



**PODER EXECUTIVO
MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO AMAZONAS
INSTITUTO DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA**



**OPENSIMALS: UM REPOSITÓRIO ABERTO PARA AUXILIAR NO
ENSINO DE MODELAGEM DE SOFTWARE EMPREGANDO
ESTRATÉGIAS DE APRENDIZAGEM ATIVA**

WILLIAMSON ALISON FREITAS SILVA

Manaus
Fevereiro, 2020

WILLIAMSON ALISON FREITAS SILVA

**OPENSIMALS: UM REPOSITÓRIO ABERTO PARA AUXILIAR NO
ENSINO DE MODELAGEM DE SOFTWARE EMPREGANDO
ESTRATÉGIAS DE APRENDIZAGEM ATIVA**

Tese de doutorado submetida ao corpo docente
do Programa de Pós-Graduação em Informática
da Universidade Federal do Amazonas (PPGI-
UFAM).

Área de concentração: Engenharia de Software.

Orientadora: Prof^ª. Tayana Uchôa Conte, D.Sc.

Co-Orientador: Prof. Igor Fábio Steinmacher, D.Sc.

Manaus
Fevereiro, 2020

Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

S586o Silva, Williamson Alison Freitas
OPENSMALS: Um repositório aberto para auxiliar no ensino de modelagem de software empregando estratégias de aprendizagem ativa / Williamson Alison Freitas Silva . 2020
262 f.: il. color; 31 cm.

Orientador: Tayana Uchôa Conte
Coorientador: Igor Fábio Steinmacher
Tese (Doutorado em Informática) - Universidade Federal do Amazonas.

1. Ensino de Modelagem de Software. 2. Ensino de Modelagem de Software. 3. Estratégias de Aprendizagem Ativa. 4. Repositório Aberto. 5. OpenSMALS. I. Conte, Tayana Uchôa. II. Universidade Federal do Amazonas III. Título



PODER EXECUTIVO
MINISTÉRIO DA EDUCAÇÃO
INSTITUTO DE COMPUTAÇÃO

PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA



FOLHA DE APROVAÇÃO

**"OPENSIMALS: Um Repositório Aberto para auxiliar no
Ensino de Modelagem de Software empregando
Estratégias De Aprendizagem Ativa"**

WILLIAMSON ALISON FREITAS SILVA

Tese de Doutorado defendida e aprovada pela banca examinadora constituída pelos Professores:

Jose Francisco de Magalhaes Netto
Prof. José Francisco de Magalhães Netto - PRESIDENTE

Alberto Nogueira de Castro Junior
Prof. Alberto Nogueira de Castro Junior - MEMBRO INTERNO

Andreia Malucelli
Prof.^a Andreia Malucelli - MEMBRO EXTERNO

Gustavo Henrique Lima Pinto
Prof. Gustavo Henrique Lima Pinto - MEMBRO EXTERNO

Igor Fabio Steinmacher
Prof. Igor Fabio Steinmacher - MEMBRO EXTERNO

Manaus, 06 de Fevereiro de 2020

“Ensinar exige respeito aos saberes dos educandos.

Ensinar exige a convicção de que a mudança é possível.

Ensinar exige compreender que a educação é uma forma de intervenção no mundo.

Ensinar exige tomada consciente de decisões.

Ensinar exige saber escutar.

Ensinar exige disponibilidade para o diálogo.

Ensinar exige querer bem aos educandos.

Ensinar não é transferir conhecimento.

(Paulo Freire)

Aos meus dois grandes amores: meu marido, Gueviston Lima, e minha mãe, Sandra Maria.

AGRADECIMENTOS

Ao Gueviston Lima pelo amor, carinho, amizade, incentivo, cuidados e companheirismo em todos os momentos desta jornada. Te agradeço acima de tudo pela paciência e compreensão diante de tantas ausências ao longo de todos esses anos. Agradeço pelas horas e horas de conversas que tivemos e que você sempre me aconselhava, escutava e, quando necessário, brigava. Ao seu lado passei por muitos momentos bons e inúmeros momentos ruins (claro, porque a vida não é feita apenas de bons momentos) e em nenhum deles você soltou minha mão. Sem você, seu apoio e sua força não conseguiria estar aqui. Não tenho palavras para descrever o quanto você foi e é fundamental para o fechamento desse ciclo. TE AMO, MEU AMOR!! Agradeço, carinhosamente, a minha mãe Sandra Maria da Silva e ao meu pai Claudemir Nogueira Freitas, pelo apoio, preocupação e conselhos sempre tão seguros e úteis. Agradeço, minha mãe, pelas ligações nas madrugadas. Parece que a senhora sabia quando eu precisava escutar a sua voz ou apenas desabafar um pouco. Amo vocês.

Aos sinceros e verdadeiros amigos que fiz ao longo dessa caminhada, em ambientes acadêmicos ou não. Em especial aos meus amigos, Ana Oran, Bruna Ferreira, Walter Nakamura, Léo Marquez, Elizamary Nascimento, Randerson Queiroz, Anna Beatriz, Ursula Campos, Adriana Damian, Felipe Manzoni, Patrícia Matsubara, Thiago Oliveira, Daniel Ycaro, Maria Eduarda, Dhanielly Lima, Ana Paula Chavez Steinmacher, Giuseppe Iaffaldano e Andrea Stockl. Vocês foram essenciais para essa caminhada. E agradeço aos meus colegas do grupo de pesquisa USES, pela cooperação, compartilhamento de conhecimento e troca de experiências vividas. Aos professores-pesquisadores que abriram as portas para colaboração com a nossa pesquisa, em especial ao professor Bruno Gadelha, Marco Aurélio Gerosa, Guilherme Horta Travasso, Breno Nicolau, Jobson Massollar e Elaine Oliveira. É sempre um grande prazer ter vocês por perto.

Aos membros da banca de defesa, Prof. Alberto N. de Castro J, Profa. Andreia Malucelli, Prof. Gustavo Pinto, Prof. José F. Magalhães. Sou extremamente grato pela contribuição de vocês. É sempre um prazer receber o *feedback* de vocês.

À Universidade Federal do Amazonas, ao Instituto de Computação e ao PPGI-UFAM pelo apoio durante todo o doutorado. À Helen, ao Robson, ao Márcio e, em especial, ao meu querido amigo Frank Azevedo pela ajuda em todos os momentos e por ser sempre essa pessoa solidária e sempre disposto a ajudar. Ao professor Eduardo Feitosa por sempre estar disposto a

auxiliar nos processos pertinentes ao PPGI. À CAPES, pelo apoio financeiro concedido ao longo do doutorado.

Ao meu coorientador do doutorado Igor Steinmacher que aceitou me orientar nessa caminhada, por sempre me fazer pensar fora da caixa seja relacionada a pesquisa ou sobre minha vida, por ter me aceito no doutorado sanduíche na *Northern Arizona University* e me mostrar um mundo de possibilidades que eu nem imaginava que existia, por me proporcionar uma experiência agradável e inesquecível em Flagstaff, por me fazer quebrar meus medos e encarar tudo de frente. Só tenho uma coisa a dizer: Muito obrigado por tudo!!!

Finalmente, um agradecimento especial à minha querida orientadora, Tayana Conte. Tay, obrigado por você sempre estar por perto e me guiando nessa caminhada, por sempre me motivar e sempre me fazer ir além, por sempre abrir novas oportunidades para mim e para todos que estão a sua volta, por todo o aprendizado que você me ofereceu, por me fazer amar à prática da docência e a da pesquisa, por me ensinar que quando se quer algo bem feito é preciso ter dedicação e várias e várias horas de trabalho, por todas as horas de trabalhos que passamos juntos, pelas várias reuniões discutindo a pesquisa, pelos momentos de risos e de choro que sempre me faziam crescer. Muito obrigado pela sua dedicação a este trabalho em todos os momentos e, principalmente, pelo seu apoio constante durante esta jornada.

A todos que direta ou indiretamente contribuíram para a realização desta pesquisa.

RESUMO

Modelagem é um tópico central na Educação em Engenharia de Software. A modelagem de software possibilita que os engenheiros de software compreendam de forma aprofundada os conceitos ou problemas específicos durante o desenvolvimento de software usando diferentes níveis de detalhes. A *Unified Modeling Language* (UML) é um padrão de modelagem amplamente usado na indústria de software como uma linguagem para especificar sistemas de software. No entanto, os estudantes enfrentam dificuldades enquanto estão aprendendo modelagem de software empregando os diagramas UML. Uma causa possível é a abordagem tradicional centrada no professor que ainda domina os cursos de Computação. Para ensinar a modelagem de software de maneira eficaz, os professores que geralmente empregam métodos tradicionais podem usar estratégias de aprendizagem ativa. Contudo, os professores relutam em mudar suas abordagens de ensino devido às diversas barreiras que impedem a aplicação de estratégias ativas de aprendizagem. Além disso, há relativamente poucas pesquisas que tratam sobre como mitigar essas barreiras. O objetivo apresentado nesta tese de doutorado é apoiar os professores na adoção de estratégias de aprendizagem ativas no ensino de modelagem de software. Para atingir esse objetivo, esta pesquisa foi guiada pela abordagem de pesquisa *Design Science Research* (DSR). O DSR permitiu a delimitação do problema de pesquisa, o desenvolvimento, a avaliação e a evolução de um artefato. A abordagem DSR é um processo iterativo e compreende três ciclos de pesquisa interligados entre si: O Ciclo de Relevância, o Ciclo de *Design* e o Ciclo de Rigor. Durante o Ciclo de Relevância, realizou-se a análise da viabilidade do tema de pesquisa. Para isso, foi conduzido um Mapeamento Sistemático da Literatura para compreender as principais dificuldades percebidas pelos estudantes ao realizar a modelagem de software e identificar as estratégias de ensino utilizadas pelos professores para apoiar o ensino de modelagem de software. Foram também conduzidos estudos experimentais exploratórios para obter uma melhor compreensão, sob as perspectivas dos estudantes e professores, das estratégias de aprendizagem ativa identificadas na literatura. O Ciclo de Design possibilitou a o desenvolvimento, avaliação e evolução de um artefato. O artefato proposto é um repositório chamado OpenSMALS (*Open Repository for Software Modeling Teaching from Active Learning Strategies*). O OpenSMALS fornece diretrizes específicas sobre como os professores podem aplicar as estratégias de aprendizagem ativa, além de ajudá-los na identificação das estratégias que melhor se adequam ao seu contexto de ensino. Para avaliar e

evoluir o OpenSMALS, foram conduzidos quatro Ciclos de *Design* – em quatro universidades diferentes – para avaliar o uso do OpenSMALS. Os resultados mostraram que o OpenSMALS reduziu satisfatoriamente as barreiras enfrentadas pelos professores e alcançou um nível de maturidade adequado a ser adotado por outros professores. Por fim, o Ciclo de Rigor refere-se principalmente à geração e o uso do conhecimento. Com isso, os principais fundamentos estão relacionados ao conhecimento sobre educação em Engenharia de Software, as estratégias de aprendizagem ativa, o Mapeamento Sistemático da Literatura, os estudos experimentais, análise qualitativa e quantitativa, Grupos de Foco, entrevistas, dentre outros. Com relação à geração de conhecimento, a principal contribuição para a base de conhecimento é o próprio OpenSMALS, como um novo repositório aberto que ajuda os instrutores a identificar estratégias de aprendizagem ativa mais adequadas ao seu contexto de ensino. Além disso, (i) o processo de uso do OpenSMALS em configurações reais serve como exemplo para outros instrutores, (ii) a investigação realizada para desenvolver o OpenSMALS, (iii) o conjunto de estratégias de aprendizagem ativa que podem ser usadas para ensinar diagramas UML (iv) os aspectos a serem considerados na implementação da estratégia e (v) o conhecimento relacionado à análise dos impactos da estratégia no ensino do diagrama UML.

Palavras-chave: Ensino de Modelagem de Software, Educação em Engenharia de Software, Estratégias de Aprendizagem Ativa, Repositório Aberto, OpenSMALS.

ABSTRACT

Modeling is a core topic in Software Engineering Education. Software modeling enables an in-depth understanding of specific concepts or problems using different detail levels. Unified Modeling Language (UML) has been adopted as a standard modeling language in the software industry for the graphical representation of design models using diagrams. Nevertheless, students face difficulties while learning software modeling. One possible cause is the traditional teacher-centered approach that still dominates in Computer Science. To teach software modeling effectively in computing courses, instructors who usually employed traditional methods could use active learning strategies. However, instructors are reluctant to change their teaching approaches due to several barriers that hinder the application of active learning strategies. Besides, relatively little research addresses how to mitigate them. The objective presented in this doctoral thesis is to support instructors to implement active learning strategies in software modeling teaching. To achieve this objective, we conducted a Design Science Research (DSR). DSR allowed the delimitation of the research problem, the development, evaluation and evolution of an artifact. DSR approach is an iterative process and comprises three interlinked research cycles: the Relevance Cycle, the Design Cycle, and the Rigor Cycle. During the Relevance Cycle, we carried out the feasibility analysis of the research topic. For this, we conducted a Systematic Literature Mapping to understand the main difficulties perceived by students when performing software modeling. Besides, we identified the teaching strategies applied by instructors to support software modeling teaching. We also conducted exploratory experimental studies to obtain a better understanding, from the perspective of students and instructors, of the active learning strategies identified in the literature. The Design Cycle enabled the development, evaluation, and evolution of an artifact. We proposed an artifact called OpenSMALS (**O**pen Repository for **S**oftware **M**odeling Teaching using **A**ctive Learning Strategies). OpenSMALS provides specific guidelines on how instructors can apply these strategies and helping instructors to identify the active learning strategies best suit their teaching context. We performed four DSR design cycles — in four different universities — to evaluate the use of OpenSMALS. Our results show that OpenSMALS satisfactorily reduced the barriers faced by instructors, and it achieved an appropriate maturity level to be adopted by other instructors. Finally, the Rigor Cycle refers primarily to generating and using knowledge. The main foundations are knowledge related to Software Engineering Education, Active

Learning Strategies, Systematic Literature Mapping, experimental studies, qualitative and quantitative analysis, focus groups, interviews, among others. The main contribution to the knowledge base is OpenSMALS itself, as a new open repository that helps instructors identify AL strategies best suited to their teaching context. Additionally, (i) the process of using of OpenSMALS in real settings serves as an example for other instructors, (ii) the investigation performed to develop OpenSMALS, (iii) the set of active learning strategies that can be used to teach UML diagrams, (iv) the aspects to be considered when implementing the strategy, and (v) the knowledge related to the analysis of impacts of the strategy in teaching UML diagram.

Keywords: Teaching Software Modeling, Software Engineering Education, Active Learning Strategies, Open Repository for Software Modeling Teaching, OpenSMALS.

