

Um Estudo Experimental em Modelagem de Requisitos de Sistemas de Informação

Vitor Padilha Gonçalves

Universidade Federal de Juiz de Fora
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Bacharelado em Ciência da Computação

Orientador: Prof. Michel Heluey Fortuna



Juiz de Fora, MG
Dezembro de 2007

Um Estudo Experimental em Modelagem de Requisitos de Sistemas da Informação

Vitor Padilha Gonçalves

Monografia submetida ao corpo docente do Departamento de Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Juiz de Fora, como parte integrante dos requisitos necessários para a obtenção do grau de Bacharel em Ciência da Computação.

Aprovada pela banca constituída pelos seguintes professores:

Prof. Michel Heluey Fortuna – orientador

MSc. Engenharia de Sistemas/Informática, IME/RJ, 1982

Profa. Jesuliana Nascimento Ulysses

MSc. em Computação, UFF, 2003

Prof. Tarcísio de Souza Lima

MSc. em Informática, PUC/RJ, 1988

Juiz de Fora, MG
Dezembro de 2007

Agradecimentos

A todos que colaboraram na realização deste estudo, principalmente aos participantes (Carol, Fábio Campos, Gilmar Pedretti, Marcelo, Rafael e Vinícius), e a meu orientador que esteve sempre presente na condução deste trabalho.

Sumário

Lista de Reduções	vi
Lista de Figuras	vii
Lista de Tabelas	viii
Resumo	ix
Capítulo 1 - Introdução.....	1
Capítulo 2 – Engenharia de Software Experimental.....	3
2.1. Objetivos de Estudos Experimentais em Engenharia de Software	3
2.2. Dificuldades de Experimentos no Contexto da Engenharia de Software	3
2.3. Tipos de Estudos Experimentais	4
2.3.1. Pesquisa de Opinião (<i>Surveys</i>)	4
2.3.2. Estudo de Caso	5
2.3.3. Experimento.....	5
2.3.4. Comparação entre os Tipos de Estudos Experimentais.....	6
2.4. Elementos de um Estudo Experimental	6
2.5. O Processo de Experimentação	8
2.5.1. Definição.....	9
2.5.2. Planejamento.....	9
2.5.3. Operação	13
2.5.4. Análise	13
2.5.5. Empacotamento	14
Capítulo 3 – Modelagem de Requisitos com Casos de Uso: Abordagem Tradicional e Abordagem Informacional.....	15
3.1. Modelagem de Sistemas com Casos de Uso	15
3.2. Modelagem Informacional de Requisitos	19
3.2.1. Modelo Integrado de Requisitos.....	19
3.2.2. Derivação do Modelo de Classes do Domínio	22
Capítulo 4 – O Experimento	26
4.1. Definição do Experimento	26
4.1.1. Objetivo do Estudo Experimental.....	26
4.1.2. Questões e Métricas.....	27
4.2. Planejamento do Experimento	30

4.2.1. Seleção do Contexto	31
4.2.2. Seleção dos Participantes.....	31
4.2.3. Seleção das Variáveis	32
4.2.4. Formulação das Hipóteses	33
4.2.4.1. Hipóteses para Granularidade	33
4.2.4.2. Hipóteses para Uniformidade de Abstrações	34
4.2.4.3. Hipóteses para Uniformidade de Associações	35
4.2.4.4. Hipóteses para Uniformidade de Atributos	35
4.2.4.5. Hipóteses para Uniformidade de Operações	36
4.2.4.6. Hipóteses para Uniformidade Representacional	37
4.2.5. Design do experimento	38
4.2.6. Descrição da instrumentação	38
4.2.7. Validade	39
4.2.7.1. Validade Interna.....	39
4.2.7.2. Validade de Construção.....	40
4.2.7.3. Validade de Conclusão.....	40
4.2.7.4. Validade Externa.....	41
4.3. Execução do Experimento.....	41
4.4. Análise do Experimento	42
4.4.1. Análise das Ameaças	42
4.4.2. Teste da Hipótese de Granularidade dos Modelos	43
4.4.3. Teste das Hipóteses de Uniformidade entre Modelos	45
4.4.4. Conclusão Geral sobre os Resultados do Experimento	51
Capítulo 5 - Conclusão	52
Referências Bibliográficas	53
Anexo A - Questionário de Caracterização dos Participantes	55
Anexo B - Sumário do Sistema Desenvolvido (FGV)	56
Anexo C - Modelo de Casos de Uso do Participante P3	58
Anexo D - Modelo de Classes de Domínio do Participante P3	67
Anexo E - Modelo de Casos de Uso do Participante P4	68
Anexo F - Modelo de Classes de Domínio do Participante P4.....	73

Lista de Reduções

ES	Engenharia de Software
ESE	Engenharia de Software Experimental
GQM	Goal/Question/Metric
II	Interface Informacional (de um UC)
MIR	Modelagem Informacional de Requisitos
MOD	Modelo de (classes de) Objetos de Domínio
NIO	Nível Informacional de Objetivos
UC	<i>Use Case</i>
UML	<i>Unified Modeling Language</i>

Lista de Figuras

Figura 2.1 – Elementos de um experimento	7
Figura 2.2 – Princípios de um experimento	8
Figura 2.3 – O processo experimental	9
Figura 2.4 – Estrutura hierárquica de GQM	10
Figura 2.5 – Princípios e tipos de validade de um experimento	13
Figura 3.1 – Exemplos de UC's de um sistema de gerência de restaurante	18
Figura 3.2 – Especificação da II de três UC's do sistema de gerência de restaurante	22
Figura 3.3 – Especificação dos fluxos do sistema <i>Restaurante</i>	24
Figura 3.4 – Diagrama de classes parcial, do sistema <i>Restaurante</i>	25
Figura 3.5 – Diagrama de classes do sistema <i>Restaurante</i>	26
Figura 4.1 – Ilustração dos elementos da métrica de uniformidade de abstrações, entre dois modelos de classes	29
Figura 4.2 – Ilustração dos elementos da métrica de uniformidade de atributos, entre dois modelos de classes	30
Figura 4.3 – Número de classes no modelo de cada participante	45

Lista de Tabelas

Tabela 2.1 – A comparação de estratégias empíricas	6
Tabela 3.1 – Glossário de termos do sistema de gerência de restaurante	19
Tabela 3.2 – Símbolos utilizados na linguagem de especificação da II dos UC's.....	21
Tabela 3.3 – Dicionário de itens elementares	23
Tabela 4.1 – Cronograma de execução do experimento.....	41
Tabela 4.2 – Caracterização dos participantes do experimento.....	43
Tabela 4.3 – Número de classes por participante	45
Tabela 4.4 – Abstrações nos modelos	46
Tabela 4.5 – Associações nos modelos	47
Tabela 4.6 – Atributos nos modelos	48
Tabela 4.7 – Operações nos modelos	49
Tabela 4.8 – Operações nos modelos, excluídas as operações construtoras	49
Tabela 4.9 – Uniformidade de associações, atributos, operações e representacional	50
Tabela 4.10 – Comparação das médias de uniformidade	51

Resumo

O presente trabalho apresenta um experimento comparativo entre duas técnicas de modelagem de requisitos de sistemas computacionais com casos de uso: a técnica tradicional e uma especialização desta, denominada Modelagem Informacional de Requisitos. São apresentados alguns conceitos sobre essas duas técnicas e sobre a Engenharia de Software Experimental, relevantes para o trabalho. O experimento realizado testou algumas hipóteses sobre a granularidade e a uniformidade dos modelos de classes de domínio, produzidos com cada técnica, gerando evidências de que, por exemplo, os modelos produzidos com a técnica de Modelagem Informacional de Requisitos, por diferentes modeladores, para um mesmo sistema, são mais uniformes do que aqueles produzidos com a técnica tradicional de casos de uso. Esse e outros resultados são discutidos à luz das características e peculiaridades de cada técnica.

Palavras-Chave

Engenharia de Software Experimental, Casos de Uso, Modelo de Classes, Modelagem Informacional de Requisitos.

Capítulo 1

Introdução

Ainda é pouco difundida, entre os profissionais e pesquisadores em Engenharia de Software, a importância da experimentação para a validação dos produtos, processos, e técnicas utilizadas na área. Mas a experimentação é de fundamental importância para uma ciência que está sempre inovando e criando ferramentas, e onde muitas vezes algumas afirmações são feitas sem nenhuma prova científica. Recentemente, uma parte significativa destes pesquisadores tem chamado a atenção para a importância de se utilizar estudos experimentais para formação de uma base científica confiável.

No contexto de modelagem de requisitos para sistemas da informação, FORTUNA & BORGES (2005) relatam alguns problemas relacionados à técnica de Casos de Uso (UC's - *Use Cases*) (JACOBSON *et al.*, 1992). Segundo esses autores, os problemas decorrem da ênfase excessiva no detalhamento do comportamento interno do sistema, da falta de uma regra objetiva para orientar os analistas na escolha do nível de abstração a adotar na modelagem, e do insuficiente detalhamento das informações trocadas entre o ambiente e sistema.

Outros pesquisadores têm apontado problemas na integração entre UC's e a modelagem de classes de um sistema. Em particular, SVETINOVIC, BERRY & GODFREY (2005) evidenciaram a dificuldade de se conseguir uniformidade entre os modelos de classes de domínio, produzidos por diferentes modeladores para um mesmo sistema, a partir dos UC's do sistema e seguindo as regras usuais da Análise Orientada a Objetos (COAD & YOURDON, 1990).

Mais recentemente, FORTUNA, WERNER, e BORGES (2007) propuseram uma especialização da técnica de UC's, denominada de *Modelagem Informacional de Requisitos* (MIR). Uma das características dessa técnica é prover um conjunto de regras para a derivação de um diagrama de classes de domínio, a partir dos UC's de um sistema.

O principal objetivo deste trabalho é fazer uma avaliação comparativa das duas técnicas - UC's e MIR, através de um experimento, buscando, principalmente, verificar até onde a MIR é uma solução para o problema mencionado anteriormente, da falta de uniformidade entre modelos de classes. Mais especificamente, serão comparadas a granularidade e a uniformidade dos modelos de classes produzidos nas duas técnicas.

Outro objetivo deste trabalho é prover um exemplo de como um estudo experimental pode ser realizado no contexto da Engenharia de Software, com a finalidade

de subsidiar e motivar novos trabalhos desse tipo. A presente monografia está dividida da seguinte forma: o capítulo 2 aborda uma parte dos conceitos, processos, objetivos, tipologia, e dificuldades, apresentados por alguns autores na área de Engenharia de Software Experimental; o capítulo 3 introduz as duas técnicas de modelagem de requisitos a serem avaliadas no estudo experimental - UC's e MIR; o capítulo 4 descreve todo o processo de experimentação realizado pelo autor desta monografia e seu orientador; finalmente, o capítulo 5 apresenta algumas conclusões e sugestões para trabalhos futuros.

Capítulo 2

Engenharia de Software Experimental

O principal objetivo deste capítulo é dar uma visão de como estudos experimentais podem ser aplicados na Engenharia de Software (ES).

O presente capítulo está baseado, principalmente, em WOHLIN *et al.* (2000), que é atualmente uma das principais referências em Engenharia de Software Experimental (ESE). Serão apresentados os objetivos, dificuldades e elementos de um estudo experimental em ES, bem como os tipos usuais de estudos e o processo que deve ser seguido para realização dos mesmos.

2.1. Objetivos de Estudos Experimentais em Engenharia de Software

Segundo TRAVASSOS, GUROV & AMARAL (2002), o objetivo da engenharia de software experimental é o aprimoramento da Engenharia de Software (ES) a partir da aplicação do método científico (experimentação) na caracterização, avaliação, previsão, controle e melhoria de métodos, produtos, processos e técnicas, que apóiam o desenvolvimento de software.

Além disto, há uma preocupação com o avanço da própria área de experimentação, através do estudo e da pesquisa de novos modelos para o planejamento, execução e empacotamento de estudos experimentais em ES. Estas atividades são fundamentais para que a ES incorpore cada vez mais os princípios de ciência, pois formam uma base teórica de conhecimentos validados (WOHLIN *et al.*, 2000).

Outros objetivos da prática experimental em engenharia de software são:

- A observação e experimentação podem ajudar a abrir novas áreas de investigação dentro da ES;
- A experimentação pode acelerar o processo de desenvolvimento de software, eliminando suposições errôneas, ou antecipando suas conseqüências. (CONRADI, 2001 *apud* TRAVASSOS, GUROV & AMARAL, 2002)

2.2. Dificuldades de Experimentos no Contexto da Engenharia de Software

Na maioria das vezes, o desenvolvimento de software é uma tarefa complexa, podendo demorar muito tempo e ter um alto custo. Por isso, grande é o esforço por parte dos

pesquisadores e dos profissionais da área, na busca de métodos e processos mais eficientes para o desenvolvimento de software. Nos últimos anos, tem crescido a percepção da importância de se lançar mão da experimentação como mais uma forma de evoluir, com segurança, a ES. Entretanto, esta também não é uma tarefa fácil.

BARROS, WERNER & TRAVASSOS (2002) colocam que, apesar do interesse crescente pelos estudos experimentais em ES, ainda não se tem uma base sólida ideal para a realização desses estudos.

JURISTO & MORENO (2000) descrevem também outros problemas particulares da ES, que dificultam a realização de estudos experimentais. São eles:

- Os desenvolvedores de software ainda não dão a devida importância à experimentação, e os engenheiros de software não procuram validar cientificamente suas teorias;
- A formação dos atuais engenheiros e desenvolvedores de software é insuficiente; faltam, por exemplo, conhecimentos específicos de estatística, necessários à análise e interpretação dos resultados de experimentos;
- O fator humano tem grande influência no desenvolvimento de software, o que torna mais complexa a tarefa de realizar experimentos nessa área;
- Poucos pesquisadores se dispõem a repetir estudos experimentais de outros profissionais da área;
- O grande número de técnicas, métodos ou modelos dificultam a escolha daqueles que participarão nos estudos experimentais comparativos.

2.3. Tipos de Estudos Experimentais

Existem três tipos de estudos experimentais: pesquisa de opinião (*survey*), estudo de caso e experimento (ROBSON, 1993 *apud* WOHLIN *et al.*, 2000). As próximas seções descrevem cada um deles.

2.3.1. Pesquisa de Opinião (*Survey*)

A pesquisa de opinião é uma investigação retrospectiva (WOHLIN *et al.*, 2000). Os dados coletados neste tipo de experimentos são referentes a métodos, técnicas ou modelos que já tenham sido utilizados.

A coleta dos dados a serem analisados pode ser feita através de questionários e entrevistas (WOHLIN *et al.*, 2000). As grandes vantagens da pesquisa de opinião é a sua

simplicidade e baixo custo. Como desvantagem existe a subjetividade inerente aos resultados obtidos com base na opinião das pessoas que participam da pesquisa.

Uma pesquisa de opinião tem o objetivo de descrever, explicar ou explorar uma determinada população (BABIE, 1990 *apud* WOHLIN *et al.*, 2000).

2.3.2. Estudo de Caso

O estudo de caso é a observação ou monitoração de projetos e atividades que estão ocorrendo em ambiente real ou ligeiramente adaptado. Ou seja, a interferência do experimentador sobre a realização do projeto ou atividade é pouca ou mesmo nenhuma (estudo de caso observacional). Os dados são coletados seguindo um plano do que deve ser observado no estudo (TRAVASSOS, GUROV & AMARAL, 2002).

Nesse tipo de estudo não se tem muito controle sobre fatores externos, pois tais estudos são realizados no ambiente real de trabalho, onde tais fatores estarão naturalmente atuando sobre o objeto de estudo. Por este motivo, YIN (1994) e STAKE (1995) *apud* WOHLIN *et al.* (2000) ressaltam a importância de se produzir uma documentação sobre os fatores externos que poderiam, de alguma maneira, ter afetado o estudo.

Segundo KITCHENHAM (1995) *apud* WOHLIN *et al.* (2000), o arranjo de um estudo de caso pode ser feito de três maneiras:

- Os resultados obtidos a partir da utilização de novos métodos são comparados com um *baseline*¹;
- Métodos diferentes são usados em projetos semelhantes para comparação dos resultados;
- Projetos da empresa são escolhidos de maneira aleatória para a aplicação do método novo, enquanto outros continuam usando o método antigo.

2.3.3. Experimento

Experimento é o tipo de estudo caracterizado pelo maior controle do experimentador sobre o ambiente, os participantes e o objeto do estudo. Entre outras coisas, o experimentador pode alterar o valor de algumas variáveis de entrada e observar os efeitos dessa mudança sobre aquilo que lhe interessa estudar.

Com maior controle sobre o estudo realizado, os resultados são mais confiáveis, pois existe uma minimização dos efeitos que os fatores externos exercem sobre o estudo. Por outro

¹ Marco estável, representado pelo conjunto de técnicas, métodos e processos bem estabelecidos, em um ambiente de desenvolvimento ou produção.

lado, é preciso um grande cuidado no planejamento e preparação do ambiente do experimento, para que os resultados obtidos sejam aplicáveis a um ambiente real. Normalmente, a realização de um experimento deve seguir um processo de cinco etapas: definição, planejamento, operação, análise e empacotamento. Essas etapas são descritas mais adiante (seção 2.7).

2.3.4. Comparação entre os Tipos de Estudos Experimentais

A tabela 2.1 (abaixo) procura comparar os três tipos de estudos experimentais, segundo alguns fatores. Entre outras coisas, pode-se observar que o custo do estudo aumenta no mesmo sentido do aumento do controle do experimentador sobre a investigação.

Tabela 2.1. A comparação de estratégias empíricas (WOHLIN *et al.*, 2000)

Fator	Pesquisa de Opinião (Survey)	Estudo de Caso	Experimento
O controle da execução	Nenhum	Nenhum	Tem
O controle da medição	Nenhum	Tem	Tem
O controle da investigação	Baixo	Médio	Alto
Facilidade de repetição	Alta	Baixa	Alta
Custo	Baixo	Médio	Alto

2.4. Elementos de um Estudo Experimental

O conhecimento de alguns conceitos é fundamental para condução de um estudo experimental. O texto a seguir descreve brevemente cinco deles: participantes, objetos, variáveis, tratamentos e hipóteses.

Os participantes ou sujeitos de um experimento são as pessoas envolvidas na sua execução, que produzem os resultados a serem analisados e interpretados.

Objeto é um artefato usado para verificar o relacionamento causa-efeito numa teoria (TRAVASSOS, GUVOV & AMARAL, 2002). Exemplo: o código-fonte de um programa, a ser inspecionado através de duas técnicas diferentes de leitura.

Na condução de um estudo experimental, o principal objetivo é estudar o resultado quando se aplicam diferentes valores para as variáveis de entrada. Segundo WOHLIN *et al.* (2000), existem dois tipos de variáveis em um experimento, as variáveis dependentes e as variáveis independentes.

As variáveis independentes são aquelas que são manipuladas e controladas, como, por exemplo, técnicas de construção de casos de uso, as experiências das pessoas, ou mesmo os ambientes onde o estudo está sendo executado (WOHLIN *et al.*, 2000). São também chamadas **variáveis de entrada** ou **fatores** (TRAVASSOS, GUROV & AMARAL, 2002).

Os tratamentos são os valores assumidos pelas variáveis independentes, ou seja, pelas variáveis de entrada do processo de experimentação. O tratamento é aplicado à combinação de objetos e participantes. Por exemplo, em um estudo visando comparar duas técnicas, a variável independente assume, como valores, cada uma das técnicas sob investigação.

As variáveis dependentes são os resultados do processo de experimentação, ou seja, representam os efeitos que uma ou mais variáveis independentes exercem nos objetos do experimento. São também denominadas de **variáveis de saída** ou **resultados** (TRAVASSOS, GUROV & AMARAL, 2002).

