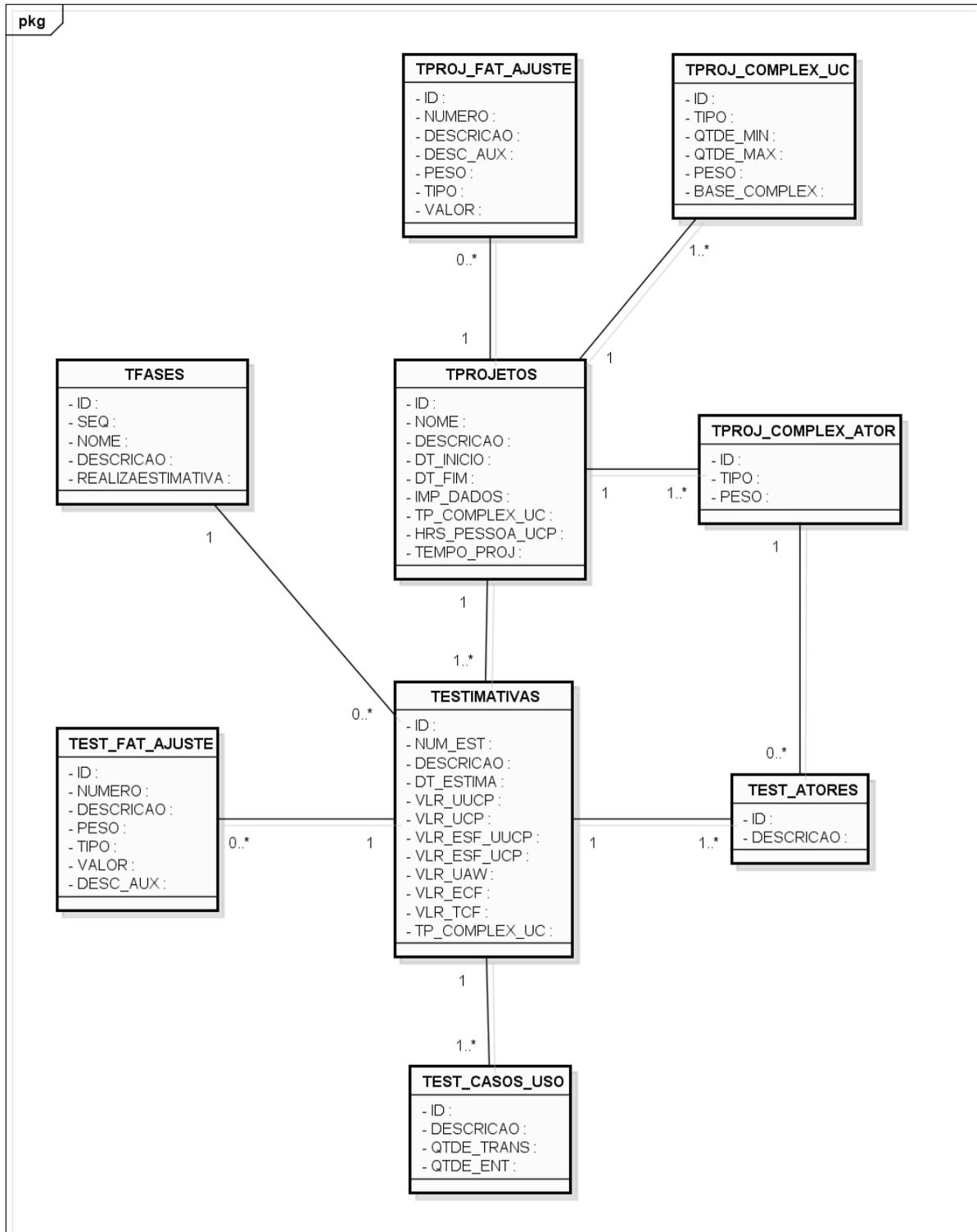
## ANEXO 20 – Diagrama de Sequência: Finalizar Projeto



# ANEXO 21 – Diagrama de Classes de Projeto



## ANEXO 22 – Diagrama ER



## **ANEXO 23 – Documento de Estimativa do Projeto de Requisição de Materiais**

### **PROJETO REFERENTE AO PROCESSO DE REQUISIÇÃO DE MATERIAIS –189 horas**

Tempo utilizado para Estimativa de Horas: 6h30min

Estimativa do projeto contempla:

**Entendimento Geral do Escopo – 8 horas**

**Análise – 40 horas**

**Elaboração da Especificação Geral: 16 horas**

**Desenvolvimento:**

**1ª Fase – 30 horas**

- Criação de Tabelas e Scripts: 3 horas
- Programa: Geração de Solicitação (15horas)
- Programa: Requisição de Materiais (12 horas) Estimativa Final com Ajustes de Testes – Acordado em Reunião