LISTA DE FIGURAS

Figura 1.1. Visão geral dos ciclos de <i>Design Science Research</i> neste trabalho, baseado em Hevner e Chatterjee (2010), similar ao processo adotado por Fernandes Matsubara (2019)...	24
Figura 2.1. Diagramas disponíveis pela UML (adaptada de OMG, 2015).....	31
Figura 3.1. Distribuição dos estudos por ano.	43
Figura 3.2. Distribuição dos estudos identificados por tipo de publicação.	44
Figura 3.3. Tipos de Dificuldades relacionadas à aos diagramas da UML.	46
Figura 3.4: Tipos de dificuldades relacionadas ao diagrama de casos de uso da UML.....	48
Figura 3.5. Tipos de dificuldades relacionadas ao diagrama de classes da UML.	50
Figura 3.6. Tipos de dificuldades relacionadas ao diagrama de sequência da UML.	53
Figura 3.7. Mapeamento dos resultados obtidos a partir da combinação das subquestões de pesquisa SQ2, SQ3 e SQ5.	58
Figura 3.8. Mapeamento dos resultados obtidos a partir da combinação das subquestões de pesquisa SQ3, SQ4 e SQ6.	59
Figura 3.9. Mapeamento dos resultados obtidos a partir da combinação das subquestões de pesquisa SQ6, SQ7 e SQ8.	60
Figura 4.1. Procedimento seguido no estudo.....	73
Figura 4.2. Sessão do Grupo Focal.....	74
Figura 4.3. Boxplots com os resultados da corretude.....	77
Figura 4.4. Boxplots com os resultados da completude.	80
Figura 4.5. Procedimento seguido no estudo.....	89
Figura 4.6. Boxplot para a corretude dos diagramas.	91
Figura 4.7. Boxplot para a completude dos diagramas.	92
Figura 4.8. Resultados das Percepções dos estudantes sobre os métodos.....	93
Figura 4.9. Percepção dos estudantes sobre os métodos PBL e ErrEx.	96
Figura 4.10. Nível de engajamentos dos estudantes em relação aos métodos.	98
Figura 4.11. Exemplo das cartas Passo da Modelagem e Perguntas-Guia.	106
Figura 4.12. Exemplo das cartas Dicas e Exemplo.	107
Figura 4.13. Pontuação e Regras utilizadas no jogo.....	107
Figura 4.14. Gráfico de Avaliação da categoria Motivação.	109
Figura 4.15. Gráfico de Avaliação da categoria Experiência do Usuário.	111
Figura 4.16. Gráfico de Avaliação da categoria Aprendizagem.	112
Figura 4.17. Gráfico das medidas de autoavaliação.	113
Figura 5.1. Visão geral do método de pesquisa.....	117
Figura 5.2. Quadro criado para orientar a discussão do Grupo Focal do tipo <i>Lovers X Haters</i>	120
Figura 5.3. Sessão do Grupo Focal.....	120
Figura 5.4. Exemplo de codificação.	122
Figura 5.5. Percepções dos estudantes sobre as estratégias (1/2).....	124
Figura 5.6. Percepções dos estudantes sobre as estratégias (2/2).....	125
Figura 6.1. Modelo conceitual elaborado com as informações coletadas das estratégias.	142
Figura 6.2. Exemplo de grupos de perguntas da primeira versão da árvore de decisão para o questionário.	144
Figura 6.3. Estratégia sugerida pelo questionário.	145
Figura 6.4. Portal web com informações sobre as estratégias (1/2).	146
Figura 6.5. Portal web com informações sobre as estratégias (2/2).	147

Figura 6.6. Visão geral do método de pesquisa.....	149
Figura 6.7. Extrato de perguntas da árvore de decisão do questionário - versão final.....	152
Figura 6.8. Exemplo de percepção dos estudantes sobre a estratégia (<i>Think-Pair-Square</i>) na segunda versão do OpenSMALS.....	153
Figura 6.9. Uma nova versão do portal OpenSMALS (v3).....	156
Figura 6.10. Páginas da Web com artefatos que os professores podem empregar em seus cursos.....	157
Figura 6.11. Alterações feitas para adicionar tempo e esforço para preparar as estratégias.	160

LISTA DE TABELAS

Tabela 2.1. Barreiras percebidas pelos professores ao adotar Estratégias de Ativas de Aprendizagem, incluindo a quantidade de grupos que mencionaram cada barreira	36
Tabela 3.1: Sub-Questões de Pesquisa.	42
Tabela 3.2. Locais de publicação com mais de duas publicações.	44
Tabela 3.3. Resultados do SM para cada uma das SubQuestões.....	45
Tabela 3.4. Resultados por tipo de diagrama.....	56
Tabela 3.5. Métricas identificadas para avaliar os diagramas da UML.	56
Tabela 3.6. Mapeamento entre dificuldades enfrentadas pelos estudantes e as tecnologias propostas na literatura.....	63
Tabela 4.1. Fórmula utilizada para as métricas.	75
Tabela 4.2. Valores da corretude dos diagramas e da especificação.	76
Tabela 4.3. Valores da completude dos diagramas e da especificação.	79
Tabela 4.4. Distribuição dos estudantes por grupo.	88
Tabela 4.5. Resumo do Resultado da Avaliação dos Diagramas por Grupo.....	90
Tabela 4.6. Preferência dos Métodos para Aprender a modelar.....	92
Tabela 4.7. Itens avaliados nos questionários pós-modelagem.	95
Tabela 4.8. Descrição das Cartas utilizadas no jogo.	105
Tabela 5.1. Informações sobre as turmas.....	117
Tabela 5.2. Itens avaliados em questionários pós-modelo.	119
Tabela 5.3. Fatores e subfatores identificados no estudo qualitativo.	126
Tabela 5.4. Primeiro fator: Pontos de vista durante a modelagem.	127
Tabela 5.5. Segundo fator: Aprendizagem colaborativa.	127
Tabela 5.6. Terceiro fator: Apoio durante a modelagem.....	128
Tabela 5.7. Quarto fator: A estratégia permite que o estudante receba <i>feedback</i>	129
Tabela 5.8. Quinto fator: abstração e resolução de problemas.	129
Tabela 5.9. Primeiro Desafio: Os estudantes precisam ter conhecimento prévio sobre o conteúdo a ser ensinado.	130
Tabela 5.10. Segundo Desafio: Os professores precisam ter cuidado ao definir o nível de dificuldade dos cenários utilizados.....	131
Tabela 5.11. Terceiro Desafio: Os professores precisam definir adequadamente o tempo para aplicar as estratégias.	131
Tabela 5.12. Quarto Desafio: Os professores precisam tomar cuidado no desenvolvimento dos materiais necessários para implementar as estratégias.	132
Tabela 5.13. Benefícios das estratégias percebidos pelos professores ao empregar estratégias ativas de aprendizagem.....	132
Tabela 6.1. Estratégias selecionadas para compor o OpenSMALS.....	140
Tabela 6.2. Categorias das estratégias de aprendizagem ativa, baseado em Barkley <i>et al.</i> (2014).	142
Tabela 6.3. Informações sobre os professores e a turma.	148
Tabela 6.4. <i>Script</i> semiestruturado da entrevista.	150
Tabela 6.5. Percepção do professor I01 sobre o OpenSMALS (v1).	151
Tabela 6.6. Benefícios percebidos pelos professores usando o OpenSMALS (v2).	154
Tabela 6.7. Dificuldades percebidas pelos professores usando o OpenSMALS (v2).	154
Tabela 6.8. Sugestões de melhoria recomendadas pelos professores do OpenSMALS (v2).	155

Tabela 6.9. Sugestões identificadas no OpenSMALS (v2) e que devem ser aprimoradas em uma nova versão.	155
Tabela 6.10. Benefícios e dificuldades percebidos pelos professores que usaram (v3).	158
Tabela 6.11. Sugestões de melhoria recomendadas pelos professores do OpenSMALS (v3).	159

SUMÁRIO

CAPÍTULO 1 – INTRODUÇÃO	20
1.1 CONTEXTO	20
1.2 QUESTÃO DE PESQUISA	22
1.3 OBJETIVOS DA PESQUISA	22
1.4 METODOLOGIA DE PESQUISA	23
1.5 ORGANIZAÇÃO.....	27
CAPÍTULO 2 – FUNDAMENTAÇÃO TEÓRICA E TRABALHOS RELACIONADOS	29
2.1 ENSINO DE ENGENHARIA DE SOFTWARE.....	29
2.2 MODELAGEM DE SOFTWARE.....	30
2.3 ESTRATÉGIAS DE APRENDIZAGEM ATIVA	33
2.3.1 Barreiras relacionadas à adoção de estratégias de aprendizagem ativa	35
2.4 TRABALHOS RELACIONADOS	38
2.5 CONSIDERAÇÕES SOBRE O CAPÍTULO	40
CAPÍTULO 3 – CICLO DE RELEVÂNCIA (PARTE I) – MAPEAMENTO SISTEMÁTICO SOBRE AS DIFICULDADES ENFRENTADAS PELOS ESTUDANTES DURANTE A MODELAGEM DE SOFTWARE UTILIZANDO OS DIAGRAMAS DA UML	41
3.1 INTRODUÇÃO.....	41
3.2 RESULTADOS DO MSL	42
3.2.1 Tendências das publicações	42
3.2.2 Visão Geral dos Resultados	44
3.2.3 Dificuldades relacionadas à UML e seus diagramas.....	46
3.2.4 Resultados das tecnologias identificadas no MSL	54
3.2.5 Métricas para avaliar diagramas UML.....	55
3.2.6 Mapeamento dos resultados quantitativos	57
3.3 DISCUSSÃO DOS RESULTADOS DO MSL.....	60
3.4 LIMITAÇÕES DO MSL	66
3.5 CONSIDERAÇÕES SOBRE O CAPÍTULO	67
CAPÍTULO 4 – CICLO DE RELEVÂNCIA (PARTE II) - ESTUDOS EXPERIMENTAIS EXPLORATÓRIOS SOBRE O ENSINO DE MODELAGEM DE SOFTWARE COM DIAGRAMAS DA UML	69
4.1 INTRODUÇÃO.....	69
4.2 ESTUDO EXPERIMENTAL COMPARATIVO SOBRE A MODELAGEM REALIZADA INDIVIDUALMENTE E EM GRUPO	70
4.2.1 Planejamento e Execução do Estudo.....	71

4.2.2 Resultados do Estudo	76
4.2.3 Discussão dos Resultados	84
4.2.4 Considerações sobre o estudo	85
4.3 ESTUDO EXPERIMENTAL COMPARATIVO SOBRE <i>LEARNING FROM ERRONEOUS EXAMPLES</i> E <i>PROBLEM-BASED LEARNING</i>	86
4.3.1 Planejamento e Execução do Estudo.....	87
4.3.2 Resultados do Estudo Experimental	90
4.3.3 Discussão dos resultados do estudo	102
4.3.4 Considerações sobre o estudo	103
4.4 ESTUDO EXPERIMENTAL DE UM JOGO EDUCACIONAL PARA APOIAR O ENSINO DE DIAGRAMA DE ATIVIDADES	104
4.4.1 Jogo Educacional <i>ActGame</i>	105
4.4.2 Avaliação do Jogo <i>ActGame</i>	108
4.4.3 Considerações sobre o estudo	113
4.5 SÍNTESE DO CAPÍTULO	114
CAPÍTULO 5 – CICLO DE RELEVÂNCIA (PARTE III) - ESTUDO DE VIABILIDADE SOBRE O USO DAS ESTRATÉGIAS DE APRENDIZAGEM ATIVA A PARTIR DAS PERSPECTIVAS DOS ESTUDANTES E PROFESSORES	115
5.1 INTRODUÇÃO.....	115
5.2 MÉTODO DE PESQUISA	116
5.2.1 Participantes	117
5.2.2 Coleção dos dados.....	118
5.2.3 Análise de dados	121
5.3 RESULTADOS	122
5.3.1 QP1: Como os estudantes de graduação percebem as estratégias de aprendizado ativas empregadas durante o ensino de diagramas UML?.....	122
5.3.2 QP2: Quais desafios os professores percebem ao usar estratégias de aprendizagem ativas enquanto ensinam diagramas da UML?	129
5.4 DISCUSSÃO.....	133
5.5 CONSIDERAÇÕES SOBRE O CAPÍTULO	137
5.6 SÍNTESE DO CAPÍTULO	137
CAPÍTULO 6 – CICLO DE DESIGN – OPEN REPOSITORY FOR TEACHING SOFTWARE MODELING FROM ACTIVE LEARNING STRATEGIES (OPENSMAALS).....	139
6.1 INTRODUÇÃO.....	139
6.2 PRIMEIRO CICLO DE <i>DESIGN</i> : DESENVOLVIMENTO DO OPENSMAALS (v1)....	140
6.3 PLANEJAMENTO E EXECUÇÃO DOS ESTUDOS EXPERIMENTAIS CONDUZIDOS PARA AVALIAR AS VERSÕES DO OPENSMAALS	147
6.3.1 Planejamento	148
6.3.2 Execução	149
6.3.3 Análise dos dados.....	150

6.3.4 Resultados do estudo piloto	151
6.4 SEGUNDO CICLO DE <i>DESIGN</i> : DESENVOLVIMENTO DO OPENSMAIS (v2)....	151
6.4.1 Resultados do primeiro estudo experimental	154
6.5 TERCEIRO CICLO DE <i>DESIGN</i> : DESENVOLVIMENTO DO OPENSMAIS (v3) ...	155
6.5.1 Resultados do segundo estudo experimental	158
6.6 QUARTO CICLO DE <i>DESIGN</i> : DESENVOLVIMENTO DO OPENSMAIS (v4)	159
6.7 CONSIDERAÇÕES SOBRE O ESTUDO	160
CAPÍTULO 7 – CICLO DE RIGOR DA PESQUISA E CONSIDERAÇÕES FINAIS	162
7.1 CONSIDERAÇÕES	162
7.2 RIGOR DA PESQUISA	163
7.2.1 Contribuições da pesquisa.....	164
7.2.2 Publicações.....	165
7.3 IMPLICAÇÕES DA PESQUISA	167
7.4 LIMITAÇÕES DA PESQUISA	168
7.5 PERSPECTIVAS FUTURAS	170

CAPÍTULO 1 – INTRODUÇÃO

Este capítulo apresenta a motivação para realização desta tese de doutorado, a definição do problema, os objetivos propostos e a metodologia de pesquisa empregada para condução da pesquisa.

1.1 CONTEXTO

A abstração se tornou uma das habilidades mais importantes (Hazzan e Kramer, 2007; Sien, 2011; Zehetmeier *et al.*, 2019), pois possibilita uma profunda compreensão de conceitos ou problemas específicos usando diferentes níveis de detalhes (Hazzan e Kramer, 2007; Sien, 2011). A partir do pensamento abstrato, estudantes e engenheiros de software podem criar uma representação ou um modelo abstrato do mundo real. Esse mapeamento é chamado de modelagem de software (Hazzan e Kramer, 2007). A modelagem de software é um conceito-chave na educação em Engenharia de Software (do inglês, *Software Engineering Education - SEE*) (Störrle, 2017; Agner *et al.*, 2019).

Unified Modeling Language (UML) (Rumbaugh *et al.*, 2004) tem sido adotada como uma linguagem de modelagem padrão na indústria de software para a representação gráfica dos modelos de projeto (Störrle, 2017), empregando os diagramas. Apesar desse *status*, professores e pesquisadores relataram que estudantes enfrentam dificuldades ao aprender a modelagem de software, devido à complexidade de seus conceitos quando comparados a outros aspectos da Engenharia de Software (ES) (Ma, 2017; Sien 2011). Além disso, estudantes relatam diversas dificuldades ao aprender a modelagem de software com os diagramas UML, tais como: dificuldade em compreender a sintaxe e da semântica do diagrama (Ma, 2017); dificuldade em organizar as informações nos diagramas (Bolloju e Leung, 2006); dificuldade em utilizar corretamente relacionamentos do tipo generalização/especialização nos diagramas (Ma, 2017; Bolloju e Leung, 2006). Essas dificuldades, se não forem sanadas ou minimizadas durante o ensino de modelagem, podem afetar a qualidade final do software. Desta forma, os diagramas criados pelos estudantes representarão de forma incompleta e incorretamente o software que está sendo desenvolvido (Ma, 2017; Szmurło e Śmiałek, 2006).

As dificuldades enfrentadas pelos estudantes podem estar relacionadas com a maneira como a modelagem de software, empregando os diagramas UML, vem sendo ensinada (Szmurło e Śmiałek, 2006; Al-Tahat, 2014). A maioria dos professores conduzem ainda um

ambiente de aprendizagem centrado no professor para apresentar os conceitos de modelagem para os estudantes (Al-Tahat, 2014; Lin, 2019). Além disso, a natureza complexa dos cursos de ES impede que os professores encontrem facilmente uma estratégia pedagógica que atenda às necessidades dos estudantes, resultando em uma experiência de aprendizado ineficiente (Capuano *et al.*, 2012). De acordo com Connolly *et al.* (2007), os estudantes geralmente consideram difícil trabalhar em problemas que não têm uma solução bem definida ou que são ambíguos e vagos. Para aprender a resolver problemas de modelagem, os estudantes precisam obter mais experiência prática, que poderiam advir da participação em projetos de software reais, simulações, dramatizações, estudos de caso ou outras atividades de aprendizagem experiencial (Kurkovsky *et al.*, 2019). Dessa forma, algumas estratégias pedagógicas utilizadas cotidianamente não estão sendo mais eficazes para treinar a criatividade, porque os estudantes aprendem passivamente e não tem chance de aprender-fazendo (*learning by doing*) (Kurkovsky *et al.*, 2019; Lin, 2019).

Portanto, os professores precisam adaptar ou empregar novas estratégias pedagógicas para proporcionar aos estudantes um ambiente desafiador que os envolva ativamente no processo de aprendizagem (de Freitas *et al.*, 2016). Uma abordagem promissora nessa direção é a Aprendizagem Ativa (do inglês, *Active Learning*) (Prince, 2004; Brigggs, 2005). Segundo Bonwell e Eison (1991), *Active Learning* é tipicamente definida como uma aprendizagem que exige que os estudantes se envolvem de maneira cognitiva e significativa com o material didático. Então, os estudantes estão “*envolvidos com as informações apresentadas, realmente pensando sobre elas (analisando, sintetizando, avaliando), em vez de apenas recebê-las passivamente*” (King, 1993). O objetivo da aprendizagem ativa é oferecer oportunidades para os estudantes pensarem criticamente sobre o conteúdo por meio de uma série de atividades que ajudam a prepará-los para os desafios das situações profissionais (Fioravanti *et al.*, 2018). Aprendizagem ativa envolve os estudantes em um ambiente que estimula a colaboração, aprimora o desempenho acadêmico e as atitudes dos estudantes (Fioravanti *et al.*, 2018). Além disso, verificou-se que essas estratégias estão fortemente vinculadas ao ensino de modelagem, pois permitem que os estudantes elaborem várias soluções para um determinado problema (Schilling e Sebern, 2012; Knobloch *et al.*, 2018); (b) melhorem as habilidades de trabalho em equipe (Knobloch *et al.*, 2018; Nunes *et al.*, 2015); (c) melhorem as habilidades de comunicação oral e escrita usadas para modelar os diagramas (Knobloch *et al.*, 2018; Nunes *et al.*, 2015).