A figura 2.1 (a seguir) ilustra como variáveis dependentes e independentes estão fortemente ligadas:

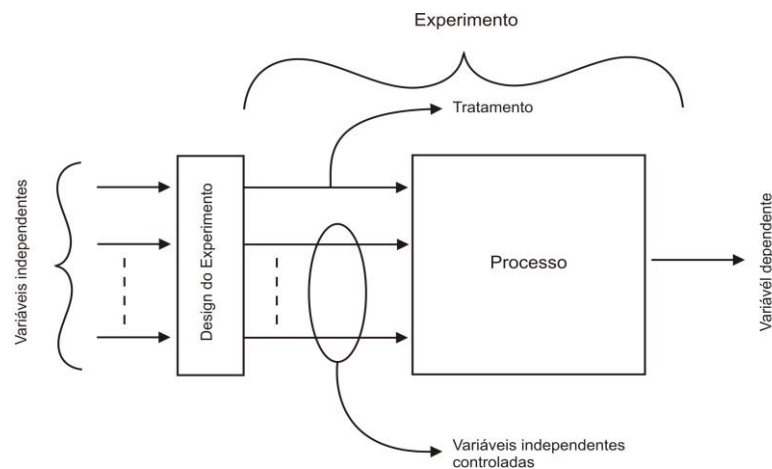


Figura 2.1. Elementos de um experimento (adaptado de WOHLIN *et al.*, 2000)

Hipóteses são suposições baseadas em alguma teoria, que um estudo experimental procura confirmar ou refutar. Se a hipótese puder ser confirmada a partir dos resultados do estudo, tem-se novas evidências da validade da teoria. Em outras palavras, o efeito causa-efeito esperado na teoria, deve ser concretizado no mundo real. O objetivo do mapeamento do mundo teórico para o mundo real é representado pela figura 2.2.

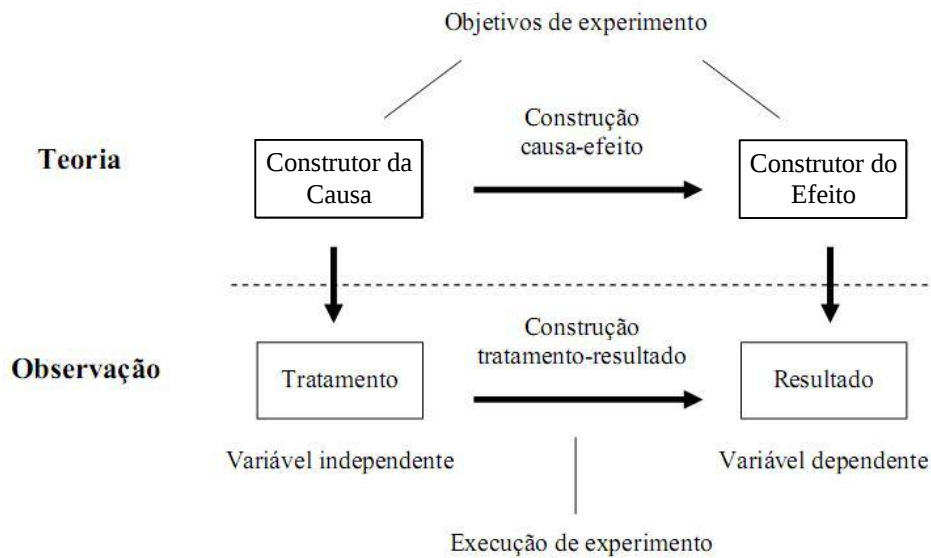


Figura 2.2. Princípios de um experimento (adaptado de WOHLIN *et al.*, 2002)

2.5. O Processo de Experimentação

Processos são importantes, pois eles promovem um padrão a ser adotado quando se deseja obter um produto final. Portanto, foram definidas atividades que devem ser seguidas para a condução de experimentos.

Segundo WOHLIN *et al.* (2000), o processo para realização de um experimento é dividido em cinco etapas, que são: definição, planejamento, execução, análise e empacotamento. A figura 2.3 ilustra esse processo.

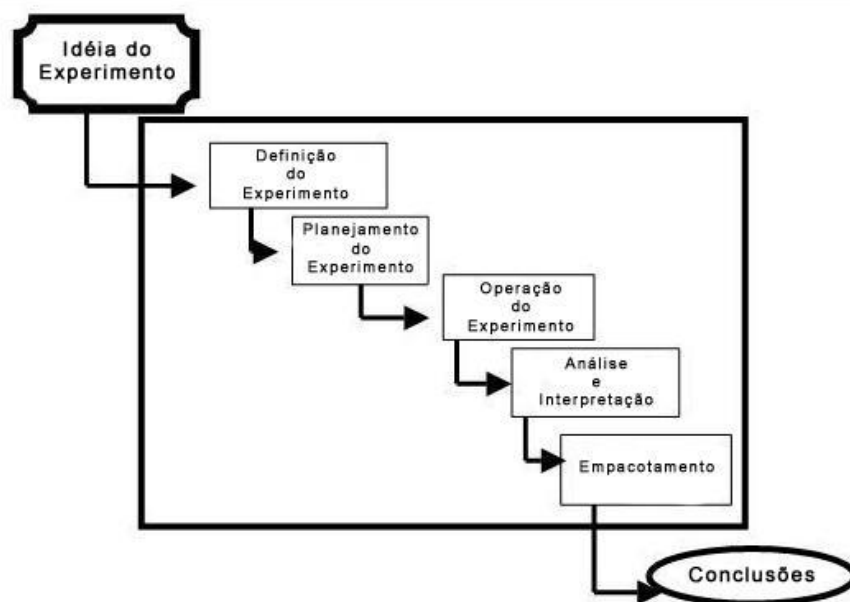


Figura 2.3. O processo experimental (WOHLIN *et al.*, 2000)

Nas subseções seguintes serão detalhadas as etapas deste processo de experimentação.

2.5.1. Definição

Nesta etapa, as hipóteses que serão testadas no experimento devem ser claramente estabelecidas. Elas não precisam ser formalizadas nesta etapa, mas devem ser bem elaboradas, para não ser necessário voltar novamente a esta etapa no futuro.

A proposta da fase de definição é definir os objetivos do experimento de acordo com uma estrutura padrão.

Segundo RODRIGUES (2006), o paradigma GQM (*Goal/Question/Metric*) pode contribuir para que o processo de medição esteja alinhado com os objetivos do estudo experimental. Neste sentido, as metas (*goals*) do experimento devem ser claramente definidas. Depois, para cada meta, são elaboradas questões que permitem verificar o alcance da meta. Por fim, para cada questão, são definidas uma ou mais métricas capazes de responder à questão de forma quantitativa (subjettiva ou objetivamente). A figura 2.4 representa a abordagem GQM. O experimento definido no capítulo 4 utilizou o paradigma GQM.

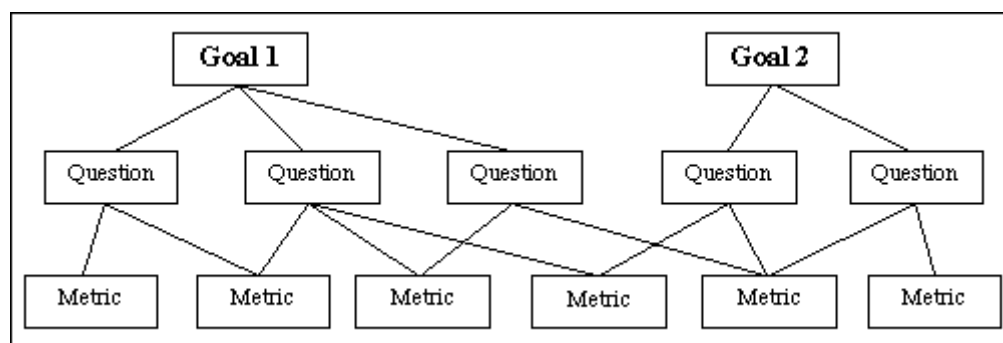


Figura 2.4. Estrutura hierárquica de GQM (WOHLIN *et al.*, 2000)

2.5.2. Planejamento

No planejamento, os fundamentos do estudo são estabelecidos. Esta etapa envolve especificar precisamente o contexto do estudo, as hipóteses, as variáveis a serem medidas, os participantes, o projeto (*design*) do estudo a ser executado, os instrumentos que serão utilizados ao longo da execução, e as ameaças à sua validade (WOHLIN *et al.*, 2000).

A seleção do contexto do experimento consiste em apresentar como o experimento será executado, identificando as seguintes dimensões (WOHLIN *et al.*, 2000):

- *On-line* vs. *off-line*: se o experimento será ou não executado em ambiente de trabalho;
- Estudante vs. profissional: indica o nível de experiência dos participantes;
- Problema real vs. inventado: define se o problema é real ou não;
- Geral vs. específico: indica a aplicabilidade dos resultados do experimento, ou seja, se é para todo domínio da Engenharia de Software ou não.

A formulação das hipóteses serve de base para a análise estatística. Para cada métrica envolvida no estudo, em geral se define duas hipóteses: a hipótese nula e hipótese alternativa. A hipótese nula é aquela que afirma a inexistência de relação causa-efeito entre as variáveis independentes e dependentes, no que diz respeito à métrica em questão, computada a partir dos resultados do experimento. A hipótese alternativa é inversa à hipótese nula. Se a hipótese nula puder ser rejeitada, então a hipótese alternativa poderá ser aceita ou não, a um *nível de significância*² previamente adotado. Os testes estatísticos verificam se é possível rejeitar a hipótese nula, de acordo com os dados coletados e suas propriedades estatísticas. Em geral, os testes estatísticos comparam as médias dos valores das métricas, entre grupos de participantes que aplicam tratamentos diferentes às variáveis independentes.

No planejamento do estudo se escolhe quais serão as variáveis dependentes e independentes (conforme definição apresentada na seção 2.5). Também é feita a seleção dos participantes, o que determina a capacidade de generalização dos resultados do estudo. Para que os resultados possam ser generalizados para a população-alvo do experimento, os participantes selecionados devem constituir uma amostra representativa dessa população.

Em geral, um experimento envolve uma série de testes, cada um correspondendo a um determinado conjunto de tratamentos, atribuído às variáveis independentes. O projeto do experimento descreve como os testes são organizados e executados (WOHLIN *et al.*, 2000). O projeto do experimento também define, em grande parte, que métodos estatísticos poderão ser aplicados para se tentar rejeitar a hipótese nula. O projeto de um experimento deve considerar três princípios básicos: *aleatoriedade*, *balanceamento* e *blocagem*.

² Probabilidade máxima admissível para o risco de se rejeitar (erroneamente) a hipótese nula, sendo ela verdadeira.

A alocação dos objetos e participantes em cada teste deve ser feita de forma aleatória, bem com a ordem de execução dos testes. A aleatoriedade também é a forma ideal de se selecionar os participantes do experimento. Se a alocação dos tratamentos for feita de tal forma que cada tratamento seja utilizado por um mesmo número de participantes, então se tem um projeto balanceado. A blocagem possibilita a eliminação dos efeitos de um fator (variável independente controlável – vide seção anterior), sobre os resultados do experimento. Isso é útil quando não se deseja levar em conta esse fator no experimento. Por exemplo, se a experiência dos participantes com certa técnica pode influenciar os resultados de um experimento, mas não se quer levá-la em consideração, pode-se dividir os participantes em dois blocos – o dos participantes com experiência e o dos participantes sem experiência. Com isso, os efeitos podem ser estudados apenas dentro de cada bloco, e não entre os blocos (WOHLIN *et al.*, 2000).

A análise estatística dos dados coletados fica facilitada quando se escolhe um dos projetos padronizados de experimento, pois para cada um desses projetos já estão determinados os métodos estatísticos adequados a essa análise. WOHLIN *et al.* (2000) apresenta quatro projetos padronizados: um fator com dois tratamentos; um fator com mais de dois tratamentos; dois fatores com dois tratamentos; e mais de dois fatores com dois tratamentos para cada fator.

Outra fase importante do planejamento é a descrição da instrumentação, ou seja, dos instrumentos ou artefatos a serem utilizados nos testes. Estes são, em geral, de três tipos: instrumentos de medição, objetos e diretrizes (que pode incluir material didático para treinamento dos participantes) (WOHLIN *et al.*, 2000). Exemplos de objetos utilizados em experimentos são documentos que contém a especificação ou o código-fonte de um sistema.

A principal questão relacionada aos resultados de um experimento é a sua validade. É muito importante avaliar os riscos à validade dos resultados antes mesmo de executar o experimento, de forma que providências possam ser tomadas para minimizar os riscos. COOK (1979) *apud* WOHLIN *et al.* (2000) apresenta quatro tipos de validades a experimentos, sendo elas: validade interna, validade de conclusão, validade de construção e validade externa.

A validade interna se preocupa com a relação de causa-efeito que deve existir entre os tratamentos aplicados e os resultados coletados. A análise desse tipo de validade envolve avaliar se algum fator desconhecido ou não controlado está determinando os resultados.

A validade de conclusão está diretamente ligada à possibilidade de se estabelecer uma relação estatística, a um dado nível de significância, entre cada tratamento e os dados (resultados) coletados com a sua aplicação.

A validade de construção considera o relacionamento entre teoria e observação. Supondo que existe de fato um relacionamento causal entre os tratamentos e respectivos resultados, é necessário garantir também: 1) que o tratamento reflete bem o construtor da causa; e 2) que o resultado reflete bem o construtor do efeito (veja figura 2.5 mais adiante).

A validade externa trata da generalização dos resultados. Considerando que existe de fato um relacionamento causal entre o construtor da causa, e o efeito, pode o resultado ser generalizado para fora do escopo do experimento? Entre as ameaças à validade externa estão: a escolha equivocada dos participantes (por exemplo, sujeitos não representativos da população-alvo), preparação inadequada do ambiente, e execução realizada segundo um cronograma que afeta os resultados (WOHLIN *et al.*, 2000).

A figura 2.5 mostra as dimensões da teoria e da observação no projeto de um experimento, e os pontos onde cada tipo de validade deve ser considerado.

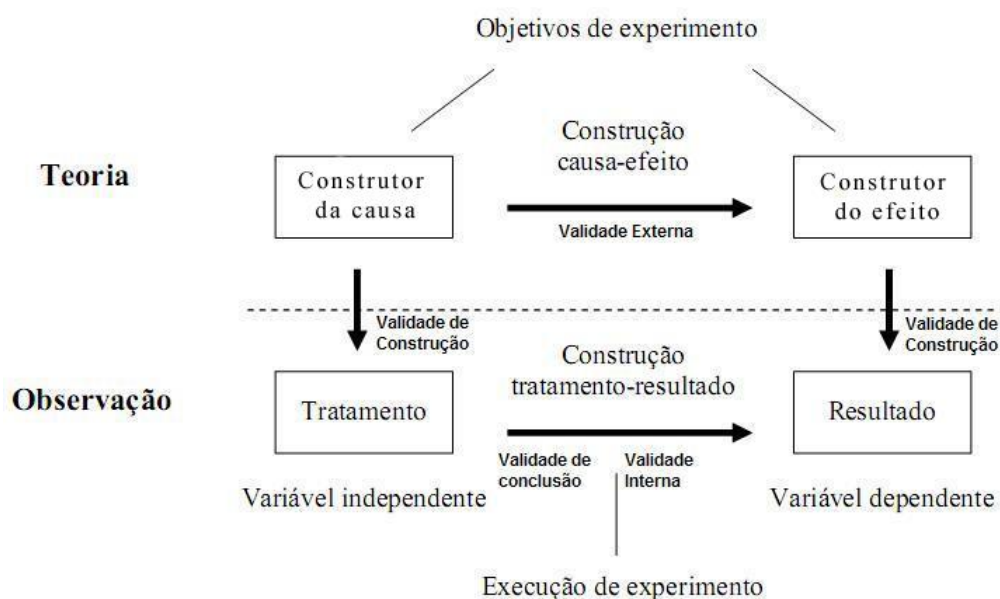


Figura 2.5. Princípios e tipos de validade de um experimento (adaptado de WOHLIN *et al.*, 2000)

É de fundamental importância a verificação do planejamento, observando cada detalhe e corrigindo logo possíveis erros, pois uma vez iniciada a execução, em geral não é possível voltar ao planejamento sem que o experimento seja cancelado e os participantes invalidados para qualquer outro estudo com os mesmos objetivos.

2.5.3. Operação

Após a definição e o planejamento do experimento, é necessário coletar os dados que deverão ser posteriormente analisados. Isso é feito na etapa de operação, onde cada teste, representado por um conjunto de tratamentos aplicados aos fatores, é executado pelos participantes.

Segundo WOHLIN *et al.* (2000), esta etapa pode ser dividida em três fases: preparação, execução e validação dos dados coletados. A fase de preparação envolve a motivação dos participantes e a elaboração dos materiais (instrumentação) mencionados na etapa de planejamento. Com isso, o experimento pode ser executado resultando na coleta dos dados observados. Por fim, os dados coletados devem ser validados, o que pode resultar na desconsideração de alguns deles, por erro de medição ou de interpretação. A validação de dados pode envolver os participantes para, por exemplo, verificar a interpretação dos mesmos, pelo experimentador.

2.5.4. Análise

Após a execução do experimento, o objetivo é chegar a conclusões através dos dados coletados. TRAVASSOS, GUROV & AMARAL (2002) afirmam que neste momento é necessário aplicar estatística descritiva, escolher o teste estatístico apropriado para o teste das hipóteses, explicar os resultados considerando as ameaças à sua validade, realizar uma análise de custo-benefício, e interpretar corretamente os resultados negativos.

A estatística descritiva serve para uma análise preliminar ao teste de hipóteses, permitindo uma primeira visão de como os dados estão distribuídos. Medidas de tendência central, de dispersão e de dependência, juntamente com a visualização em gráficos, auxiliam nessa análise e permitem a detecção de *outliers*³ (WOHLIN *et al.*, 2000). Tais *outliers* podem resultar, por exemplo, de erros de interpretação dos participantes.

O objetivo do teste das hipóteses é verificar se é possível, através de testes estatísticos, rejeitar a hipótese nula. Caso afirmativo, analisam-se as hipóteses alternativas com base nos resultados que o teste da hipótese nula produziu. Com isto é possível tirar conclusões estatisticamente significantes sobre os resultados.

No experimento apresentado no capítulo 4, o teste estatístico das hipóteses não foi realizado, por conta do escopo do trabalho e também devido ao baixo número de participantes no estudo.

³ Dados muito distantes da média.

Uma importante atividade de análise é a avaliação das ameaças à validade dos resultados do experimento. O Capítulo 4 inclui a análise desse tipo, para o experimento desenvolvido.

Por fim, deve ser apresentada uma conclusão geral sobre o experimento, para indicar, por exemplo, o custo-benefício do estudo, além de outros aspectos que o experimentador achar relevante.

2.5.5. Empacotamento

TRAVASSOS, GUROV & AMARAL (2002) relatam a importância de repetição para a validação de um experimento. Se os resultados obtidos em um experimento puderem ser reproduzidos por diferentes pesquisadores aplicando o mesmo projeto, então existirão fortes indícios de que esses resultados são efetivamente válidos e aplicáveis ao contexto do experimento. Para que um experimento possa ser repetido, é fundamental dispor de uma documentação cuidadosamente preparada e padronizada. À atividade de preparação desse documento, dá-se o nome de **empacotamento** do experimento.

WOHLIN *et al.* (2000) sugerem uma estrutura a ser seguida para o empacotamento do experimento, contendo uma introdução, um resumo dos problemas ocorridos, o planejamento do experimento, a descrição de como foi a execução do experimento, a análise dos dados, a interpretação dos resultados, uma conclusão final, além de um apêndice contendo todos os materiais utilizados no experimento. O empacotamento do experimento desenvolvido no presente trabalho foi feito seguindo essa estrutura.

Capítulo 3

Abordagens para a Modelagem com Casos de Uso:

Tradicional e Informacional

O objetivo da Engenharia de Software é propiciar um conhecimento amplo do processo de desenvolvimento de software. A especificação de requisitos de um software é uma etapa desse processo, e normalmente envolve a elaboração de mais de um tipo de modelo. Em geral, utiliza-se o modelo de casos de usos (JACOBSON *et al.*, 1992) (BITTNER & SPENCE, 2002) para descrição do comportamento externo do sistema, e o modelo de classes (de domínio) (BOOCH, RUMBAUGH & JACOBSON, 2000) para especificar a funcionalidade interna e informações de estado necessárias para uma adequada especificação do comportamento externo.