**2ª Fase – 65 horas**

- Programa: Cancelamento de Solicitação de Materiais (25 horas)
- Programa: Alteração de Solicitação de Materiais (16horas)
- Programa: Listagem de Requisições de Materiais (7horas)
- Relatório: Listagem de Requisições de Materiais (16horas)
- Outros (1hora)

**Testes – 30 horas**

**Documentação – Não Estimado**

## ANEXO 24 – Caso de Uso: Gerar Solicitação de Requisição de Materiais

<p>1. Nome do Caso de uso: Gerar Solicitação de Requisição de Materiais</p> <p>1.1. Descrição Resumida</p> <p>A geração de solicitações de materiais deve permitir a geração de mais de uma solicitação para uma mesma ordem de fabricação. Este processo é necessário, pois podem existir itens com almoxarifados padrão diferentes na mesma ordem de fabricação. Portanto, devem ser geradas solicitações de requisição de materiais diferentes para cada almoxarifado para facilitar o processo de separação de peças, pois existem diferentes responsáveis por almoxarifado.</p> <p>Para as solicitações avulsas também deve permitir a geração de mais de uma solicitação para uma mesma solicitação, sendo possível informar um local de entrega, ou seja, um texto livre onde seja possível informar o local onde serão entregues os materiais da solicitação de requisição de materiais.</p> <p>1.2. Atores</p> <p>1.2.1. Planejador da Fábrica;</p>
<p>2. Fluxos de Eventos</p> <p>2.1. Fluxo Básico</p> <p>2.1.1. O planejador seleciona a solicitação a partir de uma ordem de fabricação e informa a ordem e o código do funcionário solicitante;</p> <p>2.1.2. O planejador informa os itens que deseja fazer a solicitação de requisição;</p> <p>2.1.3. O sistema gera as solicitações apresentando uma mensagem com os números das mesmas. Caso, os itens selecionados tenham almoxarifados padrões diferentes, será gerada uma solicitação por almoxarifado padrão.</p> <p>2.2. Fluxos Alternativos</p> <p>1.1.1. No passo 1, se a solicitação for avulsa o planejador informa o centro de custo e local de entrega;</p>
<p>3. Requisitos Especiais</p>
<p>4. Pré-condições</p> <p>4.1. Ordem de fabricação/ Centro de custo e funcionário solicitante cadastrados no sistema.</p>
<p>5. Pós-condições</p> <p>5.1. Solicitação de requisição de materiais gerada.</p>
<p>6. Pontos de Extensão</p>

## **ANEXO 25 – Caso de Uso: Listagem de Solicitações de Requisição de Materiais**

<p>1. Nome do Caso de uso: Listagem de Solicitações de Requisição de Materiais.</p> <p>1.1. Descrição Resumida</p> <p>Permite que o usuário da expedição emita o relatório de separação do seu almoxarifado através do programa de Listagem de Solicitações de Requisição e separe os materiais anotando no relatório a quantidade separada;</p> <p>1.2. Atores</p> <p>1.2.1. Usuário da expedição;</p>
<p>2. Fluxos de Eventos</p> <p>2.1. Fluxo Básico</p> <p>2.1.1. O usuário da expedição informa o seu almoxarifado;</p> <p>2.1.2. O usuário da expedição seleciona a solicitação que deseja listar;</p> <p>2.1.3. O usuário da expedição imprime a solicitação de requisição de materiais e separa os materiais anotando no relatório as quantidades separadas.</p> <p>2.2. Fluxos Alternativos</p>
<p>3. Requisitos Especiais</p>
<p>4. Pré-condições</p> <p>4.1. Solicitação de requisição de materiais avulsa gerada.</p>
<p>5. Pós-condições</p> <p>5.1. Material solicitado separado;</p>
<p>6. Pontos de Extensão</p>