Há diversos relatos na literatura evidenciando que, nos cursos de graduação em Ciências, Tecnologia, Engenharia e Matemática, as estratégias de aprendizagem ativa são mais

eficazes do que aulas completamente passivas (Freeman *et al.*, 2014; Tharayil *et al.*, 2018; Michael, 2007). Apesar dessas evidências, há vários relatos evidenciando que não houve um aumento nas taxas de adoção dessas estratégias por parte dos professores (Fraser *et al.*, 2014; Tharayil *et al.*, 2018; Michael, 2007). Isso pode ser explicado pelo fato de os professores percebem uma série de barreiras específicas que dificultam a adoção das estratégias de aprendizagem ativa: (a) a quantidade de tempo gasto na preparação de uma aula empregando estratégias de aprendizagem ativa é muito longo (Tharayil *et al.*, 2018); (b) os professores acham difícil cobrir o plano de ensino do semestre ao adotar essas estratégias (Shadle *et al.*, 2017); (c) as possíveis respostas negativas dos estudantes ao introduzir novas estratégias de ensino nas aulas (Chi e Wylie, 2014; Michael, 2007); e (d) a literatura frequentemente não fornece detalhes suficientes sobre como empregar as estratégias de aprendizagem ativa (Borrego *et al.*, 2013). Em relação a este último ponto, com tantas demandas conflitantes, poucos professores têm tempo para pesquisar na literatura educacional sobre as estratégias. Portanto, minimizar as barreiras adotando estratégias de aprendizagem ativa em suas salas de aula é o primeiro passo para ajudar a mudar a maneira como os professores ensinam modelagem de software.

1.2 QUESTÃO DE PESQUISA

O problema tratado nesta tese de doutorado está relacionado com a melhoria do processo de ensino de modelagem de software empregando os diagramas da UML. Neste contexto, a questão de pesquisa geral que norteou esta pesquisa de doutorado é:

Como apoiar os professores na adoção de estratégias de aprendizagem ativa para o ensino de modelagem de software?

1.3 OBJETIVOS DA PESQUISA

O objetivo geral desta tese de doutorado é apoiar os professores durante a adoção das estratégias de aprendizagem ativa enquanto ensinam modelagem de software utilizando diagramas da UML. Para atingir esse objetivo principal, são definidos os seguintes objetivos específicos:

- Identificar as dificuldades enfrentadas pelos estudantes enquanto aprendem a modelagem de software com os diagramas da UML.

- Identificar as estratégias de ensino relatadas na literatura, que são empregadas pelos professores, para apoiar o ensino de modelagem com os diagramas da UML.
- Apresentar evidências, sob a perspectiva dos estudantes e dos professores, da percepção do uso de estratégias de aprendizagem ativa utilizadas para o ensino/aprendizagem dos diagramas da UML.
- Propor e avaliar um artefato que forneça um apoio sistemático para os professores empregarem as estratégias de aprendizagem ativa, de acordo com o seu contexto de ensino.

1.4 METODOLOGIA DE PESQUISA

Para alcançar o objetivo geral, esta pesquisa foi guiada por meio da abordagem *Design Science Research* (DSR) (Hevner e Chatterjee, 2010; Wieringa, 2009). A Figura 1.1 mostra o Mapa de DSR utilizado nesta tese de doutorado. Combinado com DSR foram utilizados vários métodos de pesquisa para demonstrar as etapas realizadas nesta pesquisa durante a execução dos ciclos de DSR. Entre os quais destacam-se o mapeamento sistemático da literatura, os estudos experimentais que incluem a análise quantitativa da corretude e completude dos diagramas UML elaborados pelos estudantes e a análise qualitativa das sessões de entrevistas de Grupo Focal (*Focus Group*) realizadas com os estudantes, e a análise qualitativa de entrevistas realizadas com os estudantes e professores. Na Figura 1.1, as estrelas indicam o capítulo em que os conteúdos são detalhados nesta tese.

DSR estabelece um processo sistemático que apoia na criação e avaliação de novos artefatos à medida que são desenvolvidos e usados para resolver problemas práticos de interesse geral (Hevner e Chatterjee, 2010). Além disso, o artefato original pode fornecer novas oportunidades para melhorar a prática muito antes que os profissionais reconheçam qualquer problema (Hevner, 2007). A abordagem DSR é um processo iterativo, que combina os paradigmas comportamentais e de *Design Science*, e compreende três ciclos de pesquisa interligados entre si (Hevner e Chatterjee (2010): O Ciclo de Relevância, o Ciclo de *Design* e o Ciclo de Rigor. A Figura 1.1 resume as informações principais relacionadas aos ciclos de DSR nesta pesquisa.

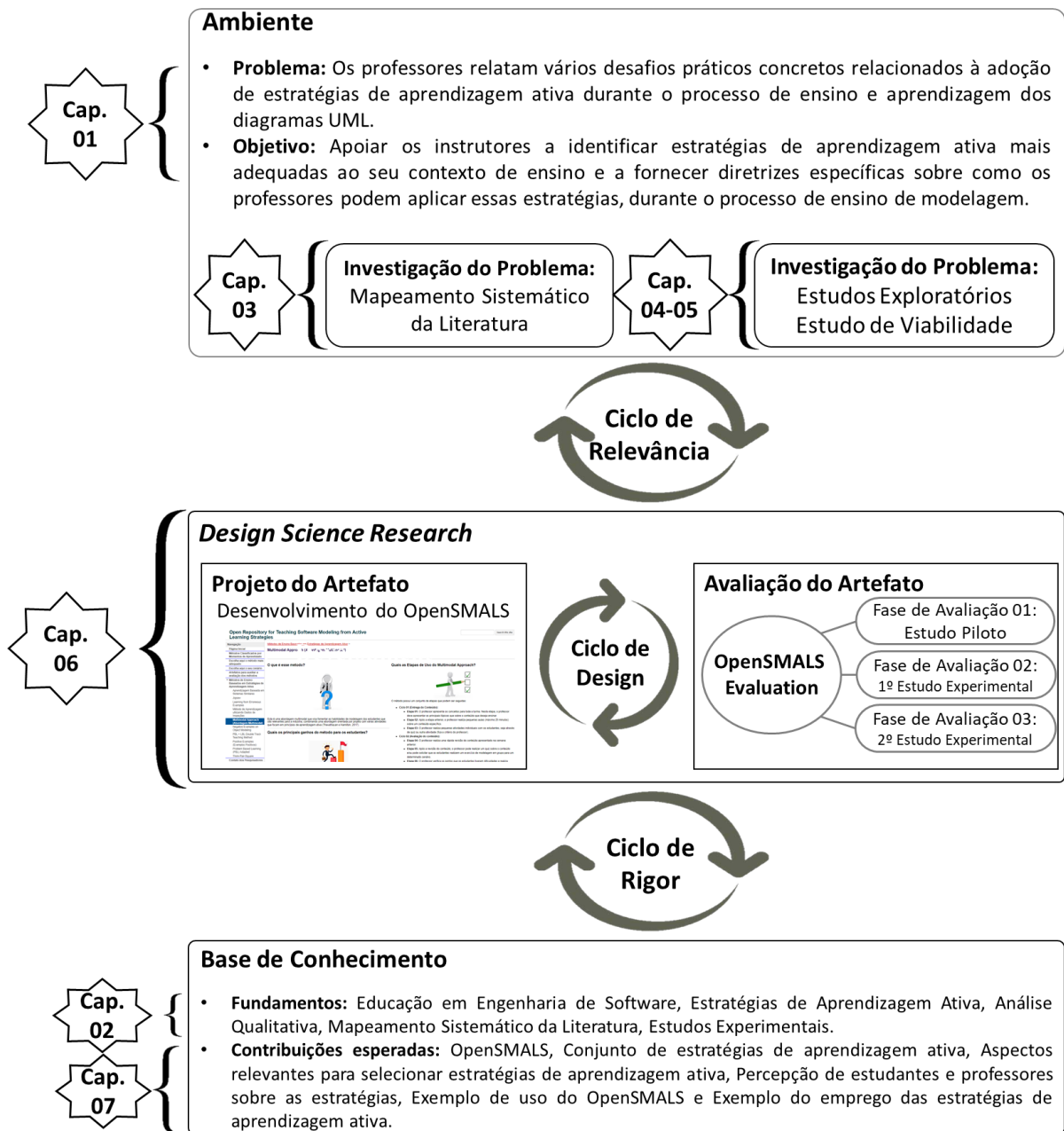


Figura 1.1. Visão geral dos ciclos de *Design Science Research* neste trabalho, baseado em Hevner e Chatterjee (2010), similar ao processo adotado por Fernandes Matsubara (2019).

O **Ciclo de Relevância** envolve a definição da motivação/problema a ser tratado, os requisitos de pesquisa (por exemplo, a oportunidade/problema a ser resolvido) e os critérios de aceitação para a avaliação final dos resultados da pesquisa. A saída do DSR deve ser retornada ao ambiente para estudo e para a avaliação no domínio da aplicação (Hevner e Chatterjee, 2010). A principal motivação para a condução desta pesquisa está relacionada à quantidade de evidências sobre a resistência da equipe acadêmica em adotar as estratégias de aprendizagem ativa (Pundak e Rozner, 2008; Zellweger, 2005; Eickholt, 2018). Para entender melhor sobre as barreiras enfrentadas pelos professores, foi conduzido um estudo secundário, estudos

exploratórios e um estudo de viabilidade da pesquisa. O estudo secundário, primeira investigação, conduzido por meio de um Mapeamento Sistemático da Literatura (MSL), teve dois objetivos: entender as dificuldades percebidas pelos estudantes ao realizar modelagem de software e identificar as estratégias de ensino utilizadas pelos professores para apoiar o ensino de modelagem de software. Os estudos experimentais exploratórios visaram compreender a influência do processo de aprendizagem de modelagem de software (segunda investigação) nas instituições de ensino, com foco no diagrama de casos de uso, especificação de casos de uso, diagrama de atividades, diagrama de classes e diagrama de sequência. O estudo de viabilidade teve como objetivo verificar se a temática de pesquisa é viável, ou seja, verificar se a adoção de estratégias de aprendizagem ativa é um processo viável e aceitável de ser empregado em diferentes contextos de ensino e em diversas instituições de ensino. Para isso, analisou-se e a percepção dos estudantes e dos professores sobre o uso das estratégias de aprendizagem ativa durante o ensino de modelagem de software (terceira investigação). Os dois estudos (exploratório e viabilidade) contemplaram diferentes instituições de ensino do Brasil

Considerando os resultados identificados nos estudos exploratórios e no MSL foram definidos dois requisitos que guiaram o desenvolvimento e a avaliação do artefato proposto:

(Req1) O artefato deve fornecer procedimentos sobre como os professores podem aplicar estratégias de aprendizagem ativa em sala de aula, além de prover ferramentas para apoiar o uso dessas estratégias.

(Req2) O artefato deve possibilitar que os professores identifiquem as estratégias de aprendizagem ativa mais adequadas ao seu contexto, durante o ensino de modelagem de software.

No **Ciclo de Design**, o objetivo é desenvolver e avaliar uma solução concreta que atenda ao problema e atenda aos requisitos definidos (Hevner e Chatterjee, 2010). O desenvolvimento dos artefatos pode ser orientado pelos requisitos, enquanto os resultados da aplicação desse artefato no contexto do problema podem corroborar ou questionar a validade desses requisitos. No presente trabalho, o artefato proposto é um repositório que visa auxiliar os professores a empregar as estratégias de aprendizagem ativa em seu contexto de ensino, chamado *OpenSMALS (Open Repository for Software Modeling Teaching from Active Learning Strategies)*. O repositório se concentra na tentativa de reduzir os desafios práticos enfrentados pelos professores, fornecendo diretrizes específicas sobre como os professores podem aplicar as estratégias, cenários de modelagem fornecidos por outros professores, questionários de avaliação e outros. O portal OpenSMALS evoluiu com base em evidências coletadas de um

estudo piloto e dois estudos empíricos, cada um deles resultando em uma versão do OpenSMALS. No primeiro ciclo de *Design*, um estudo piloto foi realizado como um caso ilustrativo para avaliar a viabilidade de aplicação da solução proposta por um professor e se a proposta poderia resolver o problema prático identificado (Hevner e Chatterjee, 2010). Depois disso, foram conduzidos dois estudos experimentais—segundo e terceiro Ciclos de *Design*. Os estudos foram conduzidos em quatro universidades diferentes. Por fim, no quarto Ciclo de *Design*, não houve a necessidade de realizar um novo estudo experimental, apenas evoluiu-se o OpenSMALS com base nas sugestões de melhoria identificadas no terceiro Ciclo de *Design*.

Finalmente, o **Ciclo Rigor** refere-se principalmente à geração e uso de conhecimento (Hevner e Chatterjee, 2010). O Ciclo Rigor é alcançado com base em teorias e métodos de pesquisa, juntamente com a experiência e o conhecimento dos fundamentos que norteiam a pesquisa, além de contribuir para o crescimento da base de conhecimento (Hevner, 2007). Neste trabalho, os principais fundamentos estão relacionados ao conhecimento sobre educação em ES, estratégias de aprendizagem ativa e um mapeamento sistemático da literatura. A principal contribuição para a base de conhecimento é o próprio OpenSMALS, como um repositório aberto que apoia os professores na identificação de estratégias de aprendizagem ativas mais adequadas ao seu contexto de ensino.

Além disso, são contribuições:

- (i) O processo de uso do OpenSMALS em ambientes reais—que pode servir de exemplo para outros professores;
- (ii) O uso de *Design Science Research* e Engenharia de Software Experimental para o desenvolvimento, avaliação e evolução de um artefato (OpenSMALS);
- (iii) O conjunto de estratégias de aprendizagem ativas que podem ser usadas para ensinar os diagramas da UML (disponíveis no OpenSMALS);
- (iv) Os aspectos a serem considerados pelos professores ao empregar as estratégias de aprendizagem ativa no ensino de modelagem de software com os diagramas UML; e
- (v) O conhecimento relacionado à análise dos impactos da estratégia no ensino de diagramas da UML.

Vale ressaltar que para atender aos requisitos éticos da pesquisa, o objetivo da pesquisa foi explicado a cada participante e o participante era informado de que podia optar por não

participar da pesquisa a qualquer momento. Caso aceitasse participar, o participante assinava um termo de consentimento, aprovado pelo Comitê de Ética (nº 2.545.694), garantindo a confidencialidade dos dados fornecidos e o anonimato do participante (ver Material de Apoio em Silva *et al.*, 2020).

1.5 ORGANIZAÇÃO

Este capítulo introdutório apresentou a contextualização, os objetivos da pesquisa e a metodologia de pesquisa a que se refere esta tese. O conteúdo será detalhado ao longo dos próximos capítulos. Este trabalho está organizado segundo a estrutura descrita abaixo:

Capítulo 2 – Fundamentação Teórica: este capítulo apresenta os conceitos teóricos que serão necessários para que o leitor tenha uma melhor experiência ao realizar a leitura desta tese, tais como: modelagem de software, ensino de modelagem de software, estratégias de aprendizagem ativa, entre outros. Além disso, será também apresentadas outras pesquisas realizadas e que estão relacionadas à esta tese.

Capítulo 3 – Ciclo de Relevância (Parte I) – Mapeamento sistemático sobre as dificuldades enfrentadas pelos estudantes durante a modelagem de software utilizando os diagramas da UML: Este capítulo apresenta os resultados obtidos de um estudo secundário (Mapeamento Sistemático da Literatura) cujo objetivo consistiu em identificar as principais dificuldades ao aprender os diagramas da UML.

Capítulo 4 – Ciclo de Relevância (Parte II) - Estudos experimentais exploratórios sobre o uso de estratégias para apoiar o ensino de modelagem de software com diagramas da UML: Este capítulo apresenta o planejamento, execução e resultados de estudos experimentais realizados com o objetivo de investigar a influência do processo de aprendizagem de modelagem de software (diagrama de casos de uso, especificação de casos de uso, diagrama de atividades, diagrama de classes e diagrama de sequência).

Capítulo 5 – Ciclo de Relevância (Parte III) – Estudo de viabilidade sobre o uso das estratégias de aprendizagem ativa a partir da perspectiva dos estudantes e professores: O objetivo deste capítulo é apresentar um estudo de viabilidade realizado com o objetivo de verificar se a adoção de estratégias de aprendizagem ativa no contexto de diversas instituições de ensino é viável. Para isso, analisou-se as percepções dos estudantes e professores em relação às estratégias de aprendizagem ativa utilizadas para apoiar o ensino e aprendizagem de modelagem de software, empregando os diagramas da UML.

Capítulo 6 – Ciclo de *Design - Open Repository for Software Modeling Teaching from Active Learning Strategies* (OPENSMALS): Este capítulo apresenta o processo de criação, avaliação e evolução do repositório aberto que visa apoiar os professores na utilização de estratégias de aprendizagem ativa, denominado *Open Repository for Software Modeling Teaching from Active Learning Strategies* (OpenSMALS).

Capítulo 8 – Ciclo de Rigor: Este capítulo apresenta conclusões desta pesquisa, apresentando as suas principais contribuições. Além disso, são apresentadas as perspectivas futuras, fornecendo possíveis direções de continuidade desta pesquisa.

CAPÍTULO 2 – FUNDAMENTAÇÃO TEÓRICA E TRABALHOS RELACIONADOS

Este capítulo apresenta os conceitos teóricos sobre os diagramas da UML e também as pesquisas relacionadas ao ensino e aprendizado dos diagramas da UML empregando métodos de aprendizagem ativa.