Neste capítulo são abordadas duas técnicas de casos de uso, para a modelagem de requisitos de sistemas de informação: a técnica tradicional (JACOBSON *et al.*, 1992), amplamente utilizada na prática, e a técnica informacional (FORTUNA & BORGES, 2005) (FORTUNA, WERNER & BORGES, 2007), que é uma especialização da primeira. Embora a técnica tradicional não esteja restrita a sistemas de informação, a técnica informacional está, motivo pelo qual, o contexto de investigação deste trabalho também está restrito a sistemas de informação.

Cada uma dessas técnicas traz consigo diretrizes para a obtenção de um diagrama de classes a partir dos casos de uso. O experimento objeto deste trabalho visa comparar as técnicas no que diz respeito a algumas propriedades dos modelos de classes resultantes.

3.1. Modelagem de Sistemas com Casos de Uso

A breve descrição aqui apresentada sobre a modelagem tradicional de requisitos com casos de uso (*Use Cases* - UC's) está baseada principalmente em (BITTNER & SPENCE, 2002).

Os casos de uso de um sistema descrevem a utilização que seus atores⁴ fazem dele. Todo UC deve permitir o alcance de algum objetivo de, pelo menos, um dos atores, e portanto, pode ser encontrado a partir de uma análise sobre o que os atores necessitam obter com a ajuda do sistema.

⁴ Entidade externa ao sistema (pessoa, dispositivo físico, outro sistema), que com ele interage.

Um UC deve descrever uma tarefa com algum valor identificável para o ator. A principal finalidade dessa diretriz é fazer com que o modelador se mantenha em um nível adequado de abstração ou de granularidade na elaboração do modelo de UC's.

Uma vez identificado um conjunto inicial de UC's correspondentes aos objetivos dos atores, muitos outros surgirão para apoiar os primeiros. A descoberta deles ocorre par a par com a descoberta de atores. A modelagem de um UC pode levar à identificação de novos atores, o que pode, por sua vez, levantar a necessidade de novos UC's.

Uma possível estrutura para a documentação de um UC envolveria as seguintes propriedades:

- Nome do UC: Cada UC deve ter um nome que indique o que o ator obtém ao interagir com ele, podendo ser necessário utilizar várias palavras para isso. O nome deve ser único;
- Descrição breve: Descrição breve do papel e do propósito do UC ;
- Fluxo de eventos (*outline*): Uma descrição textual do fluxo básico de eventos representando a interação normal (ou mais freqüente), entre o sistema e um ou mais atores.

Vários autores propõem descrições mais detalhadas, embora reconheçam que nem sempre isso é necessário. Por exemplo, as seguintes propriedades também poderiam fazer parte da descrição de um UC (BITTNER & SPENCE, 2002):

- Requisitos especiais: Uma descrição textual de todos os requisitos não incluídos nos fluxos de eventos do UC (tais como requisitos não funcionais), mas que precisam ser considerados durante o projeto e implementação do sistema;
- Pré-condições: Descrição textual que define restrições que o sistema deve satisfazer para que o caso de uso possa ser executado ;
- Pós-condições: Descrição textual que define o estado do sistema quando o caso de uso termina;
- Pontos de extensão: Uma lista de posições dentro do fluxo de eventos do UC, onde pode ser inserido comportamento adicional (por exemplo, fluxos alternativos – veja a seguir);
- Fluxos alternativos: seqüências de eventos que representam caminhos alternativos de interação, ativadas quando uma condição pré-determinada ocorre;

- Cenários: Instâncias específicas de um UC consistindo do fluxo básico⁵ mais zero ou mais fluxos alternativos.

No experimento descrito neste trabalho, utilizou-se a forma simplificada de descrição de UC's. Para exemplificar, são mostradas a seguir (figura 3.1) as descrições simplificadas de três UC's, que fazem parte da especificação de requisitos de um sistema de gerência de restaurante. A especificação completa do sistema foi utilizada como material de apoio no treinamento dos participantes do experimento descrito no capítulo 4, que utilizaram a técnica tradicional de modelagem de requisitos com UC's.

Ator: Caixa	UC1: Abrir pedido
1. O ator Caixa informa ao sistema sua intenção de registrar um novo pedido; 2. O Sistema solicita as informações sobre o pedido (número da mesa, pratos e bebidas pedidas com respectivas quantidades); 3. O Caixa fornece as informações sobre o pedido; 4. Caso o Caixa saiba, ou o garçom tenha indicado que se trata de um cliente habitual do restaurante, ele pode indicar isso ao Sistema, fornecendo a identificação do cliente perante o Sistema (essa identificação pode ser consultada pelo Caixa, a partir do nome do Cliente - UC 6); 5. O Sistema registra o pedido; 6. O caso de uso termina.	
Ator: Caixa	UC2: Emitir notinha de encerramento de refeição
1. O ator Caixa informa ao sistema a necessidade de emitir a notinha de encerramento de uma refeição; 2. O Sistema solicita o número da mesa onde a refeição está acontecendo; 3. O Caixa informa o número da mesa; 4. Caso o Caixa saiba, ou o garçom tenha informado que se trata de um cliente habitual, e essa informação ainda não tenha sido passada ao Sistema, o Caixa pode fornecer ao Sistema a identificação do cliente (essa identificação pode ser consultada pelo Caixa, a partir do nome do Cliente - UC6); 5. O Caixa solicita ao Sistema a emissão da notinha de encerramento de refeição; 6. O Sistema emite a notinha; 7. O caso de uso termina.	
Ator: Caixa	UC3: Registrar pagamento de conta(s)
1. O Caixa informa ao Sistema sua intenção de registrar o pagamento de conta(s) de refeição; 2. O Sistema permite que o Caixa informe o número da mesa, caso se trate de uma conta em aberto, ou o número da(s) notinha(s) de pendura, no caso de conta(s) pendurada(s), ou ambos; 3. O Sistema solicita o valor monetário entregue pelo Cliente para pagamento da(s) conta(s); 4. O Caixa informa o valor entregue pelo Cliente; 5. O Sistema calcula e apresenta o valor a ser devolvido ao Cliente como troco (zero,	

⁵ Sequência mais comum de eventos durante a interação do sistema os atores.

- se não for devido troco);
6. O Caixa solicita ao Sistema o registro do pagamento;
 7. O Sistema registra o pagamento da(s) conta(s);
 8. O caso de uso termina.

Figura 3.1 - Exemplos de UC's de um sistema de gerência de restaurante

Muitos autores (entre eles BITTNER & SPENCE (2002)) sugerem a inclusão de um *glossário de termos* no final da especificação de UC's, para a definição precisa de termos comuns no domínio do sistema, evitando assim, desde o início, a ambigüidade na utilização dos mesmos. Embora os termos pudessem ser definidos dentro dos UC's, a utilização de um glossário separado traz algumas vantagens, entre elas, a possibilidade de defini-los em um único lugar. A tabela 3.1 ilustra uma parte do glossário de termos que acompanha a descrição dos UC's do sistema de gerência de restaurante:

Tabela 3.1. Glossário de termos do sistema de gerência de restaurante (parcial)

Termo	Descrição	Informação Adicional
Conta em aberto	Conta correspondente a uma refeição que ainda não foi paga ou pendurada.	Normalmente, uma conta em aberto corresponde a uma refeição que está ocorrendo e será paga assim que terminar.
Conta pendurada	Conta correspondente a uma refeição tomada por um cliente habitual, cujo pagamento foi postergado, a pedido dele.	Normalmente, o pagamento de contas penduradas ocorre no final de cada mês, por iniciativa do cliente.
Item de consumo	Prato ou bebida incluídos no cardápio do restaurante.	O gerente mantém, via sistema, um cadastro dos itens de consumo que estão sendo servidos pelo restaurante.
Notinha de encerramento (de refeição)	Ticket que descreve, principalmente, a refeição (pratos e bebidas) tomada, bem como os valores individuais e o valor total para pagamento da refeição.	No caso de cliente habitual, pode apresentar os dados do cliente (número identificador e nome, por exemplo).
Notinha de pendura	Notinha de encerramento, com propósito específico de documentar uma pendura de conta, contendo informações sobre o cliente habitual responsável pela refeição.	Deve ser assinada pelo cliente para atestar a refeição tomada e a pendura da conta.

Normalmente, a descrição de UC's de um sistema é apontada como uma fonte de informações úteis à elaboração de um diagrama de classes (de domínio) para o sistema (JACOBSON *et al*, 1992). Vários autores propõem algumas diretrizes para guiar o

aproveitamento dessas descrições neste sentido. Por exemplo, COAD & YOURDON (1990), WIRFS-BROCK, WILKERSON & WIENER (1990) e ABBOTT (1983, *apud* LARMAN, 2004) sugerem a identificação de classes gramaticais no texto dos UC's, para guiar a descoberta dos diversos elementos⁶ do diagrama de classes. Por exemplo, substantivos e verbos (de ação), que porventura ocorram nas descrições dos UC's, podem ser considerados candidatos a classes (ou atributos) e operações, respectivamente, no modelo de classes. No entanto, a decisão final sobre a inclusão ou não deles no modelo de classes, fica por conta do modelador.

O experimento descrito neste trabalho envolveu a elaboração do modelo de classes a partir das descrições de UC's de um sistema. Os participantes que aplicaram a técnica tradicional de UC's foram orientados a fazer a identificação de substantivos e verbos no texto dos UC's, como um expediente capaz de auxiliar a elaboração do modelo de classes do sistema.

3.2. Modelagem Informacional de Requisitos

Para obter uma maior consistência entre modelos de casos de uso e modelos de classes de domínio, FORTUNA & BORGES (2005) introduziram um tipo de modelagem denominado Modelagem Informacional de Requisitos (MIR), que se apóia em uma especialização dos UC's tradicionais. Graças à essa especialização, é possível determinar uma série de regras mais ou menos automatizáveis para a derivação de um modelo de classes a partir do modelo especializado de UC's (FORTUNA, WERNER & BORGES, 2007).

O restante dessa seção está baseada nas referências indicadas no parágrafo anterior, e visa apresentar e exemplificar a MIR e suas regras de derivação do diagrama de classes a partir da descrição dos UC's do sistema.

3.2.1. Modelo Integrado de Requisitos

Segundo FORTUNA, WERNER & BORGES (2007), a Modelagem Informacional de Requisitos é uma especialização de casos de uso, com as seguintes restrições: 1) a fixação de um nível especial de abstração para criação dos casos de uso; e 2) um maior detalhamento e rigor formal na especificação das informações trocadas entre o sistema e seus atores.

⁶ Classes, atributos, associações e operações, principalmente.

O nível informacional de objetivos (NIO) é o nível de abstração para objetivos dos atores, induzido pelos eventos autônomos gerados por eles. *Evento autônomo* é uma intervenção no sistema, de iniciativa de um ator, que dispara um processo do sistema, cuja execução leva ao alcance de um objetivo do ator e deixa o sistema em um estado estável. Por estado estável entende-se um estado em que fica descartada qualquer necessidade de retorno (*rollback*) a um estado anterior. Na MIR, os UC's correspondem, de forma biunívoca, aos objetivos do NIO; ou seja, são exatamente aqueles ativados pelos eventos autônomos.

As informações que fluem entre o sistema e seu ambiente (atores) durante a execução de um UC constituem a **interface informacional (II)** do UC. A especificação da II de um UC tem duas partes: a) especificação de fluxos, e b) dicionário de itens elementares, ilustradas na figura 3.2 e na tabela 3.3, respectivamente, para o mesmo sistema de gerência de restaurante considerado na seção 3.1.

O nível de detalhamento e o rigor formal preconizados pela MIR para a especificação da II dos UC's, ficam garantidos pela linguagem semiformal utilizada nessa especificação. Essa linguagem utiliza vários símbolos especiais (tabela 3.2).

Tabela 3.2. Símbolos utilizados na linguagem de especificação da II dos UC's

Símbolo	Significado
=	É composto de
+	e
	ou
[...]	Itens condicionais (podem não ser pertinentes, dependendo do contexto).
$_n\{ \dots \}_m$	Itens que se repetem, no mínimo n e no máximo m vezes.
(...)	Delimitadores

Ator: Caixa	UC1: Abrir pedido
1. O ator Caixa informa ao sistema sua intenção de registrar um novo pedido; 2. O Sistema solicita as informações sobre o pedido (identificador da mesa, e dos pratos e bebidas pedidas com respectivas quantidades); 3. O Caixa fornece as informações sobre o pedido; $\rightarrow \text{pedido} = \text{dt_pedido} + \text{id_mesa} + \text{itens_ped}$ $\text{itens_ped} = \{ \text{id_item} + \text{quant_item} \}$ 4. O Sistema registra o pedido; $\leftarrow \text{id_pedido}$ 5. O caso de uso termina.	
Ator: Caixa	UC2: Emitir notinha de encerramento de refeição
1. O ator Caixa informa ao sistema a necessidade de emitir a notinha de encerramento de uma refeição; 2. O Sistema solicita o identificador do pedido correspondente à refeição que se está encerrando; 3. O Caixa informa o identificador do pedido; 4. Caso o Caixa saiba, ou o garçom tenha informado que se trata de um cliente habitual, o Caixa pode fornecer ao Sistema a identificação do cliente (essa identificação pode ser consultada pelo Caixa, a partir do nome do Cliente - UC7); $\rightarrow \text{solicitacao_nota} = \text{id_pedido} + [\text{id_cliente}]$ 5. O Caixa solicita ao Sistema a emissão da notinha de encerramento de refeição; 6. O Sistema emite a notinha; $\leftarrow \text{nota} = \text{id_pedido} + \text{nr_mesa} + \text{dt_pedido} + \text{itens_nota} + \text{vl_nota} + [\text{nome_cliente} + \text{tel_cliente}]$ $\text{itens_nota} = \{ \text{id_item} + \text{cat_item} + \text{nome_item} + \text{pç_unit} + \text{quant_item} + \text{vl_item} \}$ 7. O caso de uso termina.	
Ator: Caixa	UC3: Registrar pagamento de conta
1. O Caixa informa ao Sistema sua intenção de registrar o pagamento de conta de refeição; 2. O Sistema permite que o Caixa informe o identificador do pedido (em aberto ou pendurado) a ser pago;3. O Sistema solicita o valor monetário entregue pelo Cliente para pagamento da conta; 4. O Caixa informa o valor entregue pelo Cliente; $\rightarrow \text{pagamento} = \text{id_pedido} + \text{vl_entregue} + \text{dt_pagto}$ 5. O Sistema calcula e apresenta o valor a ser devolvido ao Cliente como troco (zero, se não for devido troco); $\leftarrow \text{troco}$ 6. O Caixa solicita ao Sistema o registro do pagamento; 7. O Sistema registra o pagamento da conta; 8. O caso de uso termina.	

Figura 3.2. Especificação da II de três UC's do sistema de gerência de restaurante

Na figura 3.2, os sinais $\rightarrow$ e $\leftarrow$ indicam, respectivamente, *fluxo de entrada* e *fluxo de saída* de informações, do ponto de vista do sistema. Uma informação ou item em um

fluxo pode ser *elementar* (indivisível) ou composto (agregado de outros itens elementares ou compostos). Os itens compostos também são chamados de *pacotes*, e indicados pelo símbolo . *Item calculado* é todo item presente em um fluxo de saída (*item de saída*), cujo valor resulta de um cálculo efetuado pelo sistema. Um tipo especial de item calculado são os itens *identificadores*. Eles se destacam pelo nome iniciado por *id_* (por exemplo, *id_cliente*), e representam abstrações do domínio da aplicação incluídas na especificação de requisitos. Essa linguagem semiformal possibilita um maior grau de automatização na obtenção do diagrama de classes, através das regras discutidas na próxima seção.

Tabela 3.3. Dicionário de itens elementares (parcial)

Ator: Cliente		UC 3: Emitir notinha de encerramento de refeição	
Nome	Descrição	Tipo	Domínio
id_pedido	Identificador do pedido da nota a emitir.	Nr. Natural	
quant_item	Quantidade consumida de um item.	Nr. Natural	
vl_item	Valor do consumo de um item. Preço unitário × quantidade consumida do item.	Moeda	
cat_item	Categoria do item de consumo	Texto	{prato, bebida}

3.2.2. Derivação do Modelo de Classes do Domínio

A Modelagem Informacional de Requisitos fornece 14 regras para a derivação de um diagrama de classes a partir do modelo informacional de UC's. Todos os principais elementos de um diagrama de classes (classes, associações, atributos e operações) têm sua obtenção apoiada pela aplicação dessas regras.

A presente seção não pretende descrever cada uma das 14 regras, mas apenas ilustrar algumas delas. O conjunto completo de regras, bem como outras heurísticas a elas associadas, podem ser encontradas em (FORTUNA, WERNER & BORGES, 2007). A seguir são descritas 3 regras e sua aplicação ao sistema de gerência de restaurante. Para isso, a figura 3.3 apresenta a composição de todos os fluxos (de entrada e de saída) que constituem a interface informacional dos UC's desse sistema.

ATOR: Cliente	UC 1: Abrir pedido
→ pedido = dt_pedido + id_mesa + itens_ped 📄 itens_ped = ₁ {id_item + quant_item} ← id_pedido	
ATOR: Cliente	UC 2: Emitir notinha de encerramento de refeição
→ solicitacao_nota = id_pedido + [id_cliente] ← nota = id_pedido + nr_mesa + dt_pedido + itens_nota + vl_nota + [nome_cliente + tel_cliente] 📄 itens_nota = ₁ {id_item + cat_item + nome_item + pç_unit + quant_item + vl_item}	
ATOR: Cliente	UC 3: Pagar a nota
→ pagamento = id_pedido + vl_entregue + dt_pagto ← troco	
ATOR: Cliente	UC 4: Cancelar pedido
→ cancela_ped = id_pedido	
ATOR: Cliente	UC 5: Pendurar a nota
→ pendura = id_pedido + id_cliente	
ATOR: Gerente	UC 6: Cadastrar cliente habitual
→ cliente = nome_cliente + tel_cliente ← id_cliente	
ATOR: Gerente	UC 7: Atualizar o cardápio
→ item_consumo = nome_item + pc_unit + cat_item + descr_item ← id_item	
ATOR: Gerente	UC 8: Solicitar consumo diário
→ solic_consumo = dt_emissao + dt_consumo ← consumo_dia = dt_emissao + dt_consumo + consumo_itens 📄 consumo_itens = ₀ {cat_item + {id_item + nome_item + quant_item} } ₂	
ATOR: Gerente	UC 9: Solicitar receita
→ solic_receita = dt_emissao + periodo_apur 📄 periodo_apur = dt_inicio + dt_fim ← receita = dt_emissao + periodo_apur + vl_consumo + receita_realiz + receita_pend + receita_txServ + receita_total	
ATOR: Gerente	UC 10: Cadastrar mesa
→ mesa = nr_mesa ← id_mesa	
ATOR: Gerente	UC 11: Solicitar relação de notas penduradas
→ solic_penduras = dt_emissao + periodo_apur 📄 periodo_apur = dt_inicio + dt_fim ← penduras = dt_emissao + periodo_apur + {id_cliente + nome_cliente + tel_cliente + notas_pend + vl_pendCli} + vl_pend 📄 notas_pend = ₁ {id_pedido + dt_pedido + nr_mesa + itens_nota + vl_nota} 📄 itens_nota = ₁ {id_item + cat_item + nome_item + pç_item + quant_item + vl_item}	

Figura 3.3. Composição da Interface Informacional do sistema *Restaurante*

(FORTUNA, WERNER & BORGES, 2007)

Regra 1. Cada *identificador gerado* dá origem a uma classe de objetos.

Um identificador é *gerado* em um UC quando o seu valor é estabelecido durante a execução do UC. Por exemplo, *id_cliente* é um identificador gerado no UC 6 (figura 3.3) para servir de referência a um objeto (cliente) criado durante o processamento do UC.

Aplicando esta regra no sistema Restaurante (Figura 3.3), os identificadores gerados são: *id_pedido* (UC 1), *id_cliente* (UC 6), *id_item* (UC 7) e *id_mesa* (UC 10), desta maneira gerando as classes “*Pedido*”, “*Cliente*”, “*Item*” e “*Mesa*”, respectivamente.

As Regras 2a e 2b ajudam a determinar as associações entre classes, com base principalmente nos identificadores gerados nos UC’s.