## ANEXO 26 – Caso de Uso: Requisição de Materiais a partir de uma Solicitação

<p>1. Nome do Caso de uso: Requisição de Materiais a partir de uma Solicitação</p> <p>1.1. Descrição Resumida Permite que o usuário da expedição faça a requisição dos materiais separados podendo liberar os mesmos e entregar no local onde foram solicitados;</p> <p>1.2. Atores 1.2.1. Usuário da expedição;</p>
<p>2. Fluxos de Eventos</p> <p>2.1. Fluxo Básico</p> <p>2.1.1. O sistema exibe as solicitações de requisição de materiais pendentes;</p> <p>2.1.2. O usuário da expedição seleciona a requisição desejada;</p> <p>2.1.3. O sistema exibe os itens da solicitação pendentes para requisição;</p> <p>2.1.4. O usuário da expedição informa os itens que deseja requisitar;</p> <p>2.1.5. O sistema verifica o tipo de solicitação, se for “Ordem” serão geradas duas movimentações de estoque do tipo “TRA”, sendo uma de transferência de saída do almoxarifado padrão do item e uma transferência de entrada no almoxarifado da demanda da ordem (local onde os materiais devem ser depositados).</p> <p>2.2. Fluxos Alternativos</p> <p>2.2.1. Quando o tipo de solicitação for “Avulsa” será gerada uma movimentação de estoque do tipo “REQ”. As quantidades das movimentações de estoque é a quantidade informada no campo “Qtd. Ent.”.</p>
<p>3. Requisitos Especiais</p>
<p>4. Pré-condições</p> <p>4.1. Material solicitado separado.</p>
<p>5. Pós-condições</p> <p>5.1. Material entregue para o local informado.</p>
<p>6. Pontos de Extensão</p>

## ANEXO 27 – Caso de Uso: Cancelamento de Solicitações de Requisição de Materiais

<p>1. Nome do Caso de uso: Cancelamento de Solicitações de Requisição de Materiais</p> <p>1.1. Descrição Resumida</p> <p>Permite que o planejador da fábrica cancele itens das solicitações de requisição de Materiais.</p> <p>1.2. Atores</p> <p>1.2.1. Planejador Fábrica;</p>
<p>2. Fluxos de Eventos</p> <p>2.1. Fluxo Básico</p> <p>2.1.1. O planejador seleciona a solicitação que deseja cancelar;</p> <p>2.1.2. O planejador seleciona os itens que deseja cancelar;</p> <p>2.1.3. O sistema cancela a solicitação de requisição do item selecionado;</p> <p>2.2. Fluxos Alternativos</p> <p>2.2.1. Se já tiver sido entregue alguma quantidade quando o tipo de solicitação for “Ordem” o sistema deverá gerar duas movimentações de estoque do tipo “TRA”, sendo uma de transferência de saída do almoxarifado da demanda da ordem (local onde os materiais devem ser retirados) e uma transferência de entrada no almoxarifado padrão do item (local onde os materiais devem ser depositados).</p> <p>2.2.2. Se já tiver sido entregue alguma quantidade quando o tipo de solicitação for “Avulsa” será gerada uma movimentação de estoque do tipo “DEV”.</p>
<p>3. Requisitos Especiais</p>
<p>4. Pré-condições</p> <p>4.1. Solicitação de requisição de materiais gerada.</p>
<p>5. Pós-condições</p> <p>5.1. Solicitação de requisição de materiais cancelada;</p>
<p>6. Pontos de Extensão</p>

## ANEXO 28 – Caso de Uso: Alteração de Solicitação de Materiais

<p>1. Nome do Caso de uso: Alteração de Solicitação de Materiais</p> <p>1.1. Descrição Resumida</p> <p>Permite que o planejador da fábrica altere a quantidade de itens das solicitações de requisição de Materiais.</p> <p>1.2. Atores</p> <p>1.2.1. Planejador fábrica;</p>
<p>2. Fluxos de Eventos</p> <p>2.1. Fluxo Básico</p> <p>2.1.1. O planejador seleciona a ordem que originou a solicitação de materiais que deseja alterar;</p> <p>2.1.2. O sistema exibe todas as solicitações de requisição de materiais pendentes referentes à ordem selecionada.</p> <p>2.1.3. O planejador seleciona a solicitação de materiais que deseja alterar informando a nova quantidade;</p> <p>2.1.4. O sistema deverá validar se a soma das quantidades já requisitadas da solicitação é menor do que a quantidade informada para alteração;</p> <p>2.1.5. Alterar a solicitação de requisição de materiais conforme a quantidade informada.</p> <p>2.2. Fluxos Alternativos</p> <p>2.2.1. No passo 4, se a validação não for verdadeira o sistema deverá exibir uma mensagem informando ao usuário que não é possível realizar tal alteração.</p>
<p>3. Requisitos Especiais</p> <p>3.1. Somente é permitido alterar solicitações de requisição de materiais geradas a partir de ordens de fabricação.</p>
<p>4. Pré-condições</p> <p>4.1. Solicitação de requisição de materiais gerada.</p>
<p>5. Pós-condições</p> <p>5.1. Solicitação de requisição de materiais alterada;</p>
<p>6. Pontos de Extensão</p>