2.1 ENSINO DE ENGENHARIA DE SOFTWARE

Segundo Sommerville (2011), a Engenharia de Software (ES) é uma disciplina cujo foco está em todos os aspectos do processo de desenvolvimento do software, desde etapas de especificação até a etapa de manutenção. Dada a sua importância, cada vez mais, os profissionais da área de Computação devem estar capacitados (Jorge *et al.*, 2015). Logo, de acordo com Mendes *et al.* (2019), é de suma importância o sucesso no processo de aprendizagem dos conteúdos da área de ES, resultando em uma boa formação profissional dos estudantes.

O ensino de ES geralmente ocorre em cursos de graduação e de pós-graduação em Ciência da Computação, Sistemas de Informação, Engenharia de Computação e Bacharelado em Engenharia de Software, além de ter sido expandido para cursos técnicos, tecnológicos e treinamentos corporativos (Ferreira *et al.*, 2018). De acordo com as recomendações curriculares para a ES (ACM; IEEE, 2014), após finalizarem o curso de graduação, os futuros engenheiros de software devem ser capazes de:

- 1) Demonstrar o domínio dos conhecimentos e habilidades de engenharia de software e dos padrões profissionais necessários para iniciar a prática como engenheiro de software.
- 2) Demonstrar entendimento para aplicar teorias, modelos e técnicas apropriadas que forneçam uma base para identificação e análise de problemas, design de software, desenvolvimento, implementação, verificação e documentação.
- 3) Trabalhar individualmente e como parte de uma equipe para desenvolver e fornecer artefatos de software de qualidade.
- 4) Demonstrar entendimento e apreciação da importância da negociação, hábitos de trabalho eficazes, liderança e boa comunicação com os *stakeholders* em um ambiente típico de desenvolvimento de software.

- 5) Projetar soluções apropriadas em um ou mais domínios de aplicação usando abordagens de engenharia de software que integram preocupações éticas, sociais, legais e econômicas.
- 6) Conciliar os objetivos conflitantes do projeto, alcançando equilíbrio aceitável em relação a limitações de custo, tempo, conhecimento, sistemas existentes e questões organizacionais.
- 7) Aprender novos modelos, técnicas e tecnologias conforme elas surjam, e apreciar a necessidade por um desenvolvimento profissional contínuo.

Conforme visto acima, as competências esperadas que os futuros engenheiros de software possuam são muitas e, algumas vezes, muito difíceis de serem alcançadas. Durante processo de aprendizagem de Engenharia de Software, de acordo com Gary (2015), além de aprender os conceitos teóricos e adquirir as habilidades técnicas necessárias, os estudantes precisam desenvolver e aprimorar a capacidade de aplicar, evoluir e praticar essas habilidades ao longo de sua vida profissional. No entanto, o desenvolvimento das habilidades, das competências e das capacidades transversais geralmente possui menos apoio nos cursos de Computação (Marques *et al.*, 2014).

Alguns estudos relatam que vários dos pontos do currículo de ES estão sendo ensinados e treinados de forma superficial com os estudantes (Tomer e Mishra, 2015; Quezada-Sarmiento *et al.*, 2018). Isso pode estar ocorrendo devido ao fato de o ensino de Engenharia de Software ainda ser focado em uma abordagem tradicional (Andrade *et al.*, 2017). Isto significa que a preocupação dos professores, ao aplicar este modelo de ensino, é apenas em transmitir os conhecimentos aos estudantes, focando em aspectos das diversas teorias que norteiam a ES, sem apresentar a contextualização prática em que esses conceitos estão inseridos, ou seja, tornando os estudantes agentes passivos da aprendizagem (Capuano *et al.*, 2012; Lin, 2019).

2.2 MODELAGEM DE SOFTWARE

De acordo com a *Object Modeling Group* (OMG), “a modelagem é o design de aplicações de software antes da codificação.” A modelagem é uma atividade chave no desenvolvimento de software e que tem potencial transformador se for mais ampla e melhor utilizada (Agner e Lethbridge, 2017). Durante o projeto e o desenvolvimento de software baseado em modelos, a modelagem de software é usada como parte essencial do processo de desenvolvimento (da Silva, 2015). Os modelos são construídos e analisados antes da implementação do sistema, além de serem utilizados para direcionar a implementação

subsequente. Um melhor entendimento de um sistema pode ser obtido considerando-o sob diferentes perspectivas (também conhecidas como múltiplas visões), como modelos estáticos e modelos dinâmicos do sistema de software. Uma linguagem de modelagem gráfica, como a UML, apoia no desenvolvimento, entendimento e comunicação das diferentes visões (Storrle, 2011).

A UML (OMG, 2011) é a notação mais conhecida e utilizada (Agner *et al.*, 2019; Störrle, 2017), apresentando diversos tipos de diagramas para representar visões diferentes do modelo de domínio do problema relacionado ao software. Em sua versão atual (UML 2.5), a UML oferece 14 diagramas que descrevem a estrutura ou o comportamento de um sistema (Figura 2.1).

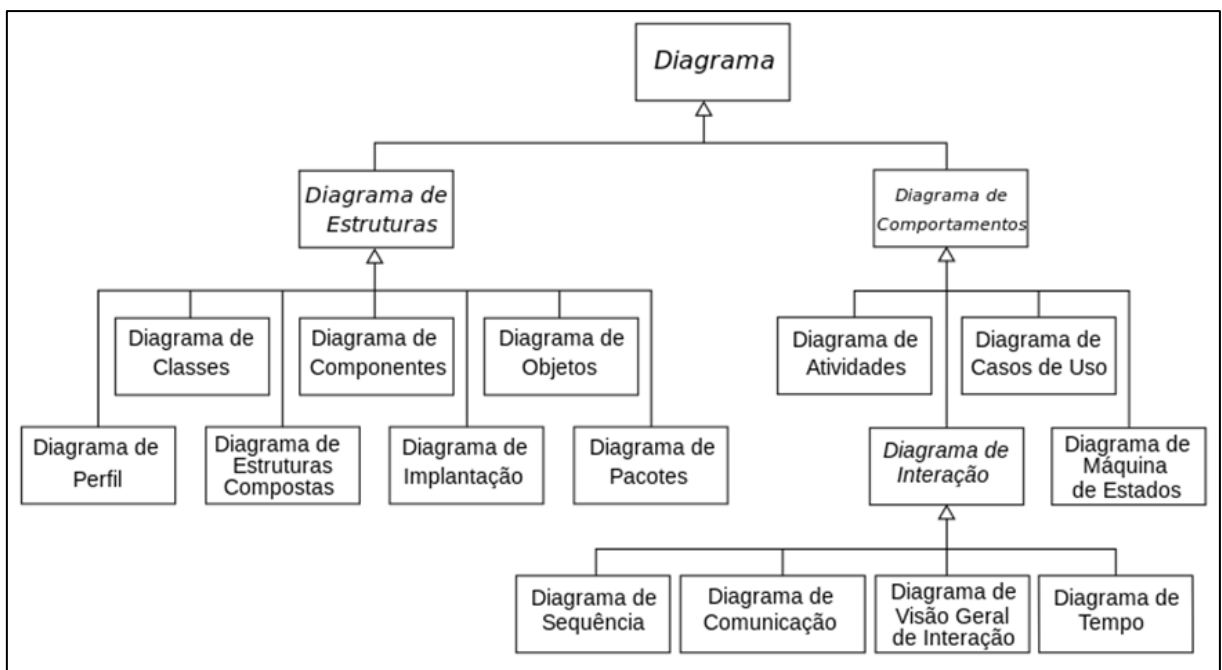


Figura 2.1. Diagramas disponíveis pela UML (adaptada de OMG, 2015).

Os diagramas da UML representam duas visualizações diferentes de um modelo de sistema (Guedes, 2008; OMG, 2015): a visão estática que define a estrutura do sistema e a visão dinâmica (ou comportamental) que elabora o comportamento dinâmico do sistema, mostrando colaborações entre objetos e mudanças nos estados internos de objetos. Mesmo assim, a UML é uma linguagem de modelagem de propósito geral, no entanto, seu uso não é apenas restrito ao software de modelagem, mas é comumente usada para engenharia de sistemas, modelagem de processos de negócios e para representar as estruturas organizacionais (Guedes, 2008). Os diagramas estruturais abrangem os diagramas de classes, de estruturas compostas, de objetos, de componentes, de implantação, de perfil e de pacotes. Os diagramas comportamentais

englobam o diagrama de casos de uso, atividades, máquina de estados, sequência, comunicação, visão geral de interação e de tempo, sendo que estes últimos quatro correspondem aos diagramas da subdivisão de diagramas de interação.

Apesar do uso de modelagem de software com os diagramas da UML ser considerada uma boa prática (Badreddin *et al.*, 2013), os professores possuem dificuldades em identificar quais as estratégias pedagógicas mais adequadas para treinar os estudantes a pensar abstratamente. (Silva *et al.*, 2019). Empregar estratégias inadequadas para ensinar os estudantes modelagem de software pode resultar na criação de modelos que não são aplicados da maneira correta ou que não são utilizados (Ma, 2017; Sien 2011). Além disso, o resultado de um treinamento ineficiente dos estudantes não resultaria em benefícios tangíveis e consequentemente em uma baixa aceitação dos diagramas por parte dos estudantes, conforme relatado por Damian *et al.* (2019). Por esse motivo, os estudantes rejeitariam a modelagem de software e pensariam que é uma tarefa adicional ou mesmo inútil no desenvolvimento de software (Petre, 2014).

Esse cenário motivou diversas pesquisas, fóruns e *workshops* voltados para investigar o ensino-aprendizagem de modelagem de software. Um exemplo é o *Educators Symposium*, realizado em conjunto com a conferência MODELS. Tal simpósio oferece aos educadores, pesquisadores e profissionais um fórum de pesquisa para discutir estratégias para melhor abordar questões como o uso apropriado de modelos e tecnologias relacionadas na sala de aula. Esta iniciativa mostra a importância da educação em modelagem de software.

Com relação ao ensino de modelagem de software, pôde-se observar na literatura que alguns trabalhos têm explorado estratégias para melhorar o ensino de modelagem de software. Há um consenso de que o ensino— tradicional e voltado para metodologias de desenvolvimento—deve ser transformado para refletir os resultados de aprendizagem que se espera que os estudantes alcancem. Garcia e Moreno (2004) relataram que as aulas práticas, no curso de Engenharia de Software, foram organizadas em sessões de *workshop*, onde cada sessão é dedicada à modelagem de software usando um dos diagramas da UML. Tamai (2005) exemplifica em seu trabalho como os formalismos da teoria dos grafos podem ser usados para ensinar modelagem (adotando a UML como exemplo) em cursos de graduação. Kuzniarz e Staron (2006) propuseram as melhores práticas para o ensino de modelagem/UML e enfatizam particularmente o papel da consistência entre diferentes diagramas ou visões. Kuzniarz e Börstler (2011) fornecem uma visão geral das diferentes maneiras de introduzir modelagem e como os estudantes podem adquirir a capacidade de criar os diagramas. Yohannis (2016) utiliza

elementos de gamificação em um jogo educacional para apoiar a aprendizagem de modelagem de software, especialmente com diagramas UML.

Além disso, os pesquisadores relatam que o ensino de modelagem de software, utilizando a UML, resulta em uma experiência muito positiva para professores e estudantes. Segundo Garcia e Moreno (2004), os benefícios podem ser resumidos em dois grupos, benefícios de ensino e de habilidades sociais. Do ponto de vista do ensino, quando o professor ensina a modelagem de software, eles introduzem a resolução de problemas, o aprendizado cooperativo e o debate sobre seus principais aspectos (Garcia e Moreno, 2004). O ensino da modelagem de software usando UML permite que os professores aprimorem as habilidades “sociais” dos estudantes como profissionais de software. Do ponto de vista das habilidades sociais, o ensino da UML permite que os estudantes trabalhem em equipe, a fim de encontrar uma solução inicial e praticar suas habilidades verbais, apresentando e defendendo a solução proposta (Garcia e Moreno, 2004).

Assim, pode-se observar que a comunidade de pesquisadores de educação em Engenharia de Software dedicou um grande esforço ao desenvolvimento de novas estratégias pedagógicas para tornar o ensino mais atraente para os estudantes, principalmente utilizando como base teórica os princípios de aprendizagem ativa (Agner e Lethbridge, 2017).

2.3 ESTRATÉGIAS DE APRENDIZAGEM ATIVA

Conforme já mencionado na Subseção 2.2, o método tradicional de ensino, ministrando aulas expositivas, aplicando provas e trabalhos complementares, continua sendo amplamente utilizado por professores (Marques *et al.*, 2014). No entanto, de acordo com Sancho-Thomas *et al.* (2009), os métodos educacionais que são centrados no professor podem não auxiliar no desenvolvimento prático de competências, além de poder ter uma eficiência de aprendizado limitada (Prikladnicki *et al.*, 2009).

Para mitigar esses problemas, pesquisadores sugerem o uso de estratégias de aprendizagem ativas com o objetivo de proporcionar aos estudantes novas experiências e oportunidades de aprendizado, melhorando o aprendizado geral dos estudantes (Aglan e Ali, 1996). A aprendizagem ativa é fundada no construtivismo social: os estudantes constroem o conhecimento através da interação com ideias, conceitos, materiais e outros artefatos (Junhua, 2013; Knight, 2004). Ensinar usando estratégias de aprendizagem ativa exige uma mudança na forma como os estudantes adquirem o conhecimento e a compreensão conceitual, em oposição

a aprendizagem passiva. Junhua (2013) e Briggs (2005) argumentam que a visão construtivista desenvolvida nas estratégias de aprendizagem ativa visa criar experiências de ensino que: (i) ajudem os estudantes a construir uma compreensão mais profunda dos conceitos teóricos em conexão com as experiências práticas; (ii) facilitem o desenvolvimento das habilidades dos estudantes; e (iii) desenvolvam as capacidades e disposições dos estudantes para se envolver em pesquisas colaborativas baseadas em projetos, melhorando assim o pensamento crítico.

Os estudantes assumem um papel central e assumem a responsabilidade por seu próprio aprendizado. Ao utilizar essas estratégias durante as aulas, os estudantes devem descobrir parte do conhecimento necessário para realizar o trabalho (Exposito, 2010). O professor, atuando como mediador, facilita o aprendizado e trabalha como um guia nesse processo. Além disso, ao aprenderem com essas estratégias, os estudantes estarão trabalhando no desenvolvimento de habilidades como responsabilidade, trabalho em equipe e a capacidade de sintetizar conceitos aprendidos (Prince, 2013). Vários estudos mostram que as estratégias ativas de aprendizagem costumam ser mais eficazes do que as aulas tradicionais para promover uma ampla gama de resultados educacionais desejáveis, incluindo maior aprendizado dos estudantes (Freeman *et al.*, 2014; Michael, 2007; Marques *et al.*, 2014). A literatura evidencia as vantagens de usar a aprendizagem ativa no currículo de cursos de graduação, como Engenharia Mecânica (Aglan e Ali, 1996; Talley *et al.*, 2007; Welch e Klosky, 2007), Design de Sistemas de Engenharia (Dennis *et al.*, 2001; Linsey *et al.*, 2010), Matemática (Keringan, 2018; Lai e Hwang, 2016) e Física (Karamustafaoglu, 2009; Base, 2006).

Ao adotar essas estratégias em sala de aula, espera-se que os estudantes aprendam o conhecimento técnico e os aspectos não técnicos necessários na indústria de software (Badreddin *et al.*, 2018). A aplicação dessas estratégias para o ensino de modelagem de software beneficia particularmente a educação dos estudantes. Os estudantes desenvolvem habilidades cada vez mais procuradas no mundo dos negócios, que os métodos tradicionais não desenvolvem. Além disso, essas estratégias aumentam a motivação e a satisfação dos estudantes e promovem efetivamente o pensamento crítico e as habilidades analíticas (Michael, 2006; Knobloch *et al.*, 2018).

Vale ressaltar que nesta tese de doutorado, os pesquisadores envolvidos estão atuando apenas como usuários em educação, ou seja, o objetivo é apenas compreender as estratégias de aprendizagem ativas existentes e agrupá-las de forma a apoiar o professor a selecionar a mais adequada para o seu contexto de ensino, além de fornecer os materiais pedagógicos necessários para aplicá-las em sala de aula. Por este motivo, não serão propostas novas estratégias para

melhorar o processo de aprendizagem dos estudantes e não se entrou em detalhes sobre as teorias de aprendizagem.

2.3.1 Barreiras relacionadas à adoção de estratégias de aprendizagem ativa

Há diversas evidências experimentais na literatura de que as estratégias de aprendizagem ativa oferecem vantagens de aprendizado aos estudantes, incluindo “*aumentar a motivação dos estudantes, retenção de conhecimento e melhoria na transferência de conteúdo ensinado em sala de aula*” (Cattaneo, 2017). Por exemplo, Freeman *et al.* (2014) realizaram uma meta-análise das estratégias de aprendizagem ativa empregadas em cursos de Ciências, Tecnologia, Engenharia e Matemática (STEM - *Science, Technology, Engineering, and Mathematics*). Os resultados mostraram que, em comparação com os cursos tradicionais, a taxa de reprovação reduziu de 33,8% para 21,8% quando os professores utilizaram estratégias de aprendizagem ativa nos cursos (Freeman *et al.*, 2014). Para reformar as práticas de ensino adotadas atualmente e melhorar o sucesso dos estudantes, faz-se necessário que os professores considerem as estratégias ativas de aprendizagem como um meio potencial para atingir esses objetivos (Michael, 2006). As universidades também devem fomentar seus investimentos em treinamento para professores sobre o uso de estratégias ativas de aprendizagem e espaços de aprendizagem projetados para apoiar essas estratégias e aumentar a taxa de sucesso dos estudantes (Brooks, 2017). No entanto, como em qualquer nova inovação, existem barreiras em relação à adoção de estratégias de aprendizagem ativa em salas de aula e desafios que surgem quando os professores tentam implementar uma estratégia ativa em suas salas de aula, limitando o efeito do uso dessas estratégias (Michael, 2007). Para melhor compreender as barreiras e desafios dos professores durante a adoção de aprendizagem ativa, será apresentada evidências de trabalhos que mostram as principais barreiras ao adotar novas estratégias de ensino por professores.