Regra 2a. Todo par $\langle id_1, id_2 \rangle$, que possa ser formado com id_1 sendo um identificador gerado presente no fluxo de saída do UC, e id_2 um identificador presente no fluxo de entrada, determina uma associação entre as classes identificadas por id_1 e id_2 . No caso de mais de um identificador gerado, as associações correspondentes a pares que compartilham o mesmo id_2 tendem a ser redundantes.

No sistema Restaurante (figura 3.3): o UC 1 permite formar os pares de identificadores $\langle id_pedido, id_mesa \rangle$ e $\langle id_pedido, id_item \rangle$. Isso determina as associações *Pedido-Mesa* e *Pedido-Item*, respectivamente.

Regra 2b. Quando os identificadores ocorrerem apenas em um dos fluxos do UC (considerando, para o fluxo de saída, apenas os identificadores gerados), todo par de identificadores determina uma associação entre as classes por eles identificadas. Em fluxos com mais de dois identificadores, algumas dessas associações tendem a ser redundantes.

No sistema Restaurante: o par de identificadores $\langle id_pedido, id_cliente \rangle$, presente nos UC’s 2 (Emitir notinha de encerramento de refeição) e 5 (Pendurar a nota), determina a associação *Pedido-Cliente*.

O resultado da aplicação das 3 regras anteriores está ilustrado na figura 3.4 a seguir:

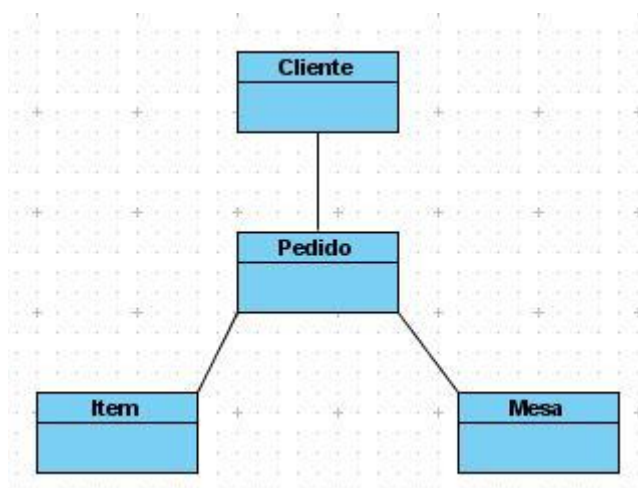


Figura 3.4. Diagrama de classes parcial, do sistema Restaurante (regras 1, 2a e 2b)

A figura 3.5 mostra o diagrama final, após a aplicação das 14 regras da MIR:

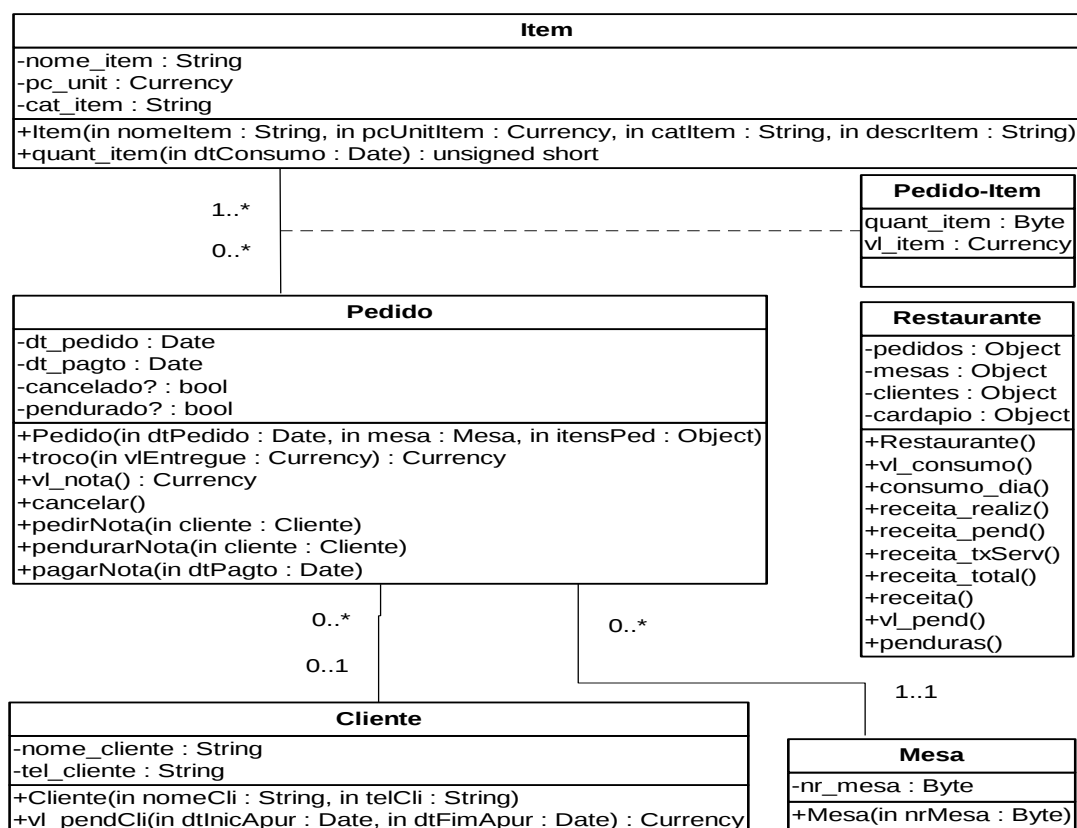


Figura 3.5. Diagrama de classes do sistema Restaurante (FORTUNA, WERNER & BORGES, 2007)

Capítulo 4

O Experimento

Este capítulo descreve um estudo experimental para comparação de duas técnicas de construção de modelos de casos de uso e de classes de domínio, que já foram caracterizadas no capítulo 3. O experimento foi conduzido através das etapas definidas em WOHLIN *et al.* (2000) e descritas no capítulo 2. Estas etapas são: definição, planejamento, execução, análise, e empacotamento do experimento. O documento resultante do empacotamento do experimento, que agrega todas as informações produzidas ao longo de todas as etapas do experimento, não é parte integrante desta monografia devido ao seu volume e por não ser essencial para a descrição do experimento. Entretanto, esse documento pode ser obtido por solicitação ao autor desta monografia.

4.1. Definição do Experimento

Seguindo a abordagem GQM (WOHLIN *et al.*, 2000), já descrita no capítulo 2, a definição do experimento foi feita em três níveis: o nível conceitual, no qual são apresentados os objetivos do estudo, o nível operacional das questões a serem respondidas para o alcance dos objetivos, e o nível quantitativo composto pelas métricas associadas às questões. O texto a seguir detalha a definição do experimento, segundo essa abordagem.

4.1.1. Objetivo do Estudo Experimental

O objetivo deste estudo experimental é comparar as duas técnicas – a técnica tradicional de Casos de Uso (JACOBSON *et al.*, 1992), (BITTNER & SPENCER, 2002) e técnica de Modelagem Informacional de Requisitos (MIR) (FORTUNA & BORGES, 2005) (FORTUNA, WERNER & BORGES, 2007) – descritas no capítulo 3, no que diz respeito aos modelos de classes (de domínio) obtidos com cada uma dessas técnicas, durante a modelagem de requisitos de sistema de informação. A restrição a sistemas de informação decorre do fato da MIR focar especificamente esse tipo de sistema.

As propriedades que serão avaliadas neste experimento são a *granularidade* e a *uniformidade* dos modelos de classes produzidos pelas técnicas consideradas.

A granularidade de um modelo de classes é medida pelo número de classes que o modelo possui, ou seja, quanto maior o número de classes, maior a granularidade do modelo.

Já uniformidade entre modelos de classes expressa o grau de semelhança entre eles. Enquanto a granularidade considera as classes presentes em cada modelo, a uniformidade considera as abstrações existentes nos modelos. Em geral, cada classe representa uma abstração; entretanto, pode ocorrer⁷ que uma mesma abstração, representada por uma única classe em um modelo, corresponda a mais de uma classe em outro modelo. É o que acontece, por exemplo, quando uma abstração representada por uma única classe em um modelo, sofre especializações em outro modelo.

Para uma avaliação abrangente da uniformidade entre modelos, foram considerados dois tipos de uniformidade:

- *Uniformidade de abstrações*: baseada na comparação das abstrações de domínio, existentes em cada modelo, a nível puramente semântico; e
- *Uniformidade representacional*: baseada na comparação de atributos, associações e operações utilizadas para representar cada abstração presente nos modelos.

Seguindo a sugestão de WOHLIN *et al.* (200), o objetivo do estudo encontra-se descrito abaixo, através do esquema GQM:

Analisar os modelos de classes de domínio obtidos com as duas técnicas de modelagem de requisitos consideradas (MIR e UC's),

Com o propósito de comparar a performance dessas técnicas,

Com respeito a granularidade dos modelos, e uniformidade entre os modelos produzidos por cada uma dessas técnicas,

Do ponto de vista de um engenheiro de requisitos,

No contexto da análise e desenvolvimento de sistemas de informação, por profissionais graduados em Ciência da Computação pela Universidade Federal de Juiz de Fora.

4.1.2. Questões e Métricas

Abaixo serão abordadas as questões e respectivas métricas utilizadas na comparação entre os modelos.

Q1: Como será medida a granularidade de um modelo de classes?

Métrica: Número de classes que participam do modelo.

Q2: Como será medida a uniformidade de abstrações entre dois modelos?

⁷ Como aliás ocorreu em dois modelos produzidos no experimento.

Métrica: Número de abstrações coincidentes, dividido pelo número de abstrações únicas, entre os dois modelos.

Essa métrica pode ser expressa pela fórmula:

$$\text{Unif-}A_{i,j} = \frac{A'_{i,j}}{(A_i + A_j) - (A'_{i,j})}$$

Onde, para $i \neq j$:

- A_i é a cardinalidade do conjunto das abstrações do Modelo i ;
- A_j é a cardinalidade do conjunto das abstrações do Modelo j ; e
- $A'_{i,j}$ é a cardinalidade do conjunto das abstrações comuns a ambos os modelos i e j .

Para esclarecer a métrica acima, considere o diagrama da figura 4.1. Nela, a uniformidade de abstrações entre os modelos está representada pela região de intercessão. Quanto maior essa região, maior é a uniformidade de abstrações dos modelos.

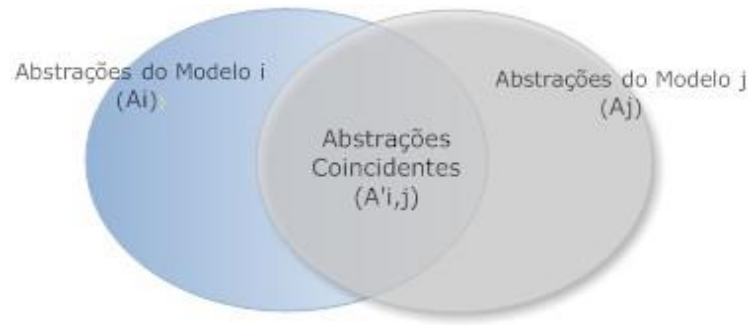


Figura 4.1 – Ilustração dos elementos da métrica de uniformidade de abstrações, entre dois modelos de classes

Q3: Como será medida a uniformidade de atributos entre dois modelos de classes?

Métrica: Número de atributos coincidentes, dividido pelo total de atributos únicos, considerando todas as classes que representam abstrações comuns aos dois modelos. Essa métrica pode ser expressa pela fórmula:

$$\text{Unif-}At_{i,j} = \frac{At'_{i,j}}{(At_{i,j} + At_{j,i}) - (At'_{i,j})}$$

Onde, para $i \neq j$:

- $At_{i,j}$ é a cardinalidade do conjunto de atributos do Modelo i , presentes nas classes que representam abstrações comuns a ambos os modelos i e j ;
- $At_{j,i}$ é a cardinalidade do conjunto de atributos do modelo j , presentes nas classes que representam abstrações comuns a ambos os modelos i e j ; e

- $At'_{i,j}$ é a cardinalidade do conjunto de atributos coincidentes, em todas as classes que representam abstrações comuns a ambos os modelos i e j .

Para esclarecer a métrica acima, considere o diagrama da figura 4.2. Nela, a uniformidade de atributos entre os modelos está representada pela região de intercessão.

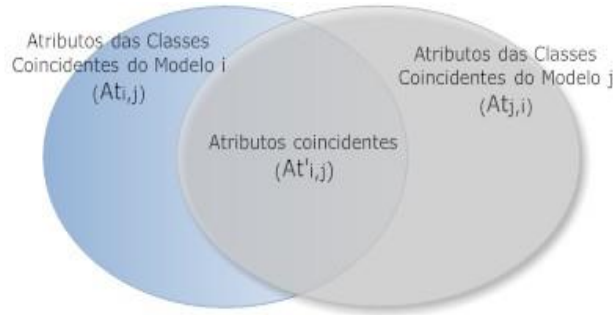


Figura 4.2 - Ilustração dos elementos da métrica de uniformidade de atributos, entre dois modelos de classes

Q4: Como será medida a uniformidade de operações entre dois modelos de classes?

Métrica: Número de operações coincidentes, dividido pelo total de operações únicas, considerando todas as classes que representam abstrações comuns aos dois modelos. Essa métrica pode ser expressa pela fórmula:

$$\text{Unif-O}_{i,j} = \frac{O'_{i,j}}{(O_{i,j} + O_{j,i}) - (O'_{i,j})}$$

Onde, para $i \neq j$:

- $O_{i,j}$ é a cardinalidade do conjunto de operações do Modelo i , presentes nas classes que representam abstrações comuns a ambos os modelos i e j ;
- $O_{j,i}$ é a cardinalidade do conjunto de operações do modelo j , presentes nas classes que representam abstrações comuns a ambos os modelos i e j ; e
- $O'_{i,j}$ é a cardinalidade do conjunto de operações coincidentes, em todas as classes que representam abstrações comuns a ambos os modelos i e j .

A ilustração dessa métrica é análoga à da métrica anterior (figura 4.2), e por isso será omitida.

Q5: Como será medida a uniformidade de associações entre dois modelos de classes?

Métrica: Número de associações coincidentes, dividido pelo total de associações únicas, considerando todas as classes que representam abstrações comuns aos dois modelos.

Essa métrica pode ser expressa pela fórmula:

$$\text{Unif-L}_{i,j} = \frac{L'_{i,j}}{(L_{i,j} + L_{j,i}) - (L'_{i,j})}$$

Onde, para $i \neq j$:

- $L_{i,j}$ é a cardinalidade do conjunto de associações do Modelo i , entre as classes que representam abstrações comuns a ambos os modelos i e j ;
- $L_{j,i}$ é a cardinalidade do conjunto de associações do modelo j , entre as classes que representam abstrações comuns a ambos os modelos i e j ; e
- $L'_{i,j}$ é a cardinalidade do conjunto de associações coincidentes, entre todas as abstrações comuns a ambos os modelos i e j .

A ilustração dessa métrica é análoga à da métrica de uniformidade de atributos (figura 4.2), e por isso será omitida.

Q6: Como será medida a uniformidade representacional entre dois modelos de classes?

Métrica: Média aritmética das uniformidades de atributos, operações e associações, entre os dois modelos.

Essa métrica pode ser expressa pela fórmula:

$$\text{Unif-R}_{i,j} = \frac{\text{Unif-At}_{i,j} + \text{Unif-O}_{i,j} + \text{Unif-L}_{i,j}}{3}$$

Onde:

- $\text{Unif-At}_{i,j}$: Uniformidade de atributos entre os modelos i e j ; e
- $\text{Unif-O}_{i,j}$: Uniformidade de operações entre os modelos i e j ; e
- $\text{Unif-L}_{i,j}$: Uniformidade de associações entre os modelos i e j .

4.2. Planejamento do Experimento

Seguindo as recomendações de WOHLIN *et al.* (2000) (descritas em parte no capítulo 2), o planejamento do experimento-alvo deste trabalho procurou cobrir os seguintes tópicos: seleção do contexto, seleção dos participantes, seleção das variáveis, formulação de

hipóteses, projeto do experimento, descrição da instrumentação, e avaliação das ameaças à validade do experimento. As subseções seguintes descrevem o planejamento do experimento, em cada um desses tópicos.

4.2.1. Seleção do Contexto

Conforme visto no capítulo 2, o contexto de um experimento pode ser caracterizado de acordo com quatro dimensões (WOHLIN *et al.*, 2000):

- Off-line vs. on-line;
- Estudantes vs. profissionais;
- Problema-exemplo vs. problema real; e
- Específico vs. geral.

O experimento em questão pode se caracterizado como **off-line**, pois não foi executado dentro de uma rotina real de trabalho, mas sim como um projeto especialmente criado para ele. Utilizou **profissionais** como participantes, embora alguns deles com pequena experiência de trabalho em modelagem de requisito⁸. Os participantes elaboraram a especificação de requisitos para um **sistema real**, a ser desenvolvido em atendimento às necessidades da empresa onde trabalha o autor desta monografia. Em vista dessa combinação de características, o presente experimento se situa em algum ponto entre o **específico e o geral**.

4.2.2. Seleção dos Participantes

Esta etapa do estudo de caso é uma das mais importantes, pois a generalização dos resultados depende fortemente de como é feita a seleção dos participantes.

Conforme visto na introdução deste capítulo, a população-alvo do experimento é constituída pelos profissionais graduados em Ciência da Computação pela Universidade Federal de Juiz de Fora (UFJF). A amostra da população utilizada no presente experimento foi selecionada por conveniência, sendo formada por seis pessoas conhecidas do autor deste trabalho, com disponibilidade para participar do estudo. Portanto, não foi utilizado o princípio (ideal) de aleatoriedade na seleção da amostra populacional⁹.

⁸ Em parte compensada por uma maior maturidade acadêmica, pois são alunos do mestrado em Modelagem Computacional da Universidade Federal de Juiz de Fora (UFJF).

⁹ Como aliás é o normal nos experimentos em Engenharia de Software, devido à dificuldade de se ter acesso a todos os indivíduos que constituem as populações de interesse nessa área.

Os seis participantes foram divididos em dois grupos com três participantes cada um. Cada grupo utilizou apenas uma das técnicas avaliadas neste estudo. Os participantes que formaram o grupo A, que utilizou a MIR, foram designados pelos números 1, 2 e 3. Já os participantes do grupo B, que utilizou a técnica tradicional, foram designados pelos números 4, 5, e 6. O único participante com experiência prévia relevante na MIR (P1 - vide tabela 4.2) foi designado para o grupo A. Outro participante (P2), também foi designado para o grupo A por já ter tido oportunidade de conhecer (ler) um modelo informacional. Para que cada grupo tivesse o mesmo número de participantes (três), no início do experimento foi solicitado que alguém mais se prontificasse a participar do grupo A (participante P3). Os demais, constituíram o grupo B (participantes P4, P5 e P6).

4.2.3. Seleção das Variáveis

Como explicado no capítulo 2, um experimento define variáveis independentes e variáveis dependentes. No caso deste experimento, a única variável independente (de entrada) é a técnica para a elaboração dos casos de uso (UC's) e obtenção do diagrama de classes do sistema. Essa variável pode assumir dois valores: a técnica tradicional e a técnica da MIR, ambas descritas no capítulo 3.

Para cada valor da variável independente, ou seja, a técnica empregada, as seguintes variáveis dependentes (de saída) contabilizam elementos dos modelos de classes produzidos com a técnica:

- C_i = Número de classes existentes no modelo produzido pelo participante i ;
- A_i = Número de abstrações existentes no modelo produzido pelo participante i ;
- $A'_{i,j}$ = Número de abstrações comuns aos modelos dos participantes i e j ;
- $At_{i,j}$ = Número de atributos no modelo do participante i , presentes nas classes que representam abstrações comuns com o modelo do participante j ;
- $At'_{i,j}$ = Número de atributos coincidentes, presentes nas classes que representam abstrações comuns aos modelos dos participantes i e j ;
- $O_{i,j}$ = Número de operações no modelo do participante i , presentes nas classes que representam abstrações comuns com o modelo do participante j ;
- $O'_{i,j}$ = Número de operações coincidentes, presentes nas classes que representam abstrações comuns aos modelos dos participantes i e j ;

- $L_{i,j}$ = Número de associações no modelo do participante i , entre classes que representam abstrações comuns com o modelo do participante j ;
- $L'_{i,j}$ = Número de associações coincidentes entre as classes que representam abstrações comuns aos modelos dos participantes i e j .

4.2.4. Formulação das Hipóteses

As seis métricas definidas (seção 4.1) levaram à formulação dos seis pares de hipóteses correspondentes, apresentados a seguir – cada par composto de uma hipótese nula e uma hipótese alternativa.

Embora o presente trabalho não aplique métodos estatísticos para a análise dos dados coletados, procurou-se utilizar na formulação das hipóteses o mesmo formalismo normalmente requerido para a aplicação de tais métodos. Com isso, caso no futuro se pretenda fazer o teste estatístico das hipóteses, a formulação adequada das mesmas já estará pronta.

4.2.4.1. Hipóteses para Granularidade

Hipótese Nula (H_0 -G): Não há diferença na granularidade dos modelos produzidos com a técnica de Modelagem Informacional de Requisitos (MIR), em relação àqueles produzidos com a técnica tradicional de casos de uso (UC's).

Hipótese Alternativa (H_1 -G): A granularidade dos modelos de classes produzidos com a MIR é menor do que a granularidade dos modelos de classes produzidos com UC's.

Formalmente:

$$H_0\text{-G: } \mu_{G\text{-MIR}} = \mu_{G\text{-UCs}} \text{ e}$$

$$H_1\text{-G: } \mu_{G\text{-MIR}} < \mu_{G\text{-UCs}}$$

onde:

- $\mu_{G\text{-MIR}}$ é a média das granularidades dos modelos de classes obtidos com a MIR; e
- $\mu_{G\text{-UCs}}$ a média das granularidades dos modelos de classes obtidos com UC's.

Para o experimento em questão, onde os participantes 1, 2 e 3 aplicaram a técnica MIR, e os participantes 4, 5 e 6 aplicaram a técnica tradicional de UC's, as médias acima podem ser representadas pelas fórmulas a seguir:

- $\mu_{G-MIR} = \frac{C_1 + C_2 + C_3}{3}$; e
- $\mu_{G-UCs} = \frac{C_4 + C_5 + C_6}{3}$.

4.2.4.2. Hipóteses para Uniformidade de Abstrações

Hipótese Nula (H_0 -Ua): Não há diferença na uniformidade de abstrações dos modelos produzidos com a MIR, em relação àqueles produzidos com a técnica tradicional de UC's.

Hipótese Alternativa (H_1 -Ua): A uniformidade de abstrações dos modelos de classes produzidos com a MIR é maior do que a uniformidade de abstrações dos modelos de classes produzidos com UC's.

Formalmente:

$$H_0\text{-Ua: } \mu_{Ua-MIR} = \mu_{Ua-UCs} \text{ e}$$

$$H_1\text{-Ua: } \mu_{Ua-MIR} > \mu_{Ua-UCs}$$

onde:

- μ_{Ua-MIR} é a média das uniformidades de abstrações entre pares distintos de modelos obtidos com a MIR (pares resultantes da combinação, dois a dois, dos modelos do grupo que aplicou a MIR); e
- μ_{Ua-UCs} a média das uniformidades de abstrações entre pares distintos de modelos obtidos com UC's (pares resultantes da combinação, dois a dois, dos modelos do grupo que aplicou os UC's).

Para o experimento em questão, onde os participantes 1, 2 e 3 aplicaram a técnica MIR, e os participantes 4, 5 e 6 aplicaram a técnica tradicional de UC's, as médias acima podem ser representadas pelas fórmulas a seguir:

- $\mu_{Ua-MIR} = \frac{Unif-A_{1,2} + Unif-A_{1,3} + Unif-A_{2,3}}{3}$; e
- $\mu_{Ua-UCs} = \frac{Unif-A_{4,5} + Unif-A_{4,6} + Unif-A_{5,6}}{3}$.

4.2.4.3. Hipóteses para Uniformidade de Associações

Hipótese Nula (H_0-U_I): Não há diferença na uniformidade de associações dos modelos produzidos com a MIR, em relação àqueles produzidos com a técnica tradicional de UC's.

Hipótese Alternativa (H_1-U_I): A uniformidade de associações dos modelos de classes produzidos com a MIR é maior do que a uniformidade de associações dos modelos de classes produzidos com UC's.

Formalmente:

$$H_0-U_I: \mu_{UI-MIR} = \mu_{UI-UCs} \text{ e}$$

$$H_1-U_I: \mu_{UI-MIR} > \mu_{UI-UCs}$$

onde:

- μ_{UI-MIR} é a média das uniformidades de associações entre pares distintos de modelos obtidos com a MIR (pares resultantes da combinação, dois a dois, dos modelos do grupo que aplicou a MIR); e
- μ_{UI-UCs} é a média das uniformidades de associações entre pares distintos de modelos obtidos com UC's (pares resultantes da combinação, dois a dois, dos modelos do grupo que aplicou os UC's).

Para o experimento em questão, onde os participantes 1, 2 e 3 aplicaram a técnica MIR, e os participantes 4, 5 e 6 aplicaram a técnica tradicional de UC's, as médias acima podem ser representadas pelas fórmulas a seguir:

$$\bullet \mu_{UI-MIR} = \frac{Unif-L_{1,2} + Unif-L_{1,3} + Unif-L_{2,3}}{3}; \text{ e}$$

$$\bullet \mu_{UI-UCs} = \frac{Unif-L_{4,5} + Unif-L_{4,6} + Unif-L_{5,6}}{3}.$$

4.2.4.4. Hipóteses para Uniformidade de Atributos

Hipótese Nula (H_0-U_{at}): Não há diferença na uniformidade de atributos dos modelos de classes produzidos com a MIR, em relação àqueles produzidos com a técnica tradicional de UC's.

Hipótese Alternativa (H_1-U_{at}): A uniformidade de atributos dos modelos de classes produzidos com a MIR é maior do que a uniformidade de atributos dos modelos de classes produzidos com UC's.

Formalmente:

$$H_0\text{-Uat: } \mu_{\text{Uat-MIR}} = \mu_{\text{Uat-Ucs}} \text{ e}$$

$$H_1\text{-Uat: } \mu_{\text{Uat-MIR}} > \mu_{\text{Uat-Ucs}}$$

onde:

- $\mu_{\text{Uat-MIR}}$ é a média das uniformidades de atributos entre pares distintos de modelos obtidos com a MIR (pares resultantes da combinação, dois a dois, dos modelos do grupo que aplicou a MIR); e
- $\mu_{\text{Uat-Ucs}}$ é a média das uniformidades de atributos entre pares distintos de modelos obtidos com UC's (pares resultantes da combinação, dois a dois, dos modelos do grupo que aplicou os UC's).

Para o experimento em questão, onde os participantes 1, 2 e 3 aplicaram a técnica MIR, e os participantes 4, 5 e 6 aplicaram a técnica tradicional de UC's, as médias acima podem ser representada pelas fórmulas a seguir:

- $\mu_{\text{Uat-MIR}} = \frac{\text{Unif-At}_{1,2} + \text{Unif-At}_{1,3} + \text{Unif-At}_{2,3}}{3}$; e
- $\mu_{\text{Uat-Ucs}} = \frac{\text{Unif-At}_{4,5} + \text{Unif-At}_{4,6} + \text{Unif-At}_{5,6}}{3}$.

4.2.4.5. Hipóteses para Uniformidade de Operações

Hipótese Nula ($H_0\text{-Uo}$): Não há diferença na uniformidade de operações dos modelos produzidos com a MIR, em relação àqueles produzidos com a técnica tradicional de UC's.

Hipótese Alternativa ($H_1\text{-Uo}$): A uniformidade de operações entre modelos de classes produzidos com a MIR é maior do que a uniformidade de operações entre modelos de classes produzidos com UC's.

Formalmente:

$$H_0\text{-Uo: } \mu_{\text{Uo-MIR}} = \mu_{\text{Uo-UCs}} \text{ e}$$

$$H_1\text{-Uo: } \mu_{\text{Uo-MIR}} > \mu_{\text{Uo-UCs}}$$

onde:

- μ_{Uo-MIR} é a média das uniformidades de operações entre pares distintos de modelos obtidos com a MIR (pares resultantes da combinação, dois a dois, dos modelos do grupo que aplicou a MIR); e
- μ_{Uo-UCs} é a média das uniformidades de operações entre pares distintos de modelos obtidos com a UC's (pares resultantes da combinação, dois a dois, dos modelos do grupo que aplicou os UC's).

Para o experimento em questão, onde os participantes 1, 2 e 3 aplicaram a técnica MIR, e os participantes 4, 5 e 6 aplicaram a técnica tradicional de UC's, as médias acima podem ser representada pelas fórmulas a seguir:

- $\mu_{Uo-MIR} = \frac{Unif-O_{1,2} + Unif-O_{1,3} + Unif-O_{2,3}}{3}$; e
- $\mu_{Uo-UCs} = \frac{Unif-O_{4,5} + Unif-O_{4,6} + Unif-O_{5,6}}{3}$.

4.2.4.6. Hipóteses para Uniformidade Representacional

Hipótese Nula (H_0-Ur): Não há diferença na uniformidade representacional dos modelos produzidos com a MIR, em relação àqueles produzidos com a técnica tradicional de UC's.

Hipótese Alternativa (H_1-Ur): A uniformidade representacional dos modelos de classes produzidos com a MIR é maior do que a uniformidade representacional dos modelos de classes produzidos com UC's.

Formalmente:

$$H_0-Ur: \mu_{Ur-MIR} = \mu_{Ur-UCs} \text{ e}$$

$$H_1-Ur: \mu_{Ur-MIR} > \mu_{Ur-UCs}$$

onde:

- μ_{Ur-MIR} é a média das uniformidades representacionais entre pares distintos de modelos obtidos com a MIR (pares resultantes da combinação, dois a dois, dos modelos do grupo que aplicou a MIR); e
- μ_{Ur-UCs} é a média das uniformidades representacionais entre pares distintos de modelos obtidos com UC's (pares resultantes da combinação, dois a dois, dos modelos do grupo que aplicou UC's).

Para o experimento em questão, onde os participantes 1, 2 e 3 aplicaram a técnica MIR, e os participantes 4, 5 e 6 aplicaram a técnica tradicional de UC's, as médias acima podem ser representada pelas fórmulas a seguir:

- $\mu_{Ur-MIR} = \frac{Unif-R_{1,2} + Unif-R_{1,3} + Unif-R_{2,3}}{3}$; e
- $\mu_{Ur-UCs} = \frac{Unif-R_{4,5} + Unif-R_{4,6} + Unif-R_{5,6}}{3}$.

4.2.5. Design do Experimento

Trata-se de um experimento do tipo **um fator com dois tratamentos** (WOHLIN *et al.*, 2000), ou seja, uma única variável independente (fator) podendo receber dois valores (tratamentos). Neste experimento, o fator é a técnica para elaboração do modelo de UC's e do diagrama de classes decorrente dos UC's, e os tratamentos são duas técnicas específicas para isso: a técnica tradicional baseada em UC's, e a MIR (ambas descritas no capítulo 3).

Conforme relatado anteriormente, os participantes foram divididos em dois grupos, cada um com três indivíduos, totalizando seis participantes. Cada um destes grupos ficou responsável pela aplicação de uma das técnicas para a elaboração dos UC's e do diagrama de classes correspondente. O número igual de participantes em cada grupo atende ao princípio do **balanceamento** na organização do experimento (capítulo 2).

4.2.6. Descrição da Instrumentação

Para a produção do modelo de classes, é muito importante que os participantes tenham conhecimento do domínio do sistema a ser modelado. Para isso, todos os participantes receberam um sumário das principais funções do sistema a modelar (Anexo B). Além disso, os participantes tiveram livre acesso aos experimentadores¹⁰ durante todo o experimento, podendo tirar suas dúvidas sobre o sistema, no momento em que elas surgiam.

Cada grupo de participantes recebeu um kit contendo todo o material de apoio e um treinamento específico para a técnica a ser aplicada por eles. O material foi entregue alguns dias antes da realização do experimento para que os participantes pudessem se familiarizar com ele, e o treinamento foi ministrado imediatamente antes do início do experimento.

O experimento foi realizado na empresa em que trabalha o autor desta monografia, e pôde contar com a infra-estrutura lá existente. O treinamento foi ministrado em sala própria, com computador, *datashow* e quadro branco. Durante a execução do experimento

¹⁰ O autor desta monografia e seu orientador.

propriamente dito, cada grupo ocupou uma sala distinta, com um computador para cada participante. Os participantes utilizaram a ferramenta *Visual Paradigm*TM (VP, 2007), previamente instalada nos computadores, para a elaboração do modelo de UC's e do modelo de classes do sistema.

4.2.7. Validade

Durante o planejamento do experimento, e tendo por base o arcabouço conceitual apresentado no capítulo 2 sobre a validade de estudos experimentais, foi feita uma análise visando detectar e debelar possíveis ameaças à validade do experimento. Nesta seção é apresentado o resultado dessa análise, para cada tipo de validade: interna, de construção, de conclusão e externa.

Outra questão considerada foi que prioridade deveria ser atribuída a cada tipo de validade, tendo em vista que, muitas vezes, um expediente para favorecer um tipo de validade pode prejudicar outro tipo. Neste aspecto, a prioridade adotada decorreu da sugestão de COOK & CAMPBELL (1979, *apud* WOHLIN *et al.*, 2000) para experimentos com a finalidade de testar teorias. Para essa categoria de experimentos, eles sugerem a seguinte prioridade (da maior para a menor): validade interna, validade de construção, validade de conclusão e validade externa. A análise que se segue é apresentada nesta ordem.

4.2.7.1. Validade Interna

Para evitar que fatores ambientais e externos produzissem efeitos que pudessem ser confundidos com os efeitos dos tratamentos (técnicas empregadas), os tratamentos foram aplicados ao mesmo tempo (um para cada grupo de participantes) e no mesmo ambiente (na empresa em que o autor trabalha). Os participantes de cada grupo ficaram em salas distintas e foram instruídos a não trocar informações entre si. Para garantir a observância dessa regra, cada sala contou com a presença constante de um dos experimentares. A tabela 4.1, a seguir, apresenta o cronograma das atividades realizadas nos dois dias de duração do experimento.

Tabela 4.1 – Cronograma de execução do experimento

SÁBADO – 23/JUNHO/2007			
	Hora	TURMA A (MIR)	TURMA B (UC's)
MANHÃ	8:00 – 9:00	Treinamento UC – Inicial KIT 1	Treinamento UC - Inicial KIT 1
	9:00 – 10:00	Treinamento UC – Suplemento KIT 1a	Experimento Fase 1: Elaboração do MUC ¹¹
	10:00 – 10:15	Coffee Break	Coffee Break
	10:15 – 11:00	Experimento Fase 1: Elaboração do MIR	Experimento Fase 1: Elaboração do UC
	11:00 – 12:00	Experimento Fase 1: Elaboração do MIR	Experimento Fase 1: Elaboração do UC
	12:00 – 13:00	Experimento Fase 1: Elaboração do MIR	Treinamento MUC → MOD KIT 2b
	13:00 – 14:30	Almoço	Almoço
TARDE	14:30 – 15:30	Treinamento MIR → MOD KIT 2a	Experimento Fase 2: Elaboração do MOD
	15:30 – 16:30	Experimento Fase 2: Elaboração do MOD	Experimento Fase 2: Elaboração do MOD
	16:30 – 17:00	Coffee Break	Coffee Break
	17:00 – 18:00	Experimento Fase 2: Elaboração do MOD	Experimento Fase 2: Elaboração do MOD
	18:00 – 18:30	Experimento Fase 2: Elaboração do MOD	---
DOMINGO – 24/JUNHO/2007			
	Hora	TURMA A (MIR)	TURMA B (UC's)
MANHÃ	9:00 – 10:00	Experimento Fase 2: Elaboração do MOD (cont.)	---
	10:00 – 11:00	Experimento Fase 2: Elaboração do MOD	Entrevistas
	11:00 – 12:00	Entrevistas	Entrevistas
	12:00 – 13:30	Almoço	Almoço

4.2.7.2. Validade de Construção

Neste aspecto, o treinamento ministrado aos participantes e o material de apoio que acompanhou o treinamento, visaram garantir um entendimento uniforme das técnicas a serem aplicadas.

4.2.7.3. Validade de Conclusão

Não foi possível obter um número maior de participantes para a execução do experimento. Com isso, a validade de conclusão ficou prejudicada, principalmente pelo fato da análise estatística não poder gerar resultados realmente significativos.

¹¹ Modelo de Casos de Uso

4.2.7.4. Validade Externa

Apesar de todos os participantes pertencerem à população-alvo do experimento, a saber, os profissionais formados em Ciência da Computação pela UFJF, a impossibilidade de se aplicar o princípio da aleatoriedade na seleção da amostra populacional, bem como o tamanho reduzido dela, levantam dúvidas sobre a capacidade dessa amostra em representar a população-alvo do experimento.

4.3. Execução do Experimento

Nesta seção são relatados alguns detalhes relativos à execução do experimento.

A elaboração dos modelos pelos participantes do grupo A (MIR) precisou de mais tempo do que o planejado: os modelos só foram terminados no final da manhã do segundo dia do experimento, ficando as entrevistas com os participantes desse grupo para a parte da tarde desse dia (vide cronograma – tabela 4.1 acima).

A amizade dos participantes com o autor deste estudo foi o principal elemento motivador para a participação. Entretanto, o treinamento ministrado aos participantes procurou despertar neles um interesse real no experimento, procurando fazer com que eles percebessem o experimento como uma oportunidade de aperfeiçoamento no conhecimento e na utilização das técnicas.

Uma vez concluída a elaboração dos modelos pelos participantes, foram realizadas as entrevistas com pares deles, pertencentes a um mesmo grupo. Isto foi feito no dia 23 de junho de 2007 com os pares P1-P3, P4-P5, P4-P6 e P5-P6. Como o participante P2 não conseguiu entregar o modelo de classes no prazo da execução do experimento, mas sim na semana posterior, as entrevistas deste participante com os demais do seu grupo foram feitas em um momento posterior. A entrevista do par P1-P2 ocorreu em 24 de outubro de 2007, e a entrevista do par P2-P3 em 26 de outubro de 2007¹². Essas entrevistas tiveram como finalidade inicial identificar as classes representativas de abstrações comuns entre os modelos do par entrevistado. Uma vez identificadas essas classes, as entrevistas prosseguiram com a identificação dos atributos, operações e associações coincidentes, para as classes comuns.

Como um subsídio adicional para a análise do experimento, os participantes preencheram um questionário (Anexo A) onde se levantou o tempo decorrido deste a graduação na UFJF, a experiência com as técnicas sob investigação, e a experiência com o

¹² O participante P1 reside em outra cidade. Isso causou a demora na marcação das entrevistas.

tipo de sistema modelado no experimento (sistema financeiro). Uma caracterização dos participantes, elaborada com base nas respostas do questionário, é mostrada na tabela 4.2.

Tabela 4.2 – Caracterização dos participantes do experimento

Participante	Tempo desde a formatura	Experiência		
		Técnica MIR	Técnica UC's	Sistemas Financeiros
P1	Dois anos e meio	Sim	Acadêmica e Profissional	Não
P2	Meio ano	Não	Acadêmica	Não
P3	Um ano e meio	Não	Acadêmica e Profissional	Sim
P4	Meio ano	-	Acadêmica	Não
P5	Meio ano	-	Acadêmica	Não
P6	Meio ano	-	Acadêmica	Não

4.4. Análise do Experimento

Na fase de análise de um estudo experimental, duas etapas são importantes: a análise de ameaças (seção 4.4.1) e o teste das hipóteses (seções 4.4.2 e 4.4.3). O presente capítulo termina com uma conclusão sobre os resultados do experimento (seção 4.4.4).

4.4.1. Análise das Ameaças

Na fase de planejamento do experimento foram consideradas precauções a serem tomadas para uma execução adequada do experimento. Nesta seção, são relatadas observações registradas durante a execução do experimento, com potencial de influenciar os resultados do mesmo. São basicamente ameaças à validade interna dos resultados.