Michael (2007) realizou uma análise das barreiras à adoção de estratégias ativas de aprendizagem. Para isso, o autor coletou o *feedback* de um grupo de 29 professores de uma universidade sobre suas percepções e os motivos pelos quais eles e outros membros do corpo docente hesitaram em ensinar empregando estratégias de aprendizagem ativa. Os professores foram divididos em quatro grupos, em que cada grupo desenvolveu uma lista de barreiras comuns à adoção das estratégias. Em seguida, houve um agrupamento das barreiras e foi gerada uma lista única. A lista completa possui 22 barreiras listadas pelos professores está apresentada na Tabela 2.1. As duas principais barreiras mais citadas pelos professores se concentram em (1)

a quantidade de tempo que os professores levam para desenvolver um novo currículo e (2) na falta de espaço adequado na sala de aula que apoie estratégias de aprendizagem ativa.

Tabela 2.1. Barreiras percebidas pelos professores ao adotar Estratégias de Ativas de Aprendizagem, incluindo a quantidade de grupos que mencionaram cada barreira

Barreiras sobre o uso de Estratégias de Aprendizagem Ativa
4 Grupos mencionaram as seguintes barreiras: (1) As estratégias de aprendizagem ativa requerem muito tempo de preparação (2) A sala de aula em que ensinamos não está adequada para o uso de aprendizagem ativa
3 Grupos mencionaram as seguintes barreiras: (3) Os estudantes não sabem como aprender ativamente (4) As estratégias de aprendizagem ativa necessitam de muito tempo para serem aplicadas nas aulas e não cobre o conteúdo como um todo. (5) Ao utilizar estratégias ativas em uma sala de aula o professor tem menos controle da turma (6) A aprendizagem ativa é comprometida porque os estudantes não vêm para a aula preparados (7) Os estudantes não estão dispostos a se envolver em aprendizado ativo (8) A aprendizagem ativa é difícil de realizar devido à heterogeneidade do estudante (9) Os estudantes não têm a maturidade necessária para a aprendizagem ativa
2 Grupos mencionaram as seguintes barreiras: (10) A avaliação do estudante é difícil em aulas que utilizam aprendizagem ativa (11) As percepções dos colegas inibem a aprendizagem ativa (12) Tamanho da turma é um impedimento ao uso de estratégias de aprendizado ativo (13) A cultura de aprendizagem de professores e estudantes é uma barreira (14) Expectativas dos estudantes sobre a aprendizagem
1 Grupo mencionou as seguintes barreiras: (15) É difícil prever os resultados da aprendizagem em uma sala de aula ativa (16) A aprendizagem ativa corre o risco de avaliações ou classificações ruins dos estudantes (17) É difícil garantir “controle de qualidade” em um curso com várias seções (18) Não há recursos de aprendizado suficientes disponíveis (19) A falta de maturidade dos professores (pessoal e profissional) é uma barreira (20) A estrutura de recompensa do corpo docente torna a aprendizagem ativa pouco atraente (21) Os períodos de aula padrão são uma barreira (22) Os professores não sabem como fazê-lo

Além do trabalho de Michael (2007), outros pesquisadores procuraram catalogar as barreiras que impedem os professores de adotar novos métodos, estratégias e tecnologias de ensino. Ocaik (2011) conduziu entrevistas que foram conduzidas com 117 membros do corpo docente em quatro universidades da Turquia. A partir das respostas das entrevistas, os autores realizaram uma análise qualitativa. A partir dos resultados, o pesquisador identificou oito barreiras, que incluíam barreiras relacionadas a processos instrucionais, preocupações com a comunidade e preocupações com questões técnicas. A barreira mais citada pelos participantes foi a relacionada ao “grau de complexidade da estratégia”. Esta barreira foi evidenciada em 24,93% das respostas. A barreira “preocupações com o apoio institucional” foi evidenciada em 17,26% das respostas. Essas duas barreiras refletem as mencionadas pelos grupos de professores de Michael (2007): “As estratégias de aprendizagem ativa requerem muito tempo de preparação” e “a estrutura de recompensa da faculdade torna a aprendizagem ativa pouco atraente”. Vale a pena notar que o autor escolheu atribuir porcentagens ao número de vezes que

cada barreira foi mencionada, não ao número de participantes que mencionaram uma barreira específica. Contudo, não está claro se barreiras específicas foram observadas repetidamente por um pequeno número de professores ou talvez menos frequentemente por um grande número de participantes.

Al-Busaidi e Al-Shihi (2012) conduziram um estudo que explorou a satisfação percebida pelos professores sobre a adoção das estratégias empregadas em um ambiente de aprendizagem misto, ou seja, ambiente em que são adotadas estratégias de ensino tradicionais e “inovadoras”. Ao todo, um total de 82 participantes responderam a um questionário, que consistia em 39 indicadores medidos em uma escala do tipo Likert. A análise quantitativa dos pesquisadores incluiu classificações das respostas dos professores a cada indicador e correlações entre cada um dos grupos de indicadores, medindo a validade de suas hipóteses declaradas. A partir dos resultados, os pesquisadores perceberam que os indicadores “formação de professores”, “qualidade da estratégia” e o “apoio ao uso da estratégia” influenciaram positivamente na satisfação dos professores. Além disso, os autores evidenciaram que a satisfação do professor com as estratégias de aprendizagem ativa se correlacionava positivamente com as intenções de uso contínuo da estratégia em cursos que combinava o uso de estratégias tradicionais com as estratégias inovadoras, diminuindo as barreiras e aumentando a taxa de adoção dos professores.

Eickholt (2016) realizou um estudo com o objetivo de investigar as principais barreiras percebidas pelos professores em relação à adoção das estratégias existentes em cursos de Computação. Para isso, o autor conduziu um *survey* (pesquisa de opinião). O *survey* foi enviado a 369 professores dos cursos de Ciências da Computação na região centro oeste dos Estados Unidos. A análise das respostas revelou que a barreira do tempo disponível permaneceu como a barreira mais comumente relatada pelos professores, com 42,8% dos entrevistados discordando da afirmação de que têm o tempo necessário para mudar seu estilo de ensino. O número de estudantes e o tamanho da turma foram as segundas e terceiras respostas mais comumente fornecidas pelos professores. O autor também relatou que aqueles professores que já tinham tido uma exposição prévia a cursos de pedagogia não possuem dificuldades em empregar estratégias ativas de aprendizagem, assim fazem uso das mesmas em sala de aula como forma de melhorar os resultados dos estudantes. Foi relatado também que muitos professores possuem dificuldades em obter conhecimentos sobre o uso das estratégias e quais os recursos necessários para empregá-las em suas aulas.

Para resumir, a literatura está repleta de estudos que mostram a importância da aprendizagem ativa nos cursos de STEM, principalmente em cursos de Computação. A aprendizagem ativa aumenta o desempenho e a experiência dos estudantes em sala de aula. No entanto, conforme visto acima, as taxas de adoção da aprendizagem ativa pelos professores são indiscutivelmente baixas e caem em algum lugar na faixa de 20 a 40% (Froyd *et al.*, 2013; Grissom *et al.*, 2017). O tempo, o custo e a falta de informações sobre as estratégias são relatados como as principais barreiras à adoção da aprendizagem ativa (Michael, 2007; Park e Choi, 2014; Turpen *et al.*, 2016). Dada a prevalência e a longevidade dessas à adoção, não foram encontradas evidências de que existem atividades, artefatos, tecnologias que poderiam ser utilizadas como suporte pelo corpo docente das instituições de ensino e que os levaria ao uso da aprendizagem ativa (Eickholt, 2016; Grissom *et al.*, 2017). Nesse caso, talvez seja possível encontrar uma solução para a barreira do tempo.

2.4 TRABALHOS RELACIONADOS

Os professores podem usar estratégias de aprendizagem ativa para envolver os estudantes nas atividades de sala de aula, tornando-os participantes ativos do processo de aprendizado. A seguir, será relatado alguns trabalhos em que os autores usaram essas estratégias para ensinar modelagem de software (diagramas UML) a estudantes de graduação.

Pavlov *et al.* (2005) desenvolveram um método de treinamento prático baseado em pantomima (representação de uma história exclusivamente através de gestos, expressões faciais e movimentos) para auxiliar os estudantes durante a elaboração dos diagramas da UML. Os autores realizaram um experimento controlado e os resultados mostraram que o método é uma maneira eficaz para simular o trabalho em equipe em projetos de software complexos. O método ainda permitiu que os estudantes pudessem verificar a expressividade dos diagramas da UML durante a modelagem do software, além de entender o quão importante estes diagramas são para realizar a comunicação dentro de uma equipe.

Bolloju *et al.* (2009) examinaram a eficácia dos padrões e anti-padrões (exemplos errados) de diagramas de classes para ensinar modelagem de software. Os autores realizaram um experimento controlado com estudantes de graduação para analisar a utilidade de exemplos negativos e positivos no ensino de habilidades de modelagem orientada a objetos. Os resultados do experimento indicaram que exemplos positivos e negativos são úteis, mas depende do tipo de tarefa que se quer realizar. Com isso, os autores perceberam que os exemplos positivos (padrões) são recomendados quando se deseja que os estudantes compreendam a modelar a

qualidade sintática dos modelos, enquanto exemplos negativos (anti-padrões) são apropriados para a compreensão semântica dos modelos.

Ishii *et al.* (2010) realizaram um estudo de caso utilizando o método *Problem-Based Learning* (PBL) para integrar a aprendizagem das habilidades de modelagem dos diagramas da UML e as habilidades sociais necessárias em cenários de desenvolvimento de software. Os resultados do estudo confirmaram que, através do método PBL, os estudantes mostraram melhorias em relação as habilidades de modelagem e as habilidades sociais. Esses resultados demonstram a eficácia educacional do método PBL em sala de aulas. Os autores acreditam que os resultados do estudo podem servir como exemplo de um modelo que promove os conhecimentos e habilidades práticas em relação ao desenvolvimento de software.

Scanniello e Erra (2014) aplicaram uma estratégia baseada no *Think-Pair-Square* para verificar como o desempenho dos estudantes ao projetar diagramas de casos de uso. Nessa estratégia, os estudantes começaram com a modelagem individual (*Think*), depois trabalharam em pares (*Pair*) e, finalmente, trabalharam em um grupo com um número par de estudantes (*Square*) de quatro ou mais. Os autores perceberam que os diagramas de casos de uso foram aprimorados à medida que os estudantes passaram de um estágio para outro do processo, e os estudantes aprimoraram sua capacidade de trabalhar de forma colaborativa.

Balaban *et al.* (2015) também investigaram se o uso de anti-padrões apoia os estudantes a melhorar as habilidades de modelar diagramas de classes da UML. Para isso, os autores apresentaram um catálogo com 43 anti-padrões para os estudantes. O catálogo apoia os estudantes a analisar as causas dos problemas, e fornece sugestões sobre como corrigi-los. Para avaliar a eficácia dos anti-padrões, os autores realizaram dois estudos experimentos controlados. Com base nos resultados, os autores perceberam que os estudantes tiveram um bom desempenho utilizando os anti-padrões. Além disso, os estudantes comentaram que os anti-padrões ajudaram a refletir sobre os problemas identificados nos diagramas de classes.

García-Holgado *et al.* (2018) implementaram uma estratégia de aprendizagem ativa baseada em projetos para aumentar a taxa de sucesso de estudantes de Engenharia de Software. Os estudantes trabalharam no desenvolvimento de um modelo de análise (modelo de caso de uso e modelos de domínio das classes do problema) de um tópico específico. Os autores notaram um aumento na taxa de sucesso dos estudantes de 41,71% para 63,89%, relatando que, embora tenham aplicado a estratégia em apenas um contexto específico, a experiência pode ser adaptada a conteúdos semelhantes em outras universidades.

Fioravanti *et al.* (2018) relataram sua experiência aplicando o método *Project Based Learning* (PjBL) combinado com gerenciamento de projetos para criar um ambiente em que os estudantes lidem com gerentes e outras partes interessadas. O objetivo é aproximar os estudantes das experiências da vida real, desenvolvendo um projeto de software no contexto da modelagem de negócios. Os autores relataram que a inclusão do PjBL em sala de aula traz inovação e dinamismo na modelagem do ensino. Diferentemente das palestras tradicionais, o PjBL instiga os estudantes a assumir um papel proativo e a serem responsáveis pela aquisição de novos conhecimentos. Em geral, os estudantes estavam entusiasmados e tinham percepções positivas sobre o PjBL e a importância de realizar a modelagem em problemas reais.

Embora a literatura relate o uso de estratégias de aprendizagem ativa em um contexto de modelagem de software, foi observado que faltam estudos relatando como os professores conduziram o processo de seleção dessas estratégias de forma apropriada. Não foi identificado na literatura um local ou repositório em que os professores possam consultar sobre as estratégias ou encontrar o material didático necessário para aplica-las em sala de aula. Isso é essencial devido ao número de barreiras específicas que impedem os professores de empregar as estratégias de aprendizagem ativa. Além disso, nenhum estudo ainda analisou os desafios e as lições aprendidas relatados pelos professores ao empregar estratégias de aprendizagem ativas durante o ensino de modelagem de software. Vale a pena investigar esses desafios para que os professores possam compreender o uso dessas estratégias a partir do ponto de vista dos estudantes e de outros professores.

2.5 CONSIDERAÇÕES SOBRE O CAPÍTULO

Este capítulo teve como objetivo apresentar os fundamentos necessários para se obter uma melhor experiência de leitura desta tese de doutorado, tais como, educação em engenharia de software, modelagem de software, estratégias para o uso de modelagem de software. Além disso, foram apresentados alguns trabalhos relacionados que empregaram estratégias que visam auxiliar durante o processo de ensino de modelagem de software, utilizando os diagramas UML.

CAPÍTULO 3 – CICLO DE RELEVÂNCIA (PARTE I) – MAPEAMENTO SISTEMÁTICO SOBRE AS DIFICULDADES ENFRENTADAS PELOS ESTUDANTES DURANTE A MODELAGEM DE SOFTWARE UTILIZANDO OS DIAGRAMAS DA UML

Este capítulo apresenta os resultados do primeiro ciclo de relevância. Neste ciclo, foi conduzido estudo secundário (Mapeamento Sistemático da Literatura) com o objetivo de identificar as principais dificuldades ao aprender os diagramas da UML e catalogar as tecnologias propostas na literatura que visam minimizar essas dificuldades.

3.1 INTRODUÇÃO

Em DSR, um problema prático é o responsável por direcionar a pesquisa, e a partir dele surgirão outros problemas práticos e questões sobre o conhecimento. Para isso, iniciou-se a “investigação do problema”, que segundo Wieringa (2009), é a etapa em que há a busca de informação como um modo de entender o problema, sem ter ainda a capacidade de mudá-lo. Para isso, foram conduzidos três estudos com o objetivo de investigar e compreender a relevância do problema: um Mapeamento Sistemático da Literatura (que será abordado neste capítulo), Estudos Exploratórios (que será apresentado no Capítulo 4) e um Estudo Preliminar para avaliar a Viabilidade da Pesquisa (Capítulo 5).

Para conhecer o problema de pesquisa mais a fundo, um Mapeamento Sistemático da Literatura (MSL) foi conduzido. MSL é um método utilizado para categorizar e resumir as informações existentes sobre uma questão de pesquisa de maneira imparcial (Kitchenham e Chartes, 2007). A partir dos resultados do MSL, é possível identificar o estado da arte e lacunas de pesquisa, a fim de sugerir áreas para uma investigação mais aprofundada. Nesse sentido, foi conduzido um MSL com o intuito de identificar e categorizar as dificuldades enfrentadas pelos estudantes ao aprender a modelar os diagramas da UML. Além disso, classificou-se as tecnologias propostas na literatura visando minimizar essas dificuldades. Para a condução do MSL, considerou-se as diretrizes fornecidas por Petersen *et al.* (2015) e Kitchenham e Chartes (2007).

Para este MSL, foi utilizada a seguinte Questão de Pesquisa (QP):

Questão de Pesquisa: Quais são as dificuldades enfrentadas pelos estudantes e que podem influenciar no ensino/aprendizagem de diagramas UML?