Um problema observado foi o entendimento muito variado sobre o sistema a ser modelado e seu domínio, a despeito de um sumário sobre isso ter sido distribuído aos participantes (Anexo B). Alguns modelos produzidos pelos participantes não continham todas as funcionalidades e dados que foram citados no sumário, bem como ocorreu a adição de funcionalidades e dados que não estavam no sumário. Em vista disso, sugere-se que futuros estudos incorporem mecanismos de uniformização da assimilação do conhecimento sobre o sistema e seu domínio, como por exemplo, uma dinâmica de grupo.

O participante P2 não conseguiu entregar os modelos na data prevista. Talvez o treinamento ministrado não tenha sido suficiente para que ele tivesse a mesma desenvoltura que os dois outros participantes do seu grupo. Uma explicação para isso é a sua menor experiência profissional com a técnica de UC's, aliada a um desconhecimento do domínio financeiro (tabela 4.2). Ele completou os modelos nos dias subsequentes, sem

ter contato com os demais colegas do seu grupo, e sob supervisão do autor desta monografia.

Ocorreram dúvidas durante o treinamento e ao longo da construção dos modelos, tanto por participantes do grupo A como por participantes do grupo B. Foi tomado muito cuidado nas respostas, para não interferir no resultado do experimento. Estas eram dúvidas referentes ao domínio do sistema e ao uso das técnicas correspondentes utilizadas no estudo. Para tentar amenizar esse problema, em futuros estudos é recomendável um treinamento mais prolongado e com um maior número de exemplos.

Para análise das uniformidades, algumas correções nos modelos de alguns participantes foram necessárias. No modelo do participante P2 foram encontrados dois erros referentes à interpretação das regras. O primeiro erro foi o fato do mesmo ter aplicado a regra R4d sendo que ele já tinha aplicado a regra R4a, desta maneira gerando operações cujo nome começa com “cadastrar”, que não deveriam estar no modelo. Além do erro de aplicação da regra R4d, o participante P2 não aplicou a regra de consistência, que verifica a utilidade de dados dos fluxos. Isto resultou em um número pequeno de atributos, o que foi percebido pelos experimentadores. Por se tratarem de erros primários, resultantes de falta de atenção na aplicação das regras, o participante P2 fez a revisão do modelo, corrigindo a aplicação da regra R2d e fazendo a verificação correspondente à regra de consistência.

O participante P3 introduziu a operação “Financeiro()” na classe do sistema, sendo que a mesma foi retirada pelo próprio participante durante a entrevista, pois ficou claro para ele que essa operação não é gerada por quaisquer das regras de derivação.

Também durante a entrevista, o participante P1 percebeu que tinha aplicado a regras R2b de forma equivocada, e eliminou algumas das associações (entre classes) que tinha criado com ela.

4.4.2. Teste da Hipótese de Granularidade dos Modelos

A tabela 4.3 (a seguir) mostra os dados coletados sobre a granularidade dos modelos de classes obtidos pelos participantes, com a utilização de cada uma das técnicas investigadas.

Tabela 4.3 – Número de classes por participante

Participante (P _i)	Técnica	Granularidade (Nº de Classes – C _i)
P1	MIR	12
P2	MIR	7
P3	MIR	9
P4	UC's tradicional	13
P5	UC's tradicional	9
P6	UC's tradicional	8

A figura 4.3 reflete os dados da tabela anterior:

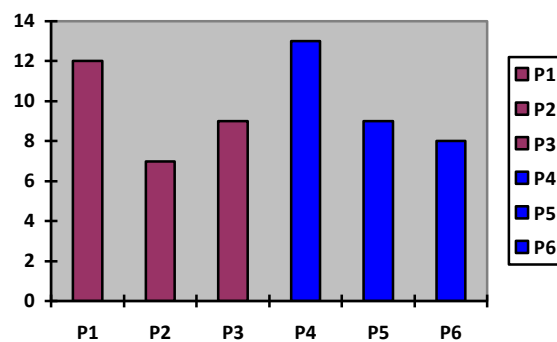


Figura 4.3 – Número de classes no modelo de cada participante

A comparação da granularidade dos modelos de classes foi feita a seguir, comparando-se as médias μ_{G-MIR} e μ_{G-Ucs} . Na seção 4.2.4 vimos que:

- $\mu_{G-MIR} = \frac{C_1 + C_2 + C_3}{3}$; e
- $\mu_{G-Ucs} = \frac{C_4 + C_5 + C_6}{3}$.

onde C_i representa o número de classes existentes no modelo do participante i.

Assim, com base nos valores de C_i da tabela 4.3, tem-se:

$$\mu_{G-MIR} = \frac{12 + 7 + 9}{3} = \frac{28}{3} \cong 9,3333$$

$$\mu_{G-Ucs} = \frac{13 + 9 + 8}{3} = \frac{30}{3} = 10.$$

Portanto, a técnica MIR produziu, em média, modelos com menos classes do que a técnica tradicional de UC's (aproximadamente 8% menos classes).

4.4.3. Teste das Hipóteses de Uniformidade entre Modelos

A tabela 4.4 mostra os dados coletados sobre as abstrações, considerando-se os modelos dos participantes de um mesmo grupo, tomados aos pares.

Tabela 4.4 – Abstrações nos modelos

Paricipantes Pi, Pj	Abstrações no modelo de Pi (Ai)	Abstrações no modelo de Pj (Aj)	Abstrações coincidentes (A'ij)	Unif-Ai,j
P1, P2	12	9	8	0,6153
P1, P3	12	11	10	0,7692
P2, P3	7	9	6	0,6
P4, P5	13	9	7	0,4667
P4, P6	13	8	6	0,4
P5, P6	9	8	5	0,4167

Conforme mencionado anteriormente (seção 4.1.1), é possível que uma abstração representada por uma única classe em um modelo, corresponda a mais de uma classe em outro modelo. Isso foi observado na comparação do modelo de P1 com os modelos dos demais participantes do grupo. Por exemplo, na comparação P1-P2 ficou evidente que a abstração representada pela classe *Pagamento* no modelo de P2, está especializada nas classes *Despesas* e *Crédito Bancário* no modelo de P1. Com isso, a contagem de abstrações no modelo de P2, em comparação com o modelo de P1, considerou a classe “Pagamento” “coincidindo” duas vezes – uma com a classe *Despesas* e outra com a classe *Crédito Bancário*. Algo análogo ocorreu na comparação do modelo de P1 com o modelo de P3. Os números em negrito na tabela 4.4 destacam a variação na contagem de abstrações dos modelos de P2 e P3, na comparação com o modelo de P1, em decorrência desse fenômeno.

A comparação da uniformidade de abstração entre os modelos de classes de cada grupo (ou seja, obtidos com uma mesma técnica) foi feita a seguir, comparando-se as médias μ_{Ua-MIR} e μ_{Ua-UCs} . Na seção 4.2.4 vimos que:

- $\mu_{Ua-MIR} = \frac{Unif-A_{1,2} + Unif-A_{1,3} + Unif-A_{2,3}}{3}$; e
- $\mu_{Ua-UCs} = \frac{Unif-A_{4,5} + Unif-A_{4,6} + Unif-A_{5,6}}{3}$.

onde $\text{Unif-}A_{i,j}$ representa a uniformidade de abstrações entre os modelos produzidos pelos participantes i e j , que aplicaram a mesma técnica (ou seja, pertenceram ao mesmo grupo experimental), sendo dada pela fórmula (seção 4.1.2): $\text{Unif-}A_{i,j} = \frac{A'_{i,j}}{(A_i + A_j) - (A'_{i,j})}$.

Assim, com base nos valores da tabela 4.4, tem-se:

$$\mu_{\text{Ua-MIR}} = \frac{\frac{8}{12+9-8} + \frac{10}{12+11-10} + \frac{6}{7+9-6}}{3} \cong \frac{0,6153 + 0,7692 + 0,6}{3} = 0,6615$$

$$\mu_{\text{Ua-UCs}} = \frac{\frac{7}{13+9-7} + \frac{6}{13+8-6} + \frac{5}{9+8-5}}{3} \cong \frac{0,4667 + 0,4 + 0,4167}{3} = 0,4278$$

Portanto, a uniformidade de abstrações entre modelos de classes produzidos pelo grupo que utilizou a técnica MIR foi, em média, aproximadamente 55% maior do que entre os modelos obtidos com a técnica tradicional de UC's.

Além disso, reforçam esse resultado as constatações de que: 1) a maior de todas as uniformidades de abstrações, obtida entre modelos de participantes de um mesmo grupo, foi obtida aplicando a técnica MIR; e 2) a menor uniformidade de abstrações, entre os pares de modelos construídos com a MIR, é maior do que a maior uniformidade de abstrações, entre os pares de modelos construídos com a outra técnica (UC's).

A tabela 4.5 mostra os dados coletados sobre as associações, considerando-se as classes que representam abstrações coincidentes, entre os modelos dos participantes de um mesmo grupo, tomados aos pares.

Tabela 4.5 – Associações nos modelos

Pares P_i, P_j	Associações no modelo de P_i ($L_{i,i}$)	Associações no modelo de P_j ($L_{j,i}$)	Associações coincidentes ($L'_{i,i}$)	$\text{Unif-}L_{i,j}$
P1, P2	5	6	3	0,375
P1, P3	10	7	6	0,5455
P2, P3	6	4	4	0,6667
P4, P5	4	5	3	0,5
P4, P6	2	7	2	0,2857
P5, P6	2	4	1	0,2

Como visto na comparação de uniformidades das abstrações dos modelos, é possível que associações ocorram entre classes que representam uma abstração em um modelo, e que corresponda a mais de uma classe em outro modelo. Desta maneira, a

associação “Pagamento-FormaPagamento” do modelo do participante P3 “coincidiu” com as associações “Despesa-Parcela” e “Credito-Parcela” do participante P1.

A comparação da uniformidade de associações entre os modelos de classes de cada grupo (ou seja, obtidos com uma mesma técnica) foi feita a seguir, comparando-se as médias μ_{UI-MIR} e μ_{UI-UCs} . Na seção 4.2.4 vimos que:

- $\mu_{UI-MIR} = \frac{Unif-L_{1,2} + Unif-L_{1,3} + Unif-L_{2,3}}{3}$; e
- $\mu_{UI-UCs} = \frac{Unif-L_{4,5} + Unif-L_{4,6} + Unif-L_{5,6}}{3}$.

onde $Unif-L_{i,j}$ representa a uniformidade de associações entre os modelos produzidos pelos participantes i e j, que aplicaram a mesma técnica (ou seja, pertenceram ao mesmo grupo experimental), sendo dada pela fórmula (seção 4.1.2):

$$Unif-L_{i,j} = \frac{L'_{i,j}}{(L_i + L_j) - (L'_{i,j})}.$$

Assim, com base nos valores da tabela 4.5, tem-se:

$$\mu_{UI-MIR} = \frac{\frac{3}{5+6-3} + \frac{6}{10+7-6} + \frac{4}{6+4-4}}{3} \cong \frac{0,375 + 0,5455 + 0,6667}{3} \cong 0,5291$$

$$\mu_{UI-MIR} = \frac{\frac{3}{4+5-3} + \frac{2}{2+7-2} + \frac{1}{2+4-1}}{3} \cong \frac{0,5 + 0,2857 + 0,2}{3} \cong 0,3286$$

Portanto, a uniformidade de associações entre modelos de classes produzidos pelo grupo que utilizou a técnica MIR foi, em média, aproximadamente 61% maior do que entre os modelos obtidos com a técnica tradicional de UC's.

A tabela 4.6 mostra os dados coletados sobre os atributos, considerando-se as classes que representam abstrações coincidentes, entre os modelos dos participantes de um mesmo grupo, tomados aos pares.

Tabela 4.6 – Atributos nos modelos

Pares Pi, Pj	Atributos no modelo de Pi (At _{i,j})	Atributos no modelo de Pj (At _{j,i})	Atributos coincidentes (At' _{i,j})	Unif-At _{i,j}
P1, P2	28	24	15	0,4054
P1, P3	38	24	16	0,3478
P2, P3	24	14	10	0,3571
P4, P5	34	22	14	0,3333
P4, P6	28	33	18	0,4186
P5, P6	24	14	9	0,3103

A comparação da uniformidade de atributos entre os modelos de classes de cada grupo (ou seja, obtidos com uma mesma técnica) foi feita a seguir, comparando-se as médias $\mu_{\text{Uat-MIR}}$ e $\mu_{\text{Uat-UCs}}$. Na seção 4.2.4 vimos que:

- $\mu_{\text{Uat-MIR}} = \frac{\text{Unif-At}_{1,2} + \text{Unif-At}_{1,3} + \text{Unif-At}_{2,3}}{3}$; e
- $\mu_{\text{Uat-UCs}} = \frac{\text{Unif-At}_{4,5} + \text{Unif-At}_{4,6} + \text{Unif-At}_{5,6}}{3}$.

onde Unif-At_{i,j} representa a uniformidade de atributos entre os modelos produzidos pelos participantes i e j, que aplicaram a mesma técnica (ou seja, pertenceram ao mesmo grupo experimental), sendo dada pela fórmula (seção 4.1.2):

$$\text{Unif-At}_{i,j} = \frac{\text{At}'_{i,j}}{(\text{At}_i + \text{At}_j) - (\text{At}'_{i,j})}.$$

Assim, com base nos valores da tabela 4.6, tem-se:

$$\mu_{\text{Uat-MIR}} = \frac{\frac{15}{28+24-15} + \frac{16}{38+24-16} + \frac{10}{24+14-10}}{3} \cong \frac{0,4054 + 0,3478 + 0,3571}{3} = 0,3701$$

$$\mu_{\text{Uat-UCs}} = \frac{\frac{14}{34+22-14} + \frac{18}{28+33-18} + \frac{9}{24+14-9}}{3} \cong \frac{0,3333 + 0,4186 + 0,3103}{3} = 0,3541$$

Portanto, a uniformidade de atributos entre modelos de classes produzidos pelo grupo que utilizou a técnica MIR foi, em média, aproximadamente 5% maior do que entre os modelos obtidos com a técnica tradicional de UC's.

A tabela 4.7 mostra os dados coletados sobre as operações, considerando-se as classes que representam abstrações coincidentes, entre os modelos dos participantes de um mesmo grupo, tomados aos pares.

Tabela 4.7 – Operações nos modelos

Pares Pi, Pj	Operações no modelo de Pi (O _{i,j})	Operações no modelo de Pj (O _{i,j})	Operações coincidentes (O' _{i,j})	Unif-O _{i,j}
P1, P2	13	12	9	0,5625
P1, P3	15	15	14	0,875
P2, P3	15	13	11	0,6471
P4, P5	27	16	7	0,1944
P4, P6	22	19	5	0,1389
P5, P6	14	14	2	0,0769

A MIR determina a criação de uma operação construtora em cada classe do modelo, com o mesmo nome da classe. Cada operação construtora gerou uma coincidência de operações. A técnica tradicional não determina a introdução de operações construtoras, embora pudesse fazê-lo da mesma forma que na MIR. Por exemplo, o participante P5 não incluiu nenhuma operação construtora em suas classes. Em vista disso, decidiu-se calcular também a uniformidade de operações sem levar em conta as operações construtoras. A tabela 4.8 apresenta os dados coletados sobre as operações quando não são computadas as operações construtoras nos modelos de ambos os grupos.

Tabela 4.8 – Operações nos modelos, excluídas as operações construtoras

Pares Pi, Pj	Operações no modelo de Pi (O _{i,j})	Operações no modelo de Pj (O _{i,j})	Operações coincidentes (O' _{i,j})	Unif-O _{i,j}
P1, P2	7	7	3	0,2727
P1, P3	7	7	6	0,75
P2, P3	10	8	6	0,5
P4, P5	24	16	7	0,2121
P4, P6	17	15	1	0,0323
P5, P6	14	12	2	0,0833

A comparação da uniformidade de operações entre os modelos de classes de cada grupo (ou seja, obtidos com uma mesma técnica) foi feita a seguir, comparando-se as médias μ_{U0-MIR} e μ_{U0-UCs} . Na seção 4.2.4 vimos que:

- $\mu_{U0-MIR} = \frac{Unif-O_{1,2} + Unif-O_{1,3} + Unif-O_{2,3}}{3}$; e
- $\mu_{U0-UCs} = \frac{Unif-O_{4,5} + Unif-O_{4,6} + Unif-O_{5,6}}{3}$.

onde Unif-O_{i,j} representa a uniformidade de operações entre os modelos produzidos pelos participantes i e j, que aplicaram a mesma técnica (ou seja, pertenceram ao mesmo

grupo experimental), sendo dada pela fórmula (seção 4.1.2): $\text{Unif-O}_{ij} = \frac{O'_{ij}}{(O_i + O_j) - (O'_{ij})}$.

Primeiramente, com base nos valores da tabela 4.7, tem-se:

$$\mu_{\text{Uo-MIR}} = \frac{\frac{9}{13+12-9} + \frac{14}{15+15-14} + \frac{11}{15+13-11}}{3} \cong \frac{0,5625 + 0,875 + 0,6471}{3} \cong 0,6949$$

$$\mu_{\text{Uo-UCs}} = \frac{\frac{7}{27+16-7} + \frac{5}{22+19-5} + \frac{2}{14+14-2}}{3} \cong \frac{0,1944 + 0,1389 + 0,0769}{3} \cong 0,1367$$

E agora, com base nos valores da tabela 4.8, que desconsidera as operações construtoras, tem-se

$$\mu_{\text{Uo-MIR}} = \frac{\frac{3}{7+7-3} + \frac{6}{7+7-6} + \frac{6}{10+8-6}}{3} \cong \frac{0,2727 + 0,75 + 0,5}{3} \cong 0,5076$$

$$\mu_{\text{Uo-UCs}} = \frac{\frac{7}{24+16-7} + \frac{1}{17+15-1} + \frac{2}{14+12-2}}{3} \cong \frac{0,2121 + 0,0323 + 0,0833}{3} \cong 0,1092$$

Portanto, a uniformidade de operações entre modelos de classes produzidos pelo grupo que utilizou a técnica MIR foi 408% maior do que entre os os modelos obtidos com a técnica tradicional de UC's (ou 365%, se forem desconsideradas as operações construtoras).

Por fim, a tabela 4.9 consolida os dados sobre a uniformidade de associações, atributos, operações e representacional, considerando-se os modelos dos participantes de um mesmo grupo, tomados aos pares.

Tabela 4.9 – Uniformidade de associações, atributos, operações e representacional

Pares Pi, Pj	Unif- L _{i,j}	Unif- At _{i,j}	Unif-O _{i,j}		Unif-R _{i,j}	
			(1)	(2)	(1)	(2)
P1, P2	0,375	0,4054	0,5625	0,2727	0,4476	0,3510
P1, P3	0,5455	0,3478	0,875	0,75	0,5894	0,5478
P2, P3	0,6667	0,3571	0,6471	0,5	0,5570	0,5079
P4, P5	0,5	0,3333	0,1944	0,2121	0,3426	0,3485
P4, P6	0,2857	0,4186	0,1389	0,0323	0,2811	0,2455
P5, P6	0,2	0,3103	0,0769	0,0833	0,1957	0,1979

- (1) Considerando operações construtoras
- (2) Não considerando operações construtoras

A comparação da uniformidade representacional entre os modelos de classes de cada grupo (ou seja, obtidos com uma mesma técnica) foi feita a seguir, comparando-se as médias μ_{Ur-MIR} e μ_{Ur-UCs} . Na seção 4.2.4 vimos que:

- $\mu_{Ur-MIR} = \frac{Unif-R_{1,2} + Unif-R_{1,3} + Unif-R_{2,3}}{3}$; e
- $\mu_{Ur-UCs} = \frac{Unif-R_{4,5} + Unif-R_{4,6} + Unif-R_{5,6}}{3}$.

onde $Unif-R_{i,j}$ representa a uniformidade representacional entre os modelos produzidos pelos participantes i e j, que aplicaram a mesma técnica (ou seja, pertenceram ao mesmo grupo experimental), sendo dada pela fórmula (seção 4.1.2):

$$Unif-R_{i,j} = \frac{Unif-L_{i,j} + Unif-At_{i,j} + Unif-O_{i,j}}{3}.$$

Primeiramente, com base nos valores da tabela 4.9, e considerando todas as operações, tem-se:

$$\mu_{Ur-MIR} = \frac{0,4476 + 0,5894 + 0,5570}{3} \cong 0,5313$$

$$\mu_{Ur-UCs} = \frac{0,3426 + 0,2811 + 0,1957}{3} \cong 0,2731$$

E agora, com base nos valores da tabela 4.9, e desconsiderando as operações construtoras, tem-se:

$$\mu_{Ur-MIR} = \frac{0,3510 + 0,5478 + 0,5079}{3} = 0,4689$$

$$\mu_{Ur-UCs} = \frac{0,3485 + 0,2455 + 0,1979}{3} = 0,2640$$

Portanto, a uniformidade representacional entre modelos de classes produzidos pelo grupo que utilizou a técnica MIR foi, em média, aproximadamente 95% maior do que entre os modelos obtidos com a técnica tradicional de UC's (ou 78%, se forem desconsideradas as operações construtoras).

4.4.4. Conclusão Geral sobre os Resultados do Experimento

A análise dos dados colhidos no experimento produziu evidências de que:

- 1) A técnica MIR produz modelos de classes de menor granularidade, em comparação aos modelos de classes obtidos com a técnica baseada nos UC's tradicionais. Em média, os modelos da MIR incluíram, aproximadamente, 8% menos classes.
- 2) A técnica MIR produz modelos de classes mais uniformes entre si, do que a técnica baseada nos UC's tradicionais. Isso foi verificado no nível semântico (conceitual) pela comparação das abstrações existentes nos modelos de classes, bem como no nível representacional, pela comparação dos atributos, associações e operações, envolvidos nas classes correspondentes às abstrações coincidentes entre (pares de) modelos. A tabela 4.10 dá uma visão geral desses resultados.

Tabela 4.10 - Comparação das médias de uniformidade

Técnica	Uniformidade de abstrações (μ_{Ua})		Uniformidade Representacional							
			Associações (μ_{Ul})		Atributos (μ_{Uat})		Operações (μ_{Uo})		Global (μ_{Uf})	
MIR	0,6615	+ 55%	0,5291	+61%	0,3701	5%	0,6949 ⁽¹⁾ 0,5076 ⁽²⁾	+408% ⁽¹⁾ +365% ⁽²⁾	0,5313 ⁽¹⁾ 0,4689 ⁽²⁾	+95% ⁽¹⁾ +78% ⁽²⁾
UC's	0,4278	-	0,3286	-	0,3541	-	0,1367 ⁽¹⁾ 0,1092 ⁽²⁾	-	0,2731 ⁽¹⁾ 0,2640 ⁽²⁾	-

(1) Considerando operações construtoras

(2) Não considerando operações construtoras

Capítulo 5

Conclusão

O presente trabalho envolveu a realização de um experimento em Engenharia de Software. Através dele, procurou-se obter evidências sobre hipóteses enunciadas para comparar a granularidade e a uniformidade dos modelos de classes construídos com o auxílio de duas técnicas: a técnica tradicional baseada em casos de uso (UC's), e a técnica da Modelagem Informacional de Requisitos (MIR), que é uma especialização da primeira. Após uma revisão dos principais conceitos na área da Engenharia de Software Experimental (capítulo 2) e das técnicas empregadas (capítulo 3), o capítulo 4 descreveu o planejamento e a execução do experimento, bem como seus resultados e conclusões deles advindas.

Entre os principais resultados da análise dos dados coletados no experimento, estão as evidências de que a MIR produz modelos de classe mais uniformes entre si, do que a técnica tradicional de UC's. Apesar da amostra populacional ser reduzida (apenas 6 indivíduos), e do fato de não se ter utilizado métodos estatísticos para o teste das hipóteses, este resultado não deixa de ser importante como mais um resultado capaz de lançar luz sobre o problema da comparação entre as duas técnicas utilizadas neste experimento.

Outros resultados secundários, mas também importantes, foram obtidos. Ficou claro para o autor deste trabalho que, embora trabalhosos, os estudos experimentais são muito importantes para a compreensão e a avaliação de técnicas e modelos da Engenharia de Software. É verdade que nenhum estudo dá certeza absoluta sobre seus resultados, devido a grande dificuldade de se controlar os efeitos de todos os fatores que não são de interesse no experimento. Porém, com base nesse tipo de estudo, é possível se afirmar algo com mais segurança, fundamentando os argumentos em resultados observados e medidos, e não simplesmente desejados ou teóricos.

Como trabalho futuro, fica a sugestão de se repetir o experimento com um número maior de participantes, introduzindo as alterações sugeridas na análise de ameaças (seção 4.4.1), e o uso de métodos estatísticos para a análise dos dados coletados e para o teste das hipóteses. Além disso, seria interessante prosseguir na implementação de pelo menos dois dos modelos produzidos, sendo um de cada técnica, para uma avaliação mais completa desses modelos face ao resultado final almejado, que é o sistema funcionando e atendendo as necessidades dos usuários.

Referências Bibliográficas

BARROS, M. O; WERNER, C. M; TRAVASSOS, G. H. **Um Estudo Experimental sobre a Utilização de Modelagem e Simulação no Apoio à Gerência de Projetos de Software**. In: XVI Simpósio Brasileiro de Engenharia de Software, Gramado, RS. 2002.

BITTNER, K; SPENCER, I. **Use case Modeling**. 1.ed. Boston: Addison Wesley, 2002.

BOOCH, G; RUMBAUCH, J; JACOBSON, I. **UML: Guia do Usuário**. Tradução: Fábio Freitas. 2. ed. Rio de Janeiro: Campus, 2000.

COAD, P., YOURDON, E. **Object Oriented Analysis**. 2.ed. NJ: Prentice Hall, 1990.

FORTUNA, M., BORGES, M.R.S. **Modelagem Informacional de Requisitos. VIII Workshop on Requirements Engineering**, Porto, Portugal, 2005.

FORTUNA, M; WERNER, C. M.; BORGES, M.R.S. **Um Modelo Integrado de Requisitos com Casos de Uso**. In: Workshop Iberoamericano de Ingeniería de Requisitos y Ambientes de Software, Isla Margarita, Venezuela, 2007.

JACOBSON, I., CHRISTERSON, M., JONSSON, P., ÖVERGAARD, G.: **Object-Oriented Software Engineering - A Use Case Driven Approach**. Addison-Wesley, 1992.

JURISTO, N; MORENO, A. M. **Basics of Software Engineering Experimentation**. 1. ed. Boston: The Kluwer Academic Publishers, 2000.

LARMAN. C. Applying UML and Patterns: **An Introduction to Object-Oriented Analysis and Design and Iterative Development** (3rd ed.), Prentice Hall, 2004.

RODRIGUES, L. G. M. **Um modelo de avaliação de requisitos no processo de desenvolvimento de software**. Campinas, 2006. 74 pg. Trabalho final (mestrado profissional). Instituto de Computação, Universidade Estadual de Campinas.

SVETINOVIC, D., BERRY, D.M., GODFREY, M. **Concept Identification in Object-Oriented Domain Analysis: Why Some Students Just Don't Get It**, In: 13th IEEE Intl. Conference on Requirements Engineering (RE'05), pp. 189-198, Paris, França, Setembro, 2005.

TRAVASSOS, G.H; GUROV, D.; AMARAL, E.A.G.G. **Introdução à Engenharia de Software Experimental**. Rio de Janeiro: COPPE/UFRJ, 2002. (Relatório Técnico ES-590/02).

VISUAL PARADIGM INTERNATIONAL. Visual Paradigm. Disponível em: <<http://www.visual-paradigm.com/>>. Acesso em: 18 dez. 2007.

WIRFS-BROCK, R., WILKERSON, B. & WIENER, L. **Designing Object Oriented Software**. Englewood Cliffs, NJ. Prentice Hall, 1990.

WOHLIN, C; RUNESON, P; HÖST, M; OHLSSON, M.C; REGNELL, B.; WESSLÉN, A.
Experimentation In Software Engineering: An Introduction. 1.ed. Boston: The Kluwer
Academic Publishers, 2000.

Anexo A

Questionário de Caracterização dos Participantes

Este questionário foi aplicado logo após o término do experimento, para uma melhor caracterização dos participantes.

<u>Questionário</u>
<u>Experiência prévia no domínio:</u>
1) Você já teve experiência na modelagem de um sistema parecido com o sistema FGV (contas a pagar/receber)? Descreva (número de sistemas, porte, tempo dedicado).
2) Mesmo não tendo modelado, como avalia o seu conhecimento/experiência anterior sobre este tipo de sistema?
<u>Experiência prévia na técnica (casos de uso):</u>
1) Você tem experiência na modelagem de casos de uso (elaboração, leitura)? Essa experiência foi apenas acadêmica (estudo ou trabalho escolar) ou também profissional (sistema real)?
2) Você tem experiência na modelagem <u>informacional</u> com casos de uso? (apenas grupo A) Descreva brevemente.
<u>Experiência profissional:</u>
1) Descreva sua experiência profissional (empresa, função, tempo na função).
<u>Formação:</u>
2) Qual é o seu nível de escolaridade? Descreva. Participou de treinamento sobre casos de uso? Com que carga horária?

Anexo B

Sumário do Sistema Desenvolvido (FGV)

O seguinte sumário foi distribuído aos participantes, no início do experimento, para servir de base para a modelagem do sistema.

Deseja-se um sistema computadorizado para melhorar o seu controle financeiro da empresa. Entre outras coisas, a empresa deseja:

- Prever e acompanhar a entrada e saída de dinheiro decorrente do recebimento dos clientes e do pagamento de contas a credores da empresa. Ou seja, para cada dia do mês ela precisa saber o que deve pagar e o que tem a receber até aquele dia. O principal objetivo é prever eventuais dificuldades de caixa, para tomar providências (por exemplo, fazer um empréstimo) em tempo hábil.
- Controlar o saldo em caixa, bem como o saldo e a movimentação de suas contas bancárias.
- Controlar as despesas efetuadas (pagamentos) por credor e item de despesa. O objetivo é ter uma gestão permanente do quanto se gasta com cada credor e em cada item de despesa, buscando maximizar os benefícios obtidos com os recursos gastos, face as metas da empresa. A quitação pode ser feita de uma só vez ou dividida em parcelas, nas respectivas datas de vencimento acordadas.
- Controlar os recebimentos de clientes, por serviço prestado. A empresa tem um leque atual de serviços que disponibiliza a seus clientes, que pode mudar ao longo do tempo. A maioria dos clientes tem contrato com a empresa, estipulando os serviços contratados e o valor a ser pago pelo conjunto dos serviços. Mesmo clientes sem contrato podem usufruir os serviços da empresa, pagando o valor-padrão desses serviços, ou um valor especial decorrente de negociação caso a caso. A quitação pode ser feita de uma só vez ou dividido em parcelas, nas respectivas datas de vencimento acordadas. Os contratos têm renovação automática a cada 1 ano contado da data da contratação, podendo ser renegociados ou cancelados a qualquer instante, mediante solicitação prévia do cliente, com pelo menos 2 meses de antecedência. É preciso manter um histórico das renovações e renegociações de contratos.

A empresa deve poder consultar, a qualquer instante, para uma data ou período escolhido:

- Recebimentos realizados e a realizar, de todos os clientes ou de um específico, correspondente a um determinado serviço ou a todos eles. Os recebimentos atrasados (fora do prazo acordado) deverão estar destacados, para uma eventual cobrança ao cliente (daí a importância de manter informações de contato com o cliente: endereço, pessoa de contato, telefone, e-mail). Recebimentos fora do prazo estão sujeitos a multa e juros, conforme estipulado no contrato ou através de acordo.
- Pagamentos realizados e a realizar, para todos os credores ou para um específico, e associados a um determinado item de despesa ou a todos eles. Os pagamentos atrasados (fora do prazo acordado) deverão estar destacados, para que possam ser priorizados e/ou renegociados (daí a importância de manter informações de contato com o credor: endereço, pessoa de contato, telefone, e-mail). Deverão ser apresentadas informações que permitam ou facilitem o pagamento, tais como: nome

do credor, forma de pagamento, valor a pagar, pessoa de contato, endereço do banco/agência ou do estabelecimento onde efetuar o pagamento, etc.

- Montante a receber e montante a pagar, bem como saldo inicial e final disponível em suas contas bancárias e no caixa, dia-a-dia, dentro do período escolhido.

Anexo C

Modelo de Casos de Uso do Participante P3

Este anexo apresenta, como exemplo, o modelo de casos de uso produzido pelo participante P3 (Grupo A - técnica informacional), para o sistema FGV.

CASOS DE USO INFORMACIONAIS:

Nr.	Ator	Nome do UC
1	Gerente Financeiro	Consultar recebimentos realizados e a realizar
2	Gerente Financeiro	Consultar pagamentos realizados e a realizar
3	Gerente Financeiro	Consultar montantes a receber e a pagar
4	Gerente Financeiro	Consultar saldo
5	Gerente Financeiro	Cadastrar clientes
6	Gerente Financeiro	Cadastrar serviços
7	Gerente Financeiro	Cadastrar credores
8	Gerente Financeiro	Cadastrar itens de despesa
9	Gerente Financeiro	Controlar pagamentos
10	Gerente Financeiro	Controlar recebimentos
11	Gerente Financeiro	Renegociar contratos
12	Gerente Financeiro	Cadastrar Forma de Pagamento
13	Gerente Financeiro	Cadastrar Conta

DESCRIÇÃO DOS UC'S:

Ator: **Gerente Financeiro** UC1: **Consultar recebimentos realizados e a realizar**

1. O Gerente Financeiro informa ao sistema a necessidade de consultar os **recebimentos realizados e a realizar**, filtrando por um determinado período, por **clientes** e por **serviços**;
2. O sistema solicita os parâmetros do filtro, podendo ser um ou vários **clientes** e um ou vários **serviços**, num determinado período;
3. O Gerente Financeiro informa os parâmetros do filtro
→ $\text{cons_recebimentos} = \text{periodo_cons} + _1\{\text{id_cliente}\} + _1\{\text{id_servico}\} + \text{estado_recebimento}$
☞ $\text{periodo_cons} = \text{dt_inicio} + \text{dt_fim}$
4. O sistema apresenta os dados sobre os recebimentos no período escolhido;
← $\text{lista_recebimentos} = \text{estado_recebimento} + \text{dt_recebimento} + \text{id_cliente} + \text{nome_cliente} + \text{endereco_cliente} + \text{tel_cliente} + \text{email_cliente} + \text{id_servico} + \text{nome_servico} + \text{valor_servico} + \text{valor_recebimento} + \text{atrasado_ou_nao}$

Ator: **Gerente Financeiro**

UC2: Consultar pagamentos realizados e a realizar

1. O Gerente Financeiro informa ao sistema a necessidade de consultar os **pagamentos realizados e a realizar**, filtrando por um determinado período, por **credores** e por **itens de despesa**;
2. O sistema solicita os parâmetros do filtro, podendo ser um ou vários **credores** e um ou vários **itens de despesa**, num determinado período;
3. O Gerente Financeiro informa os parâmetros do filtro
→ cons_pagamentos = periodo_cons + ₁{id_credor} + ₁{id_item_despesa} + estado_pagamento
☞ periodo_cons = dt_inicio + dt_fim
4. O sistema apresenta os dados sobre os recebimentos no período escolhido;
← lista_pagamentos = estado_pagamento + dt_pagamento + id_credor + nome_credor + endereco_credor + tel_credor + email_credor + id_item_despesa + valor_item_despesa + nome_item_despesa + valor_pagamento + atrasado_ou_nao

Ator: **Gerente Financeiro**

UC3: Consultar montantes a receber e a pagar

1. O Gerente Financeiro informa ao sistema sua intenção de consultar os **montantes a receber e a pagar**;
→ cons_montantes = periodo_consulta
☞ periodo_cons = dt_inicio + dt_fim + estado_montante
2. O sistema devolve o valor dos **montantes** no período determinado e os **montantes** discriminados a cada dia;
← lista_montantes = estado_montante + dt_dia + valor_montante

Ator: **Gerente Financeiro**

UC4: Consultar saldo

1. O Gerente Financeiro informa ao sistema sua intenção de consultar o saldo atual no caixa ou nas contas bancárias;
→ cons_saldo_dia = dt_saldo + [id_conta]
2. O sistema devolve o valor do saldo naquele dia
← lista_saldo_dia = dt_saldo + valor_saldo

Ator: **Gerente Financeiro**

UC5: Cadastrar Clientes

1. O Gerente Financeiro informa ao sistema sua intenção de cadastrar um novo **cliente**;
2. O sistema solicita as informações sobre o **cliente** (nome, endereço, telefone, e-mail);
3. O Gerente Financeiro fornece as informações para o cadastro;
→ cliente = nome_cliente + endereco_cliente + tel_cliente + email_cliente + possui_contrato
4. O sistema efetua o cadastro do **cliente**
← id_cliente

Ator: **Gerente Financeiro** UC6: **Cadastrar Serviços**

1. O Gerente Financeiro informa ao sistema sua intenção de cadastrar ou alterar um **serviço**;
2. O sistema solicita as informações sobre o **serviço**(nome, valor);
3. Gerente Financeiro fornece as informações para o cadastro;
→ serviço = nome_servico + valor_servico
4. O sistema efetua o cadastro do **serviço**
← id_servico

Ator: **Gerente Financeiro** UC7: **Cadastrar credores**

1. O Gerente Financeiro informa ao sistema sua intenção de cadastrar um novo **credor**;
2. O sistema solicita as informações sobre o **credor**(nome, endereço, telefone, e-mail);
3. O Gerente Financeiro fornece as informações para o cadastro;
→ credor = nome_credor + endereco_credor + tel_credor + email_credor
4. O sistema efetua o cadastro do **credor**
← id_credor

Ator: **Gerente Financeiro** UC8: **Cadastrar itens de despesa**

1. O Gerente Financeiro informa ao sistema sua intenção de cadastrar ou alterar um **item de despesa**;
2. O sistema solicita as informações sobre o **item de despesa**(nome, valor);
3. O Gerente Financeiro fornece as informações para o cadastro;
→ item_despesa = nome_item_despesa + valor_item_despesa
4. O sistema efetua o cadastro do **item de despesa**
← id_item_despesa

Ator: **Gerente Financeiro** UC9: **Controlar pagamentos**

1. O Gerente Financeiro informa ao sistema sua intenção de quitar uma dívida com um **credor**;
2. O sistema solicita o nome do **credor** ao qual se deve pagar, qual o item de despesa a ser pago, o valor a pagar e a forma de pagamento;
3. O Gerente Financeiro fornece as informações para lançar a dívida;
→ pagamento = id_credor + ₁{id_item_despesa + valor_item_despesa} + valor_pagamento + estado_recebimento + forma_pagamento + id_conta + nome_banco + num_agencia + num_conta
☞ forma_pagamento = ₁{ id_forma_pagamento + num_parcela + valor_parcela + dt_pagamento}
4. Caso a forma de pagamento seja para a data atual, o sistema registra a quitação da dívida mudando o estado do pagamento para realizado;
Caso a forma de pagamento seja em parcelas a serem pagas futuramente, o sistema registra o lançamento mudando o estado do pagamento para a realizar;
← id_pagamento + estado_pagamento

Ator: **Gerente Financeiro**

UC10: **Controlar recebimentos**

1. O Gerente Financeiro informa ao sistema sua intenção de quitar um recebimento de um **cliente**;
2. O sistema solicita o nome do **cliente** do qual se deve receber, qual **serviço** foi prestado, o valor a receber e a forma de pagamento;
3. O Gerente Financeiro fornece as informações para lançar o recebimento;
→ recebimento = id_cliente + ₁{id_servico + valor_servico} + valor_recebimento + estado_recebimento + forma_pagamento + nome_banco + num_agencia + num_conta
☞ forma_pagamento = ₁{id_forma_pagamento + num_parcela + valor_parcela + dt_pagamento}
4. Caso a forma de recebimento seja para a data atual, o sistema registra o lançamento mudando o estado do recebimento para realizado;
Caso a forma de pagamento seja em parcelas a serem recebidas futuramente, o sistema registra o lançamento mudando o estado do recebimento para a realizar;
← id_recebimento + estado_recebimento