A partir da QP apresentada acima, buscou-se identificar três aspectos nas publicações selecionadas: quais as dificuldades que estudantes (novatos ou não) têm durante a aprendizagem de diagramas UML; quais tecnologias estão sendo usadas para apoiar o ensino e aprendizagem de diagramas da UML; e, que tipo de estudos experimentais foram conduzidos usando diagramas UML ou as tecnologias propostas. Com base nesses três aspectos, foram definidas subquestões de pesquisa (SQ) para cada aspecto, a fim de responder a perguntas específicas, conforme pode ser visto na Tabela 3.1.

Tabela 3.1: Sub-Questões de Pesquisa.

Aspecto	Nº	Sub-questões de pesquisa
Dificuldades	SQ01	Quais são as dificuldades mencionadas no artigo?
Tecnologia	SQ02	Quais diagramas UML são abordados em cada artigo?
	SQ03	Há uma tecnologia ¹ que pode ser utilizada para apoiar no ensino/aprendizado dos diagramas?
	SQ04	A tecnologia é baseada em outra tecnologia existente?
	SQ05	Como a tecnologia pode ser utilizada (grupo ou individualmente)?
Experimentos	SQ06	O artigo apresenta uma avaliação experimental?
	SQ07	Qual o tipo de estudo experimental foi realizado?
	SQ08	Qual o perfil dos participantes do estudo?
	SQ09	Qual o tipo de resultado do estudo experimental?
	SQ10	O estudo experimental apresenta métrica(s) que visa(m) a avaliar os diagramas?

As atividades relacionadas as etapas de planejamento e condução do MSL podem ser encontradas no material de apoio (Silva *et al.*, 2020).

3.2 RESULTADOS DO MSL

Nas subseções a seguir serão apresentados os resultados deste Mapeamento Sistemático.

3.2.1 Tendências das publicações

Nesta seção, serão apresentadas as tendências de publicação sobre o tópico de pesquisa investigado neste MSL. Para fornecer uma visão geral do número e tipos de publicações, foram considerados todos os 181 estudos selecionados.

Pôde-se observar que os estudos selecionados foram publicados entre 1998 e 2018. De uma perspectiva temporal (Figura 3.1), observa-se que o ensino/aprendizagem de diagramas

¹ O termo “tecnologia” é utilizado como generalização para procedimentos, ferramentas, técnicas, metodologias e outras propostas realizadas na área de Engenharia de Software (Santos *et al.*, 2012).

UML—e as dificuldades e tecnologias que suportam esse processo—é uma área que não foi estudada extensivamente até 2003. Entre 1998 e 2003, observa-se uma média de 2 artigos por ano. Por outro lado, entre 2004 e 2017, houve um aumento notável e a média saltou para 10,64 publicações por ano. Este resultado evidencia o interesse científico e a necessidade de pesquisas sobre as dificuldades e tecnologias que apoiam o ensino da UML. Este mapeamento foi realizado em abril de 2019.

Os estudos analisados foram classificados para avaliar sua distribuição por (i) tipo de publicação (ou seja, periódico, conferência ou artigo do workshop) e (ii) locais de publicação direcionados. A Figura 3.2 mostra a distribuição temporal de acordo com o tipo de publicação. O tipo de publicação mais comum são os artigos de conferência com 120 (66,2%) estudos, seguido de artigos de periódicos com 44 (24,4%) e, finalmente, trabalhos de *workshops* com 17 (9,4%) estudos. Um número tão alto de artigos de conferências e periódicos pode indicar que esse tópico ainda é um tópico de pesquisa atualmente relevante, apesar de alguns estudos já terem sido publicados nos anos noventa.

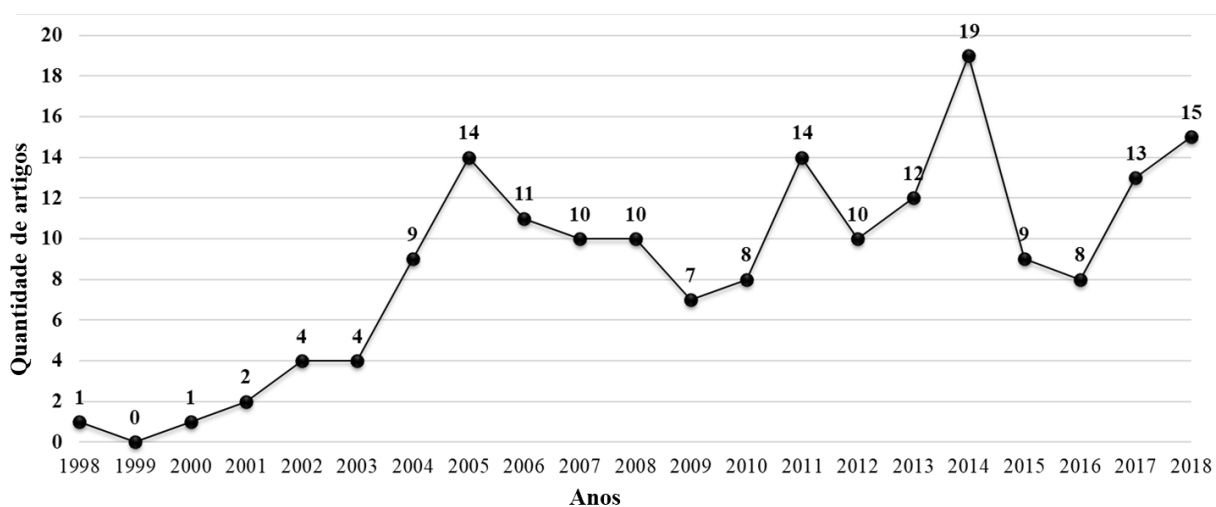


Figura 3.1. Distribuição dos estudos por ano.

A Tabela 3.2 mostra os locais de publicação que hospedaram mais de duas publicações. A partir da tabela, pode-se notar que a pesquisa está espalhada por um grande número de locais (101 locais para 181 publicações). Pode-se perceber que esse resultado é uma indicação de que o ensino/aprendizagem de diagramas UML é percebido como ortogonal para muitas áreas de pesquisa, em vez de um tópico de pesquisa específico. No entanto, a organização de comitês de eventos científicos (por exemplo, conferências ou *workshops*) totalmente dedicados à modelagem do ensino pode ajudar a fornecer uma identidade claramente definida da comunidade de pesquisa que trabalha neste tópico. Por exemplo, o *Workshop on Educators*

Symposium é um evento conjunto com a *International Conference on Model Driven Engineering Languages and Systems (MODELS)*, o fórum mais conhecido na área de modelagem de sistemas.

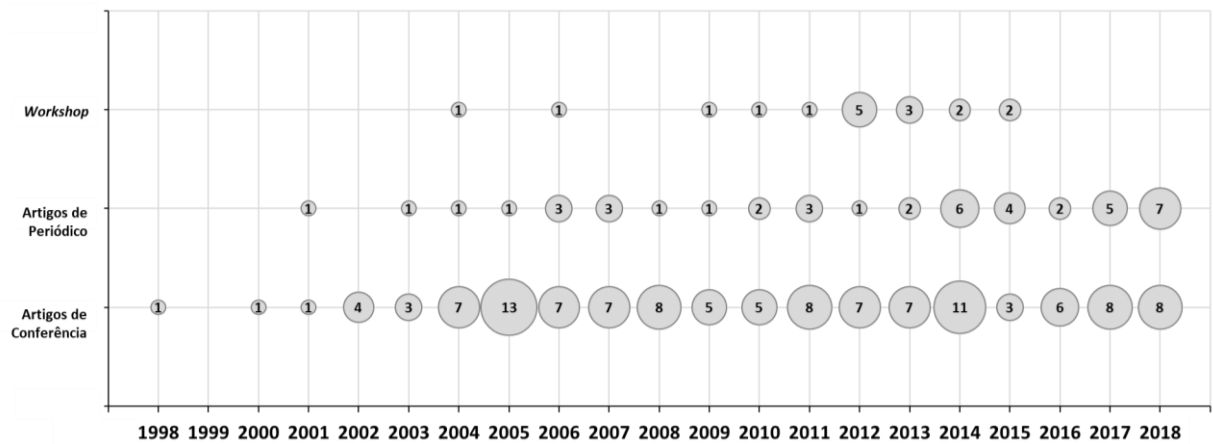


Figura 3.2. Distribuição dos estudos identificados por tipo de publicação.

Tabela 3.2. Locais de publicação com mais de duas publicações.

Local de Publicação	Tipo	#Publicações
International Conference on Model Driven Engineering Languages and Systems (MODELS)	Conferência	12
Conference on Software Engineering Education and Training (CSEE&T)	Conferência	09
International Conference on Software Engineering (ICSE)	Conferência	08
Conference on Innovation and Technology in Computer Science Education (ITiSCE)	Conferência	05
Workshop on Educators Symposium at MODELS	Workshop	05
Frontiers in Education Conference (FIE)	Conferência	04
Americas Conference on Information Systems (AMCIS)	Conferência	03
American Society for Engineering Education (ASEE)	Conferência	03
International Conference on Advanced Learning Technologies (ICALT)	Conferência	03
International Conference on Advanced Applied Informatics (IIAI-AAI)	Conferência	03
International Workshop on Requirements Engineering Education and Training (REET)	Workshop	03
Journal of Computing Sciences in Colleges	Periódico	03
Software and Systems Modeling	Periódico	03
Outros (locais com 2 ou menos)	-	117
Total	-	181

3.2.2 Visão Geral dos Resultados

Os resultados gerais, que são baseados na contagem dos estudos classificados em cada uma das respostas às subquestões de pesquisa, são apresentados na Tabela 3.3.

Tabela 3.3. Resultados do SM para cada uma das SubQuestões.

SubQuestões de Pesquisa	Possíveis respostas	Resultados	
		Quant.	%
SQ2. Qual diagrama UML é abordado em cada artigo?	Diagrama de Classes	99	81,82
	Diagrama de Casos de Uso	48	39,67
	Diagrama de Sequência	44	36,36
	Diagrama de Máquina de Estados	32	26,45
	Diagrama de Atividades	20	16,53
	Diagrama de Colaboração	7	5,79
	Diagrama de Objetos	7	5,79
	Diagrama de Pacotes	4	3,31
	Diagrama de Comunicação	3	2,48
	Diagrama de Componentes	3	2,48
	Diagrama de Implementação	2	1,65
	Diagrama de Implementação	1	0,83
	Diagramas da UML (geral)	53	43,80
	Descrição de Casos de Uso	12	9,92
SQ3. Existe alguma tecnologia para apoiar o ensino / aprendizagem de diagramas UML?	Sim	114	62,98
	Não	67	37,2
SQ4. A tecnologia é baseada em outra tecnologia existente? ¹	Sim	34	29,83
	Não	80	70,17
SQ5. Como a tecnologia deve ser usada? ¹	Individual	62	54,39
	Grupo	35	30,70
	Individual e/ou Grupo	8	7,02
	Pares	3	2,63
	Não Especificado	6	5,26
SQ6. O artigo apresenta alguma avaliação empírica?	Sim	135	74,59
	Não	46	25,41
SQ7. Que tipo de estudo empírico foi realizado? ²	Experimento	76	56,30
	Estudo de Caso	26	19,26
	Relato de Experiência	21	15,56
	<i>Survey</i>	11	8,15
	<i>Action Research</i>	1	0,74
SQ8. Qual o perfil dos participantes do estudo empírico? ²	Estudantes	106	77,94
	Profissionais	17	12,50
	Não Especificado	10	7,35
	Estudantes e Profissionais	3	2,21
SQ9. Qual é o tipo de resultado do estudo empírico? ²	Quantitativo	63	46,67
	Qualitativo	32	23,70
	Ambos (Quantitativo e Qualitativo)	40	29,63
SQ10. O estudo empírico apresenta métricas para avaliar os modelos?	Não	80	77,67
	Sim	23	22,33
Notas ¹ Os resultados das subquestões SQ4 e SQ5 foram baseados no número de tecnologias identificadas na SQ3. ² Os resultados das subquestões SQ7, SQ8 e SQ9 foram baseados no número de estudos experimentais identificados no SQ6. ³ Os resultados da SQ10 foram baseados na quantidade de estudos quantitativos e ambos (quantitativos e qualitativos) identificados no SQ9.			

Os parágrafos a seguir apresentam a análise dos resultados obtidos em cada subquestão de pesquisa. Observa-se que a SQ2 possui dados não exclusivos. Nesta subquestão, as respostas podem ser classificadas em um ou mais modelos, portanto, a soma das porcentagens é superior a 100%. Os resultados da SQ2 (Quais diagramas UML são abordados em cada estudo?)

mostraram que 82% (99 artigos) dos artigos seleccionados relatam o uso do diagrama de classes; cerca de 40% (48 artigos) empregam diagramas de casos de uso; e, cerca de 37% (44 artigos) relatam o uso do diagrama de sequência. Além disso, notou-se que cerca de 44% dos estudos utilizavam a UML como base de pesquisa, sem especificar o diagrama utilizado.

3.2.3 Dificuldades relacionadas à UML e seus diagramas

Nesta subseção, detalha-se cada uma das categorias de dificuldades encontradas. Antes de cada dificuldade, adicionou-se o ID do artigo de onde a dificuldade foi evidenciada. Todos os segmentos de texto relacionados às dificuldades identificadas podem ser encontrados no material de apoio (Silva *et al.*, 2020).

3.2.3.1 Dificuldades relacionadas aos diagramas UML

Essa categoria representa as dificuldades relacionadas à sintaxe e semântica da UML e às relações com os conceitos orientados a objetos (OO). A Figura 3.3 fornece uma visão geral deles. Nesta categoria, encontrou-se evidências de sete dificuldades diferentes que podem influenciar os engenheiros de software.

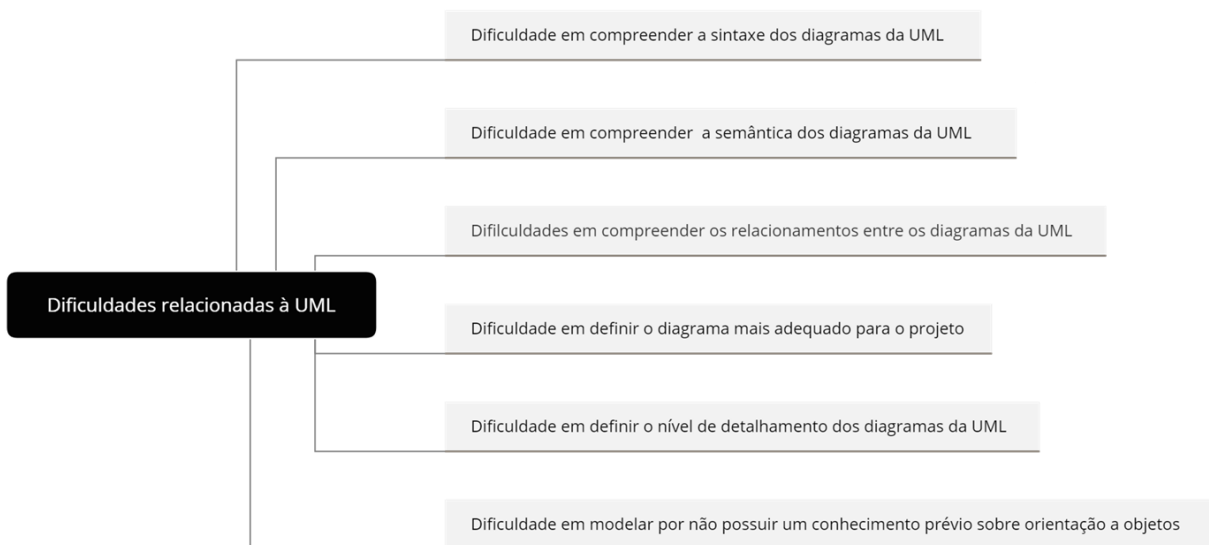


Figura 3.3. Tipos de Dificuldades relacionadas à aos diagramas da UML.

Em relação à **dificuldade em compreender a sintaxe dos diagramas UML** [S09, S24, S29, S30, S31, S40, S52, S68, S71, S79, S132, S147], Moisan e Rigault (S40) comentam que “o problema não é a própria linguagem, mas as notações arcanas da UML não são adequadas para não-desenvolvedores, e para essa população, geralmente restringe-se a diagramas clássicos de classe, sequência e atividade.” Petre [S09] realizou uma pesquisa e percebeu que a maior parte da discussão se refere à complexidade da notação: “UML falhou como uma

ferramenta de especificação (...) os detalhes da notação são complexos.” Em um estudo experimental realizado por Siau e Loo [S71], os participantes enfatizaram que *“eles tiveram problemas para memorizar as notações e construções na UML.”* Com base nos resultados do estudo, os autores concluem que um grande número de *constructos* da UML aumenta a dificuldade de aprender *“porque os humanos são limitados pelo tamanho de sua memória de curto prazo.”* Portanto, percebe-se que, para facilitar o aprendizado da UML, é necessário reduzir o número de *constructs*, resolver inconsistências entre *constructs* e diagramas, remover ambiguidades na UML e aumentar a prática durante o processo de aprendizagem.