Ator: **Gerente Financeiro**

UC11: **Renegociar contratos**

1. O Gerente Financeiro informa ao sistema sua intenção de renegociar um contrato com o **cliente** contratado;
2. O sistema solicita o nome do **cliente** e o contrato a ser renegociado;
3. O Gerente Financeiro fornece as informações para gerar um novo contrato de recebimento para esse **cliente**;
→ recebimento = id_cliente + ₁{id_servico + valor_servico} + valor_recebimento + estado_recebimento + forma_pagamento + nome_banco + num_agencia + num_conta
☞ forma_pagamento = ₁{ id_forma_pagamento + num_parcela + valor_parcela + dt_pagamento}
4. Caso a forma de recebimento seja para a data atual, o sistema registra o lançamento mudando o estado do recebimento para realizado;
Caso a forma de pagamento seja em parcelas a serem recebidas futuramente, o sistema registra o lançamento mudando o estado do recebimento para a realizar;
← id_recebimento + estado_recebimento
5. O sistema cancela o lançamento antigo;
6. O sistema registra o histórico da renegociação;
7. Caso o contrato tenha um ano da data de contratação, o sistema renovará esse contrato automaticamente e manterá um histórico dessa renovação

Ator: **Gerente Financeiro**

UC12: **Cadastrar forma de pagamento**

1. O Gerente Financeiro informa ao sistema sua intenção de cadastrar uma **forma de pagamento**;
2. O sistema solicita as informações sobre a **forma de pagamento**(numero de parcelas, valor da parcela, data do pagamento);
3. O Gerente Financeiro fornece as informações para o cadastro;
→ forma_pagamento = num_parcela + valor_parcela + dt_pagamento
4. O sistema efetua o cadastro do **item de despesa**

← id_forma_pagamento

Ator: **Gerente Financeiro** UC13: **Cadastrar Conta**

1. O Gerente Financeiro informa ao sistema sua intenção de cadastrar uma nova conta;
2. O sistema solicita as informações sobre a conta (banco , agencia, numero da conta);
3. O Gerente Financeiro fornece as informações para o cadastro;
→ conta = nome_banco + num_agencia + num_conta + valor_saldo
4. O sistema efetua o cadastro da conta
← id_conta

GLOSSÁRIO

Termo	Descrição	Informação Adicional
Pagamento (ou recebimento) realizado	É um lançamento que já foi quitado.	
Pagamento (ou recebimento) a realizar	É um lançamento que deverá ainda ser quitado em uma data futura	
Cliente	É o stakeholder responsável por consumir os serviços disponibilizados pela empresa	
Serviço	Serviço que a empresa vende para os clientes a fim de gerar faturamento	
Credor	Entidade fornecedora de despesas para a empresa	
Item de despesa	Qualquer entidade que representa um custo para a empresa. Ex: água, luz, telefone, salários, etc...	
Montantes a receber e a pagar	Valor total de todos lançamentos de entrada(receber) e saída(pagar) de dinheiro	

DICIONÁRIO DE ITENS ELEMENTARES

UC1 Consultar recebimentos realizados e a realizar

Item	Descrição	Tipo-base ¹³	Domínio
id_cliente	Identificador do cliente	Nr. Automático	
id_servico	Identificador do serviço	Nr. Automático	
nome_cliente	Nome de um cliente	Texto	
nome_servico	Nome de um serviço	Texto	
dt_inicio	Data a partir da qual será feita a consulta	Data	

¹³ Número, Nr. automático, Nr. natural, Nr. decimal, Nr. inteiro, Data, Hora, Moeda, Email, Percentual, Sim/Não, Telefone, Texto, URL.

Item	Descrição	Tipo-base ¹³	Domínio
dt_fim	Data até onde será feita a consulta	Data	
estado_recebimento	Indica se é um recebimento a realizar ou realizado	Nr. Natural	0 – a realizar 1 – realizado
dt_recebimento	Data em que deve ser cobrado o recebimento	Data	
valor_recebimento	Valor a ser cobrado pelo recebimento	Moeda	
atrasado_ou_nao	Indica se o recebimento está ou não atrasado	Sim/Não	

UC2 Consultar pagamentos realizados e a realizar

Item	Descrição	Tipo-base	Domínio
id_credor	Identificador do credor	Nr. Automático	
id_item_despesa	Identificador do item de despesa	Nr. Automático	
nome_credor	Nome de um credor	Texto	
nome_item_despesa	Nome de um item de despesa	Texto	
dt_inicio	Data a partir da qual será feita a consulta	Data	
dt_fim	Data até onde será feita a consulta	Data	
estado_pagamento	Indica se é um pagamento a realizar ou realizado	Nr. Natural	0 – a realizar 1 – realizado
dt_pagamento	Data em que deve ser quitado o pagamento	Data	
valor_pagamento	Valor a ser pago pela despesa	Moeda	
atrasado_ou_nao	Indica se o recebimento está ou não atrasado	Sim/Não	

UC3 Consultar montantes a receber e a pagar

Item	Descrição	Tipo-base	Domínio
dt_inicio	Data a partir da qual será feita a consulta	Data	
dt_fim	Data até onde será feita a consulta	Data	
estado_montante	Indica se é um montante a receber ou a pagar	Nr. Natural	0 – a receber 1 – a pagar
dt_dia	Data do dia o qual será discriminado o montante	Data	

Item	Descrição	Tipo-base	Domínio
valor_montante	Valor total do montante por cada dia e por todos os dias	Moeda	

UC4 Consultar saldo

Item	Descrição	Tipo-base	Domínio
dt_saldo	Data de consulta do saldo	Data	Data atual
valor_saldo	Valor do saldo	Moeda	

UC5 Cadastrar clientes

Item	Descrição	Tipo-base	Domínio
nome_cliente	Nome do cliente	Texto	
endereço_cliente	Endereço do cliente	Texto	
tel_cliente	Telefone do cliente	Texto	
email_cliente	E-mail do cliente	Texto	
id_cliente	Identificador do cliente	Nr. Automático	

UC6 Cadastrar serviços

Item	Descrição	Tipo-base	Domínio
nome_servico	Nome do serviço	Texto	
valor_servico	Valor padrão a ser cobrado pelo serviço	Moeda	
id_servico	Identificador do serviço	Nr. Automático	

UC7 Cadastrar credores

Item	Descrição	Tipo-base	Domínio
nome_credor	Nome do credor	Texto	
endereço_credor	Endereço do credor	Texto	
tel_credor	Telefone do credor	Texto	
email_credor	E-mail do credor	Texto	
id_credor	Identificador do credor	Nr. Automático	

UC8 Cadastrar itens de despesa

Item	Descrição	Tipo-base	Domínio
nome_item_despesa	Nome do item de despesa	Texto	
valor_item_despesa	Valor a ser pago pelo item de despesa	Moeda	
id_item_despesa	Identificador do item de despesa	Nr. Automático	

UC9 Controlar pagamentos

Item	Descrição	Tipo-base	Domínio
id_credor	Identificador do credor	Nr. Automático	

Item	Descrição	Tipo-base	Domínio
id_item_despesa	Identificador do item de despesa	Nr. Automático	
valor_item_despesa	Valor do item de despesa	Moeda	
valor_pagamento	Soma total dos valores de todas parcelas	Moeda	
num_parcela	Número da parcela	Nr. Natural	
valor_parcela	Valor da parcela	Moeda	
dt_pagamento	Data do pagamento da parcela	Data	
id_pagamento	Identificador do pagamento	Nr. Automático	
estado_pagamento	Status do pagamento	Nr. Natural	0 – realizado 1 – a realizar

UC10 Controlar recebimentos

Item	Descrição	Tipo-base	Domínio
id_cliente	Identificador do cliente	Nr. Automático	
id_servico	Identificador do serviço	Nr. Automático	
valor_servico	Valor do serviço	Moeda	
valor_recebimento	Soma total dos valores de todas parcelas	Moeda	
num_parcela	Número da parcela	Nr. Natural	
valor_parcela	Valor da parcela	Moeda	
dt_pagamento	Data do recebimento da parcela	Data	
id_recebimento	Identificador do recebimento	Nr. Automático	
estado_recebimento	Status do recebimento	Nr. Natural	0 – realizado 1 – a realizar

UC11 Renegociar contratos

Item	Descrição	Tipo-base	Domínio
id_cliente	Identificador do cliente	Nr. Automático	
id_servico	Identificador do serviço	Nr. Automático	
valor_servico	Valor do serviço	Moeda	
valor_recebimento	Soma total dos valores de todas parcelas	Moeda	
num_parcela	Número da parcela	Nr. Natural	
valor_parcela	Valor da parcela	Moeda	
dt_pagamento	Data de pagamento da parcela	Data	
id_recebimento	Identificador do recebimento	Nr. Automático	

Item	Descrição	Tipo-base	Domínio
estado_recebimento	Status do recebimento	Nr. Natural	0 – realizado 1 – a realizar

UC12 Cadastrar Forma de Pagamento

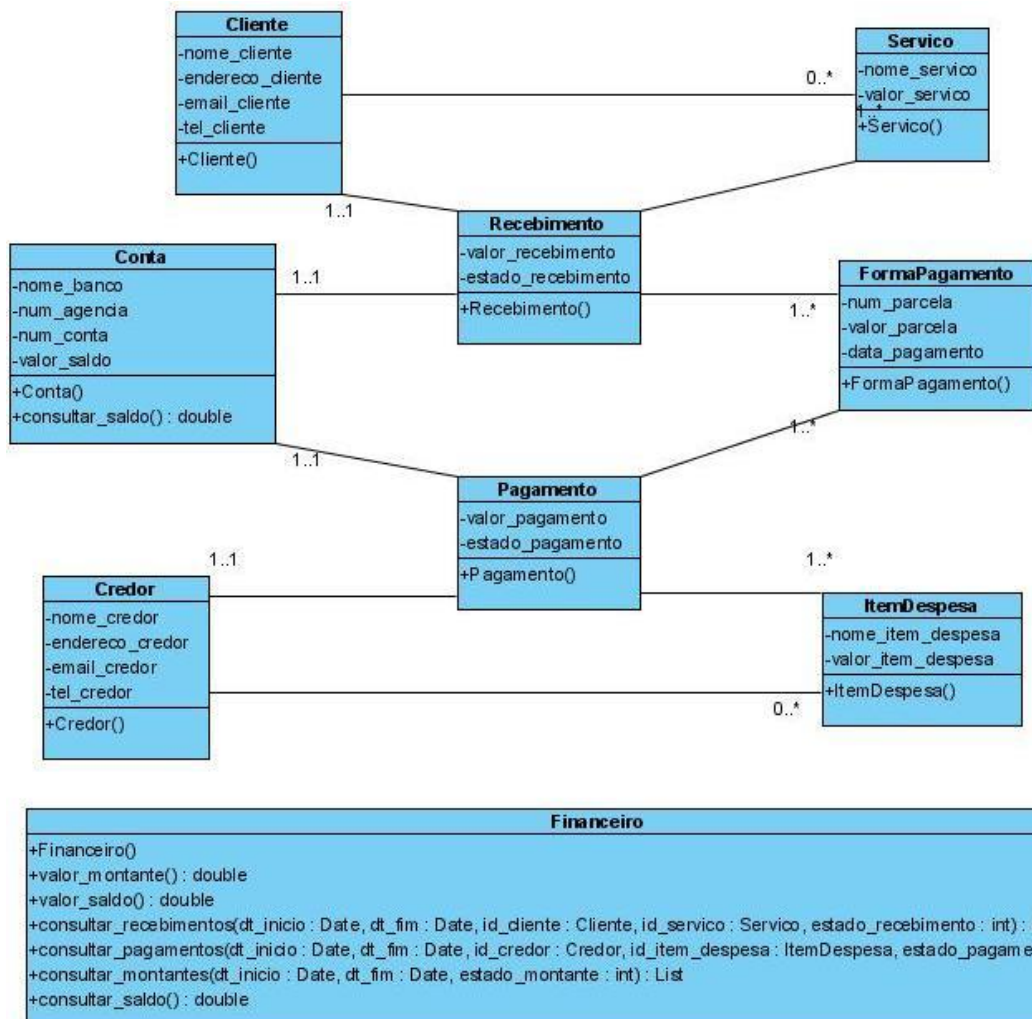
Item	Descrição	Tipo-base	Domínio
id_forma_pagamento	Identificador da forma de pagamento	Nr. Automático	
num_parcela	Número da parcela	Nr. Natural	
valor_parcela	Valor da parcela	Moeda	
dt_pagamento	Data de pagamento da parcela	Data	

UC13 Cadastrar Conta

Item	Descrição	Tipo-base	Domínio
id_conta	Identificador da conta	Nr. Automático	
nome_banco	Nome do banco	Texto	
num_agencia	Número da agência	Texto	
num_conta	Número da conta	Texto	
valor_saldo	Saldo da conta	Moeda	

Anexo D

Modelo de Classes de Domínio do Participante P3



Anexo E

Modelo de Casos de Uso do Participante P4

Este anexo apresenta, como exemplo, o modelo de casos de uso produzido pelo participante P4 (Grupo B - técnica tradicional), para o sistema FGV.

CASOS DE USO:

Nr.	Ator	Nome do UC
1	cliente	Contratar empresa
2	cliente	Pagar por serviço
3	cliente	Cancelar contrato
4	cliente	Renegociar Contrato
5	empresa	Alimentar sistema (inserir dados)
6	empresa	Pagar empresa credora
7	empresa	Verificar saldo
8	empresa	Verificar movimentação
9	empresa	Verificar despesas
10	empresa	Renegociacao de contrato
11	tempo	Renovacao de contrato(1 ano)
12	empresa	Listar recebimentos
13	empresa	Listar contato cliente
14	empresa	Listar pagamentos
15	empresa	Listar contato credor
16	empresa	Listar info. De pagamento
17	empresa	Listar montante a pagar
18	empresa	Listar montante a receber

DESCRIÇÃO DOS UC'S (*outline*):

Ator: **cliente** UC 1 : **contrata empresa**

1. cliente necessita de serviço da empresa
2. cliente procura empresa
3. empresa dispõe os serviços
4. cliente avalia as possibilidades
5. cliente efetiva contrato (de um só serviço ou pacote).
6. contrato e dados do cliente são registrados no sistema.

Ator: **cliente** UC 2: **pagar por serviço**

1. após contratar a empresa
2. cliente solicita serviço

3. empresa realiza serviço
4. cliente paga o valor contatado no dia combinado
5. pagamento é registrado no sistema.

Ator: **cliente** UC 3: **cancelar contrato**

1. cliente se motiva a deixar a empresa por qualquer motivo
2. cliente solicita o aviso prévio com dois meses de antecedência
3. passado o período, contrato é cancelado
4. transação é registrada no sistema.

Ator: **cliente** UC 4: **renegociação de contrato**

1. cliente deseja modificar termos do contrato
2. cliente contata a empresa
3. ambos renegociam os termos
4. renegociação é registrada no sistema.

Ator: **empresa** UC 5: **alimentar sistema**

1. a empresa necessita de fazer um cadastro de cliente, produto ou serviço
2. responsável (**representante da empresa**) faz **login** no sistema
3. encarregado do sistema alimenta o sistema com os dados gerados
4. Dados são armazenados no sistema
5. Sistema possivelmente os processa e emite um retorno ao usuário.

Ator: **empresa** UC 6: **pagar empresa credora**

1. empresa contrata algum serviço de uma outra empresa
2. empresa contratada executa o serviço
3. empresa contratante paga pelo serviço prestado.
4. responsável **faz o login** no sistema
5. responsável registra pagamento no sistema

Ator: **empresa** UC 7: **verificar saldo**

1. responsável (**representante da empresa**) pelo sistema **faz o login** no sistema
2. responsável faz a solicitação de saldo com os dados necessários
3. sistema processa os dados
4. saldo é exibido ao usuário da empresa.

Ator: **empresa** UC 8: **verificar movimentação**

1. responsável (**representante da empresa**) pelo sistema **faz o login** no sistema
2. responsável faz a solicitação de movimentações com os dados necessários

3. sistema processa os dados
4. movimentações são exibidas ao usuário da empresa.

Ator: **empresa** UC 9: **verifica despesas**

1. responsável (**representante da empresa**) pelo sistema **faz o login** no sistema
2. responsável faz a solicitação das despesas
3. sistema processa os dados e da as opções de listagem
4. usuário escolhe a **opção desejada**
5. sistema processa
6. sistema exibe as despesas

Ator: **empresa** UC 10: **renegociação de contrato**

1. empresa deseja modificar termos do contrato de um determinado cliente
2. empresa contata cliente
3. ambos renegociam os termos
4. renegociação é registrada no sistema.

Ator: **tempo** UC 11: **renovação do contrato**

1. o contrato de um determinado cliente está fazendo aniversário
2. sistema verifica se não há solicitação de cancelamento ou agendamento de renegociação
3. se não houver, sistema renova o contrato.

Ator: **empresa** UC 12: **Listar recebimentos**

1. responsável (**representante da empresa**) pelo sistema **faz o login** no sistema
2. responsável faz a solicitação dos recebimento
3. sistema processa os dados e da as opções de listagem
4. usuário escolhe a **opção desejada**
5. calcula juros e multa dos **recebimentos** em atraso
6. sistema processa e **marca os recebimentos** em atraso
7. sistema exibe os recebimentos pedidos

Ator: **empresa** UC 13: **listar contato cliente**

1. responsável (**representante da empresa**) pelo sistema **faz o login** no sistema
2. responsável faz a solicitação dos dados do cliente
3. sistema processa os dados e da as opções de listagem
4. usuário escolhe a **opção desejada**
5. sistema processa
6. sistema exibe os dados cadastrais pedidos.

Ator: **empresa** UC 14: **Listar pagamentos**

1. responsável (**representante da empresa**) pelo sistema **faz o login** no sistema
2. responsável faz a solicitação dos **pagamentos**
3. sistema processa os dados da as opções de listagem
4. usuário escolhe a **opção desejada**
5. calcula juros e multa dos **pagamentos** em atraso
6. sistema processa e **marca os pagamentos** em atraso para serem priorizados
7. sistema exibe os **pagamentos** pedidos

Ator: **empresa** UC 15: **Listar contato credor**

1. responsável (**representante da empresa**) pelo sistema **faz o login** no sistema
2. responsável faz a solicitação dos dados da empresa credora
3. sistema processa os dados e da as opções de listagem
4. usuário escolhe a **opção desejada**
5. sistema processa
6. sistema exibe os dados cadastrais da empresa credora pedidos.

Ator: **empresa** UC 16: **Listar info. De pagamento**

1. responsável (**representante da empresa**) pelo sistema **faz o login** no sistema
2. responsável faz a solicitação das informações dos pagamentos
3. sistema processa os dados e da as opções de listagem
4. usuário escolhe a **opção desejada**
5. sistema processa verificando se há algum pagamento em atraso
6. se houver pagamento em atraso é calculado os juros e a multa
7. sistema exibe as informações das informações solicitadas.

Ator: **empresa** UC 17: **Listar montante a pagar**

1. responsável (**representante da empresa**) pelo sistema **faz o login** no sistema
2. responsável faz a solicitação do montante a pagar
3. sistema processa os dados e da as opções de listagem
4. usuário escolhe a **opção desejada**
5. sistema processa
6. sistema exibe o montante a pagar, o saldo inicial e final dia-a dia do período.

Ator: **empresa** UC 18: **Listar montante a receber**

1. responsável (**representante da empresa**) pelo sistema **faz o login** no sistema
2. responsável faz a solicitação do montante a receber
3. sistema processa os dados e da as opções de listagem
4. usuário escolhe a **opção desejada**
5. sistema processa
6. sistema exibe o montante a receber, o saldo inicial e final dia-a dia do período.

GLOSSÁRIO

Termo	Descrição	Informação Adicional
Representante da empresa	Qualquer representante da empresa que tenha o direito de uso de sistema	
opção desejada	Opção desejada é uma maneira de generalizar as opções que o sistema dará ao usuário, como o período das movimentações poe exemplo.	Este termo foi agregado para simplificar os UCs
faz o login	Fazer o <i>login</i> é uma maneira de abstrair o tipo de entrada no sistema, esta pode ser via senha, cartão de acesso, digital, ou como for solicitado a equipe que codificará o sistema.	
marca os pagamentos/recebimento	Marcar os pagamento/recebimentos é uma maneira de alertar o usuário quanto a validade vencida das contas a receber e a pagar.	
pagamentos	Pagamentos são as contas a pagar da empresa e não deve ser confundida com os recebimentos que têm do cliente.	

Anexo F

Modelo de Classes de Domínio do Participante P4