Para a **dificuldade em compreender a semântica dos diagramas UML** [S26, S71], Siau e Loo [S71] perceberam que *“às semânticas de alguns constructs da UML não estão definidas com precisão.”* Além disso, os autores ainda relatam que *“a ambiguidade da semântica UML coloca problemas para os estudantes novatos também para a geração de códigos a partir de modelos.”*

Foi identificada também à **dificuldade para compreender os relacionamentos entre os diagramas da UML** [S24, S31, S40, S71, S126, S128, S132]. Boberic-Krstičev e Tešendić [S24] reportaram que os estudantes *“nem parecem saber que os diferentes tipos de diagramas UML, como sequência e diagramas de classes, estão relacionados entre si, embora cada um deles aborda um aspecto particular do mesmo sistema.”* O principal problema é que os estudantes aprenderam apenas notações de diagramas específicos e também aprenderam a modelar diagramas isolados um do outro. Além disso, Siau e Loo [S71] comentaram que *“diagramas inconsistentes e confusos na UML afetam a eficácia da compilação de conhecimento declarativo (por exemplo, definições, significados de diagramas e outros constructs da UML) em conhecimento processual (por exemplo, como desenhar Diagramas UML para modelar efetivamente um aspecto de um sistema).”* Os diagramas e *constructs* inconsistentes e confusos são obstáculos na modelagem usando UML.

Com relação à **dificuldade em definir o diagrama mais adequado para o projeto** [S19, S71, S132], em estudo realizado por Anda *et al.* [S132], alguns entrevistados relataram problemas com a escolha dos diagramas necessários ao projeto. Os entrevistados também mencionaram que *“eles não achavam que tinham uma compreensão suficiente do objetivo de aplicar cada um dos modelos e quando escolher quando aplicar qual.”*

Outra dificuldade identificada refere-se à **definição do nível de detalhe dos diagramas UML** [S26, S67, S132]. Anda *et al.* [S132] relatou que *“os diagramas UML eram frequentemente muito detalhados. Durante o design, nossos casos de uso e diagramas de*

sequência eram tão detalhados e incluíram já o design.” Segundo os autores, isso ocorreu porque “(a) as equipes começaram a pensar em código quando o o foco deveria estar na funcionalidade (...) (b) era difícil decidir como dividir a funcionalidade em casos de uso e quantos diagramas de sequência deveriam ser feitos para cada caso de uso.”

Com relação ao **conhecimento prévio de OO** [S67, S71, S83, S123, S128], Bolloju [S67] relatou que “a maioria dos erros pode ser atribuída à inexperiência de analistas iniciantes em habilidades de estruturação de problemas.” Siau e Loo [S71] afirmaram que “quando os participantes são introduzidos nos conceitos de UML, a ação principal é pesquisar e recuperar conceitos, conhecimentos ou regras de produção semelhantes na piscina de conhecimentos (memória de longo prazo) e, em seguida, tente criar uma nova regra de produção que combine os novos conceitos aprendidos com o conhecimento antigo.”

3.2.3.2 Dificuldades relacionadas ao diagrama de casos de uso

Essa categoria consiste nas dificuldades relacionadas especificamente ao diagrama de Caso de Uso (UC). Categorizou-se as dificuldades encontradas em Ator, Caso de Uso e Associações. A Figura 3.4 fornece uma visão geral das dificuldades identificadas.

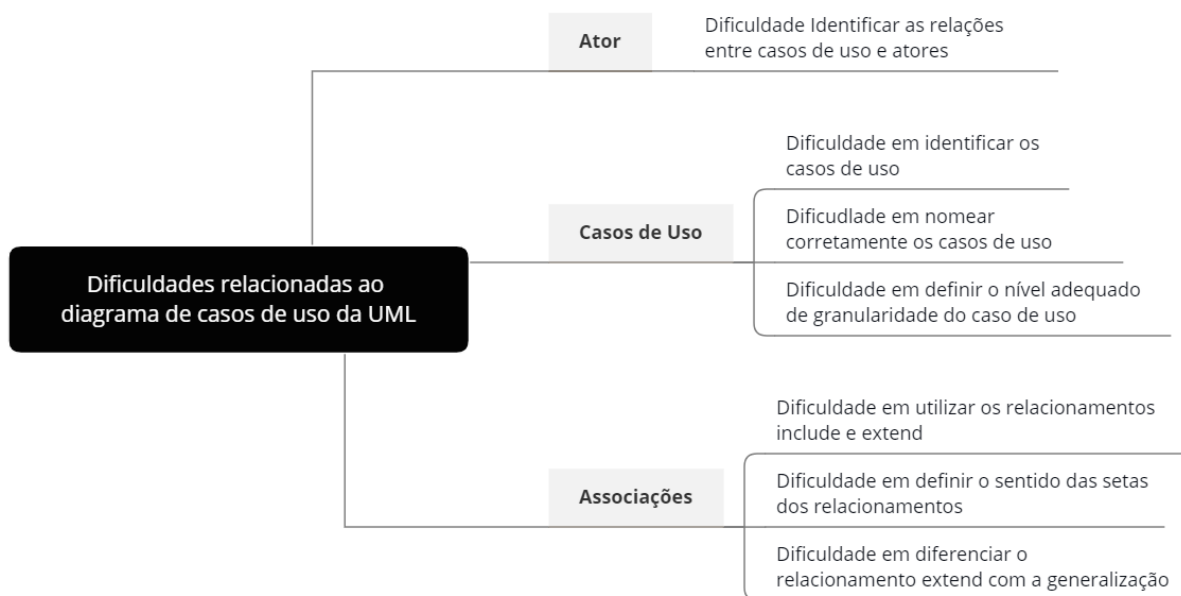


Figura 3.4: Tipos de dificuldades relacionadas ao diagrama de casos de uso da UML.

Em relação às **dificuldades relacionadas ao ator**, percebeu-se que os estudantes enfrentam problemas para identificar as relações entre casos de uso e atores [S19, S147]. Kruus *et al.* [S19] comentam que, quando estão modelando, a imaginação dos engenheiros não se limita à associação, eles usam todo tipo de dependência. Isso também é evidenciado por Mouenis *et al.* [S147], que argumentou que os engenheiros nem sempre entendem que “*existem*

casos de uso relevantes apenas para um subconjunto de atores e esses podem não necessariamente ter que ser humanos, mas podem ser, por exemplo, outros sistemas.”

Passando para a categoria **Caso de Uso**, notou-se que a dificuldade em identificar casos de uso [S79, S147] afetam a modelagem, como evidenciado por Boberic-Krstićev e Tešendić [S24] e Hallinan e Paul Gibson [S79]. Hallinan e Paul Gibson [S79] afirmaram que o principal desafio na modelagem de casos de uso era: *“aplicação correta para que houvesse o mínimo de funcionalidade necessária (...) os diagramas continham muita funcionalidade desnecessária.”* Com relação à nomeação correta de casos de uso [S19, S147], Mouenis *et al.* [S147] evidenciou que *“os estudantes encontraram dificuldade em nomear claramente os casos de uso.”* Kruus *et al.* [S19] relataram que os estudantes projetaram *“muitos casos de uso com nomes muito abstratos, que não permitem que ninguém que observe o diagrama de casos de uso e compreenda o que o sistema faz.”* Além disso, evidenciou-se que definir o nível apropriado de granularidade do caso de uso [S19, S24, S40, S132] também é um desafio para os estudantes. Boberic-Krstićev e Tešendić [S24] evidenciaram que *“os estudantes tendiam a derivar casos de uso menores e mais refinados, que representam funcionalidades atômicas.”* A partir de outra perspectiva, os autores também relataram que *“alguns estudantes tinham tendência a criar casos de uso mais amplos e abstratos, o que leva a casos de uso ambíguos e problemas no entendimento do sistema.”* Isso também ocorreu em um projeto descrito por Anda *et al.* [S132]. Os autores afirmaram que *“havia grandes diferenças entre os níveis de detalhes dos casos de uso das diferentes equipes. Algumas criaram apenas um caso de uso para uma função grande, mas que poderia ter sido dividida, enquanto outros criaram uma grande quantidade de casos de uso e casos de uso complexos para funções bastante simples.”* Isso pode ter ocorrido porque os estudantes tendem a modelar casos de uso como se estivessem especificando a interface do sistema.

Com relação às **dificuldades relacionadas às relações entre casos de uso**, percebeu-se que a dificuldade de usar as associações do tipo *include* e *extend* [S19, S24]. Boberic-Krstićev e Tešendić [S24] evidenciaram que *“durante a modelagem de casos de uso, é possível identificar relações de *include* e *extend*, em algumas situações, os estudantes não conseguiram decidir se incluem ou estendem a relação”*. Além disso, os estudantes confundem a direção das flechas nos relacionamentos [S19, S147]. De acordo com Mouenis *et al.* [S147] e Kruus *et al.* [S19], os estudantes cometem repetidamente erros na direção dos relacionamentos *include* e *extend*. Por fim, alguns estudantes não conseguem diferenciar os relacionamentos *include* e a

generalização [S147]: “Não houve distinção satisfatória entre o relacionamento ‘extend’ e o ‘generalização’.”

3.2.3.3 Dificuldades relacionadas ao diagrama de classes

Esta categoria representa os problemas relacionados a dificuldades ao modelar diagramas de classes. Identificou-se treze dificuldades e essas foram agrupadas em quatro categorias: Classes, Atributos, Associação e Métodos (ver Figura 3.5). Com relação às dificuldades no diagrama de classes, Sien [S31] realizou um estudo que investigou as dificuldades que os estudantes de graduação tinham ao modelar especificamente usando diagramas de classe e de sequência. Por isso, a maioria das dificuldades relacionadas aos diagramas foram coletadas a partir do estudo realizado por Sien [S31].

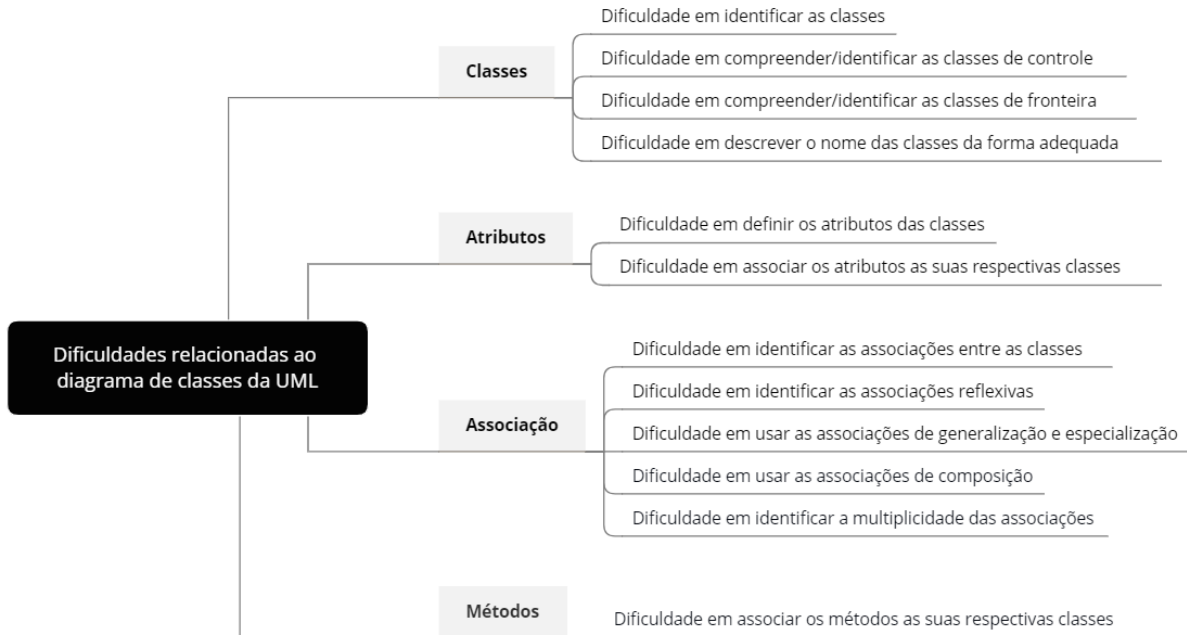


Figura 3.5. Tipos de dificuldades relacionadas ao diagrama de classes da UML.

No que diz respeito às **dificuldades relacionadas às classes**, primeiro, notou-se dificuldades relacionadas a identificação das classes [S07, S30, S31, S40, S150]. Por exemplo, Moisan e Rigault [S40] afirmaram que “*identificar classes no nível abstrato correto continua sendo uma questão difícil, mesmo para pessoas acostumadas a modelar,*” e que para realizar esse processo “*não há receitas absolutas.*” Além disso, Sien [S31] realizou um estudo e evidenciou que “*entre o grupo total de 52 estudantes, apenas 6% conseguiram identificar todas as oito classes candidatas esperadas, 31% conseguiram identificar sete classes esperadas e 4 % conseguiram identificar apenas uma classe adequada em seus diagramas de classes.*” Sien [S31] também concluiu que os engenheiros tinham dificuldade em compreender/identificar as

classes de controle [S24, S31], uma vez que “66% dos estudantes desse grupo não incluíam uma classe de controle em seus diagramas de classes, apesar de terem sido aconselhados a incluir uma classe para representar o sistema geral.”

Ainda em relação às dificuldades relacionadas às classes, evidenciou-se dificuldades em compreender/identificar as classes de fronteira [S24]. Boberic-Krstićev e Tešendić [S24] relataram “houve casos em que os estudantes ignoraram as classes de fronteira e criaram apenas classes de controle e entidade” durante a modelagem do diagrama de classes. Os autores comentaram que “a razão para isso, reside no fato de os estudantes não entenderem o papel das classes de fronteira.” Sien [S31] evidenciou que os estudantes tinham dificuldade em descrever o nome das classes adequadamente [S31]: “existem diagramas de classe com nomes de classes que não representam um conceito do mundo real (...) um estudante produziu algumas classes contendo nomes de casos de uso como nomes de classe.”

Em relação às **dificuldades relacionadas aos atributos**, os estudantes têm dificuldade em definir os atributos das classes [S07, S31]. Para ilustrar essa subcategoria, Sien [S31] percebeu que “os estudantes geralmente não incluem atributos essenciais nas classes” e “alguns estudantes produziram uma classe sem os atributos.” Além disso, Stikkolorum et al. [S07] complementou que os estudantes “também tendem a incluir um 'id' como parte dos atributos da classe.” A literatura também mostra que os estudantes têm dificuldade em associar atributos nas classes corretas [S04, S31]. A esse respeito, Sien [S31], descreveu que “os estudantes geralmente falham em incluir atributos essenciais nas aulas” ou definem os atributos em uma classe filha, mas esquecem que alguns atributos devem ser definidos em outras classes.

Em relação às **dificuldades relacionadas às associações (relacionamentos)**, Sien [S31] identificou a dificuldade em identificar as associações entre as classes [S04, S07, S30, S31] e relatou que “59% dos estudantes realizaram associações inadequadas e o diagrama de classes como um todo tinha pelo menos uma associação ausente.” No mesmo estudo, Sien [S31] evidenciou que os estudantes têm dificuldades em identificar associações unárias [S07, S31], enquanto realizam a modelagem do diagrama de classes: “nenhum dos estudantes modelou uma associação reflexiva necessária para a classe (exemplo do cenário utilizado pelos autores no estudo), mesmo que os requisitos declarassem que as tarefas atuais podem depender de tarefas que foram concluídas anteriormente.”

Também relacionados às dificuldades relacionadas às associações, alguns autores perceberam que os estudantes têm dificuldade em usar a especialização e a generalização [S07,

S19, S31, S57]. Alonso e Lemeunier [S57], por exemplo, relataram que *“os estudantes geralmente têm dificuldades em entender o significado da orientação dos relacionamentos.”* Sien [S31] relatou que os diagramas de classes produzidos pelos estudantes não incluíam muitas hierarquias do tipo generalizações/especializações. O autor relatou que *“entre os 27 diagramas modelados, 16 diagramas de classes tiveram a hierarquia definida adequadamente, 3 diagramas tinham uma mistura de hierarquias de herança apropriadas e inadequadas, e todas as hierarquias definidas nos outros 8 diagramas eram inapropriadas”* [S31]. Embora essa abordagem pareça simples e compreensível, os estudantes geralmente acreditam que definir hierarquias é uma tarefa muito mais difícil. Dessa maneira, percebe-se que os estudantes possuem dificuldades em agrupar objetos do mundo real em termos de classificação porque não são treinados adequadamente para agrupar objetos em hierarquias.

Também se identificou dificuldade em usar agregação e composição [S30, S31]. Em um estudo realizado por Boustedt [S30], *“apenas uma minoria dos participantes sabia a diferença entre o diamante branco e preto e muitos não conseguiam explicar o que os diamantes significam quando não recebem ajuda.”* Finalmente, no que diz respeito à dificuldade de identificar a multiplicidade das associações [S07], Stikkolorum *et al.* [S07] relatou que *“às vezes os estudantes sentem a necessidade de incluir uma notação não padrão nos campos de atributo, como notação de código ou números, em vez de usar o elemento de multiplicidade.”*

Quanto aos **métodos**, Stikkolorum *et al.* [S07] identificou que os estudantes têm dificuldades em associar os métodos às suas respectivas classes [S07]: *“os estudantes parecem ter dificuldades para identificar a responsabilidade de uma classe e salvar apenas essa classe para esse fim (coesão).”*

3.2.3.4 Dificuldades relacionadas ao diagrama de sequência

Esta categoria consiste nas dificuldades relacionadas ao diagrama de sequência. Identificou-se cinco dificuldades e agrupou-se em duas categorias: Objetos e Mensagens (Figura 3.6).

Nas **dificuldades relacionadas aos objetos**, percebeu-se que a dificuldade em identificar os objetos de controle [S31] foi relatado apenas no estudo de Sien [S31]. O autor mencionou que *“apenas 40% dos diagramas de sequência incluíam um objeto controlador apropriado (representando o controlador de fachada) em seus diagramas de sequência”*. Os estudantes também têm dificuldade em utilizar as entidades identificadas a partir do diagrama de classes [S31, S62].

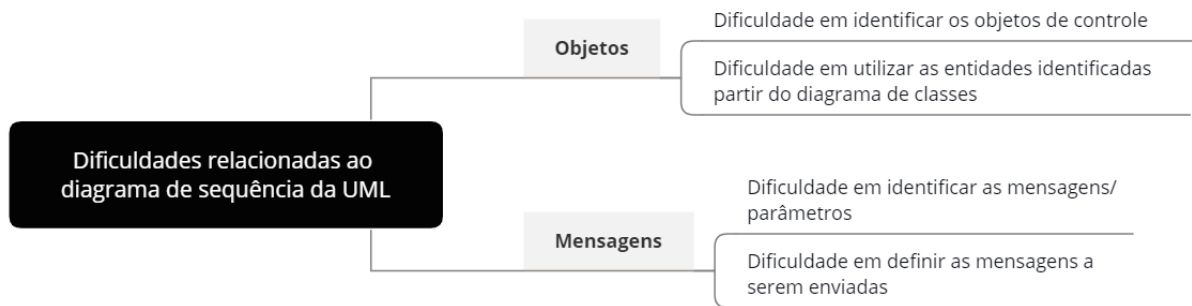


Figura 3.6. Tipos de dificuldades relacionadas ao diagrama de sequência da UML.

Em relação às **dificuldades relacionadas às mensagens**, notou-se que os trabalhos relataram dificuldade em identificar as mensagens / parâmetros [S31, S52, S62]. Sien [S31] afirmou que “*os estudantes geralmente têm dificuldade em identificar as mensagens a serem enviadas nos diagramas de sequência da UML. Eles não sabem como cumprir as responsabilidades do caso de uso, fazendo com que objetos passem mensagens uns para os outros.*” A literatura também mostra que os estudantes geralmente têm dificuldade em definir as mensagens a serem enviadas [S40, S67] nos diagramas de sequência da UML. Eles não sabem como cumprir as responsabilidades do caso de uso, fazendo com que os objetos passem mensagens entre si. Quanto à definição das responsabilidades entre as entidades do diagrama, Bolloju [67] evidenciou que “*a maioria dos erros pode ser atribuída à inexperiência de analistas iniciantes em habilidades de estruturação de problemas, como decomposição, e a dificuldades na aplicação de conceitos orientados a objetos relacionados à distribuição de responsabilidades entre os objetos participantes.*”

3.2.3.5 Dificuldades relacionadas aos diagramas de atividades, diagrama de colaboração, diagrama de comunicação e diagrama de máquina de estados

Com relação ao diagrama de atividades, foram identificadas três dificuldades relacionadas a duas categorias (fluxos de controle e nó de união). No que diz respeito à **dificuldade relacionada aos fluxos**, evidenciou-se que há dificuldade em identificar os fluxos alternativos [S24]. De acordo com Boberic-Krstićev e Tešendić [S24], “*alguns estudantes não entenderam que esses cenários se relacionam a um caso de uso, então eles criam diagramas de atividades separados para cada cenário.*” Em relação à **dificuldade no nó de união**, Kruus *et al.* [S19] percebeu que a maioria dos erros cometidos pelos estudantes está relacionado ao uso incorreto do elemento *join/fork*, resultando em “*descrições de interação em que o fluxo de interação nunca pode terminar, por exemplo, uma das transições de entrada para junção não*

chega, portanto, os eventos não são sincronizados e uma nova transição não pode ser iniciada.”

Em relação ao diagrama de colaboração, máquina de estado e comunicação, identificou-se uma dificuldade em comum: dificuldade em compreender o diagrama [S71]. Siau e Loo [S71] afirmou que “*os estudantes ficam confusos sobre quando usar os diagramas de colaboração/comunicação.*” Além disso, os autores comentam que essa dificuldade também é percebida no diagrama de máquina de estados. Especificamente, para o diagrama de colaboração, há uma certa dificuldade em perceber os blocos funcionais do sistema [S24], por esse motivo, em um estudo realizado por Boberic-Krstićev e Tešendić [S24], “*os estudantes que já tinham experiência em programação consideraram o diagrama de componentes como o diagrama mais difícil.*” Os autores acrescentaram que isso se deve ao fato de que os estudantes “*não conseguem perceber diferentes funções funcionais. blocos do sistema.*”

Sobre os demais diagramas, não foram encontradas dificuldades específicas para os mesmos. Por este motivo, não são relatados resultados para os mesmos.

3.2.4 Resultados das tecnologias identificadas no MSL

Observou-se que os estudos selecionados apresentam, ao todo, 95 tecnologias diferentes para auxiliar estudantes, profissionais de software e professores durante a modelagem de software ao empregar os diagramas da UML. As tecnologias identificadas foram classificadas em três categorias: (a) Ferramentas, (b) Métodos, Diretrizes e Abordagens e (c) *Design Instrucional*. Uma lista completa das tecnologias identificadas pode ser encontrada no material de apoio (Silva *et al.*, 2020).

Ao todo, 52 tecnologias foram classificadas como **Ferramentas**. Após identificar as ferramentas, percebeu-se que as mesmas podem ser de três tipos:

- Ferramentas CASE (*Computer-Aided Software Engineering*) (13): ferramentas que apoiam o processo de desenvolvimento dos diagramas de maneira manual, semiautomática e automática.
- Sistemas de tutoria de modelagem (33): ferramentas que guiam os estudantes durante a elaboração dos diagramas e guiando na identificação dos *constructos* dos diagramas.
- Sistemas de avaliação dos diagramas (6): ferramentas que orientam os estudantes durante a modelagem dos diagramas e guiam também na correção dos defeitos identificados automaticamente nos diagramas criados.

Com relação às tecnologias que se enquadram como **Métodos, Diretrizes e Abordagens**, foram encontradas 37 tecnologias. Agrupou-se essas tecnologias em três subcategorias:

- Métodos educacionais (19): etapas predefinidas necessárias para a correta aplicação do método.
- Diretrizes educacionais (14): procedimentos para guiar os estudantes nos durante o processo de ensino/aprendizagem de determinado diagrama.
- Abordagens educacionais (4): abordagens usadas por professores que podem ajudar os estudantes a entender melhor a sintaxe e a semântica dos diagramas.

Por fim, as tecnologias que se enquadram categoria **Design Instrucional** são as que descrevem os formatos dos cursos. Ou seja, os artigos relataram um formato novo/aprimorado para cursos que visam melhorar a maneira de ensinar diagramas da UML. Seis tecnologias foram encontradas para a categoria. Por exemplo, Sikkel e Daneva [S33] relatam que, durante muitos anos, ensinaram um grande número de estudantes (150-180) de um curso de Sistemas de Informação, no qual os estudantes aprendem a fazer diagramas UML a partir de especificações verbais. No entanto, em 2009, os autores alteraram o formato do curso para incluir tarefas que deveriam criar uma percepção mais realista do ciclo de vida dos diagramas da UML. No novo formato do curso, os estudantes recebem um caso real, com o objetivo de realizar a primeira tentativa de elaborar um diagrama. Em seguida, conversaram com os representantes dos clientes, que podem responder a perguntas e preencher o contexto e os detalhes que ainda não conhecem.

3.2.5 Métricas para avaliar diagramas UML

Para ajudar os engenheiros de software e professores a avaliar a qualidade de diagramas elaborados, nos artigos selecionados foram identificadas e classificadas métricas utilizadas para avaliar a qualidade dos diagramas. Foram considerados apenas os estudos que apresentaram resultados quantitativos ou ambos (quantitativos e qualitativos), uma vez que as métricas não são utilizadas em estudos de natureza qualitativa. Por conta disso, percebeu-se que 23 artigos relatam métricas para avaliar os diagramas. No contexto desse MSL, o termo “métrica” é definido como uma medida de extensão ou grau em que um produto (o foco aqui é em diagramas da UML) possui e exibe uma certa característica de qualidade (Boehm *et al.*, 1976).

A Tabela 3.4 mostra que o maior número de métricas identificadas foi proposto para avaliar o diagrama de classes (38,71%), seguido pelo diagrama de atividades (19,35%),

diagrama de casos de uso (16,12%) e diagrama de sequência (13,90%). Identificou-se apenas uma métrica para cada um dos outros diagramas (máquina de estados, comunicação e pacotes) e para a descrição de casos de uso. As métricas identificadas nos estudos foram agrupadas pelo tipo de diagrama ao qual cada um está relacionado, como pode ser visto na Tabela 3.5. Vale ressaltar que existem várias métricas que têm nomes diferentes, mas medem o mesmo conceito (por exemplo, em alguns trabalhos, o percentual de respostas corretas é chamado de “*appropriateness*”, enquanto outros trabalhos chamam essa métrica de “*pedagogical efficacy*”). Para elaborar a Tabela 3.5 e contar quantos artigos usam a mesma medida, os pesquisadores agruparam as métricas que consideraram o mesmo conceito. Grupos de métricas com o mesmo nome, juntamente com a definição de cada medida e os documentos em que foram definidas, são detalhados no material de apoio (Silva *et al.*, 2020).

Tabela 3.4. Resultados por tipo de diagrama.

Tipo de Diagrama	Qtd de artigos	Porcentagem	Lista dos Artigos
Diagrama de Classes	13	38,71	S15, S55, S69, S77, S01, S32, S01, S02, S03, S10, S73, S31, S37
Diagrama de Atividades	06	19,35	S10, S27, S46, S47
Diagrama de Casos de Uso	05	16,12	S01, S16, S10, S169
Diagrama de Sequência	04	12,90	S10, S69, S31, S37
Diagrama de Máquina de Estados	01	3,23	S10
Diagrama de Comunicação	01	3,23	S10
Diagrama de Pacotes	01	3,23	S10
Descrição de Casos de Uso	01	3,23	S69
Total	31	100.00	

Na Tabela 3.5, foram identificados oito métricas para avaliar o diagrama de classes. A métrica mais citada foi a baseada em “*Errors*.” Com relação ao diagrama de atividades, foram identificados cinco métricas: a métrica de “*Adequacy*” foi citada em dois artigos (S46 e S47) e as outras métricas foram discutidas apenas uma vez. Evidenciou-se cinco métricas para avaliar o diagrama de casos de uso, três métricas para o diagrama de sequência e uma métrica para os outros diagramas.

Tabela 3.5. Métricas identificadas para avaliar os diagramas da UML.

Tipo de Diagrama	Métrica	#artigos	Lista dos artigos
Diagrama de Classes	Errors	05	S15, S55, S69, S77, S32
	Adequacy	02	S31, S37
	Correctness	01	S01,
	Completeness	01	S01
	Quality attributes	01	S02
	F-Measure	01	S03
	RUBRIC	01	S10
	Efficacy	01	S73
Diagrama de Atividades	Adequacy	02	S46, S47
	RUBRIC	01	S10

Tipo de Diagrama	Métrica	#artigos	Lista dos artigos
	Efficiency	01	S27
	Efficacy	01	S27
	Comprehensibility	01	S27
Diagrama de Casos de Uso	Correctness	01	S01
	Completeness	01	S01
	F-Measure	01	S16
	RUBRIC	01	S10
	Errors	01	S69
Diagrama de Sequência	Adequacy	01	S31, S37
	Errors	01	S69
	RUBRIC	01	S10
Diagrama de Máquina de Estados	RUBRIC	01	S10
Diagrama de Comunicação	RUBRIC	01	S10
Diagrama de Pacotes	RUBRIC	01	S10
Descrição de Casos de Uso	Errors	01	S69

Com base nesses resultados, pode-se dizer que a maioria das medidas se baseia em métricas objetivas, por exemplo, o número de respostas ou perguntas corretas (para a métrica *F-Measure*). Por outro lado, existem estudos que utilizam variáveis subjetivas (por exemplo, a métrica de “*adequacy*”), relacionadas às percepções dos participantes sobre a variável medida.

3.2.6 Mapeamento dos resultados quantitativos

Após categorizar as tecnologias, as subquestões de pesquisa foram combinadas para fornecer uma visão geral das tecnologias propostas com o objetivo de auxiliar no ensino de modelagem. Esse mapeamento permitiu obter mais informações sobre como os resultados de cada subquestão estão relacionados aos demais e quais são as possíveis lacunas na pesquisa.

A Figura 3.7 mostra os resultados do mapeamento obtidos a partir da SQ2 (diagramas UML) em comparação com a SQ3 (Tecnologias para apoiar o ensino/aprendizagem de diagramas de UML) e a SQ5 (Uso da tecnologia).

Esses resultados indicam que:

- A maioria dos estudos encontrados relatam o desenvolvimento de tecnologias que são usadas para apoiar a modelagem de diagrama de classes, diagrama de casos de uso e diagrama de sequência. Além disso, há uma escassez de tecnologias aplicadas à modelagem de diagramas de pacotes, comunicação e implantação. Isso pode ter acontecido porque são realizados menos estudos experimentais com estes diagramas.
- Uma parte das tecnologias foi projetada para apoiar a modelagem de vários diagramas UML. Em relação à essas tecnologias, isso ocorreu principalmente no caso das ferramentas CASE, pois elas apoiam automaticamente o *design* dos diagramas e realizam a rastreabilidade entre esses diagramas. Por exemplo, os diagramas de

colaboração permitem que os estudantes entendam cada caso de uso, entendam como rastrear a rastreabilidade entre os diagramas em diferentes fluxos de trabalho, percebam o relacionamento básico das classes, que formam uma base sólida para identificar as classes do projeto e visualizem como a decomposição do formulário inicial o sistema está pronto.

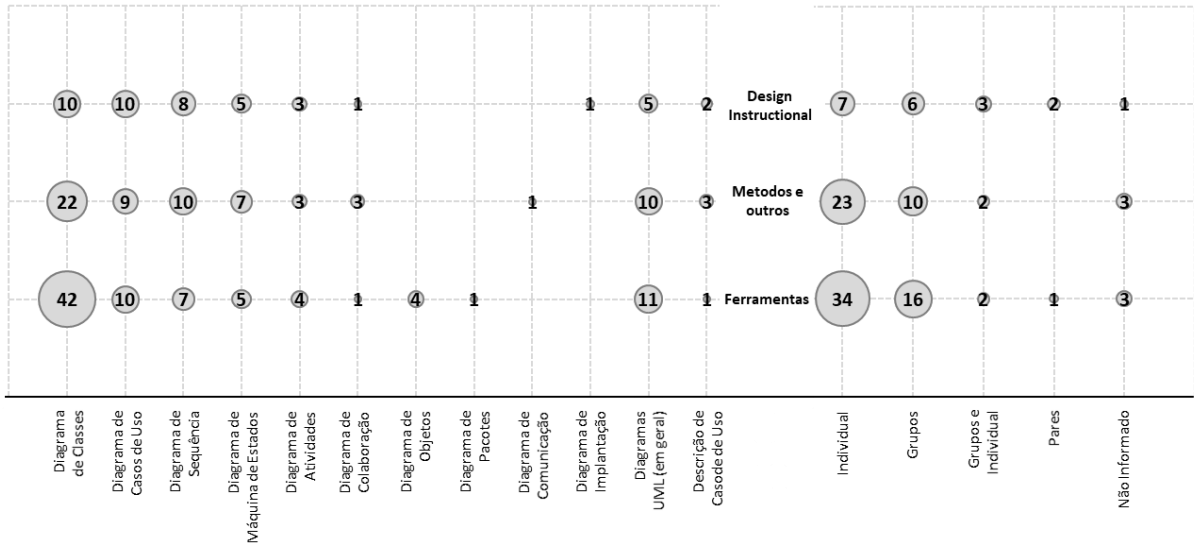


Figura 3.7. Mapeamento dos resultados obtidos a partir da combinação das subquestões de pesquisa SQ2, SQ3 e SQ5.

- A maioria das tecnologias é projetada para ser usada por um único usuário, ou seja, essas tecnologias não permitem modelagem colaborativa.

A Figura 3.8 mostra os resultados do mapeamento obtidos da SQ3 (Tecnologias para apoiar o ensino/aprendizagem de diagramas UML) em comparação com a SQ4 (Tecnologia baseada em outras existentes) e SQ6 (Avaliação empírica).

Esses resultados indicam que:

- A maioria das tecnologias identificadas foi baseada em outras tecnologias existentes. Os resultados indicaram que várias tecnologias estão sendo propostas na literatura. Além disso, percebeu-se que as novas tecnologias foram desenvolvidas para remediar os problemas/deficiências identificados nas tecnologias mais antigas. Isso foi observado principalmente com as tecnologias que ajudam os engenheiros durante a elaboração dos diagramas. Durante a modelagem do diagrama, essas tecnologias apoiam computacionalmente os engenheiros, indicando possíveis erros sintáticos, por exemplo. Além disso, essas tecnologias visam auxiliar os professores no processo de ensino, além

de estudantes e engenheiros experientes, apoiando o processo de aprendizado desses diagramas.

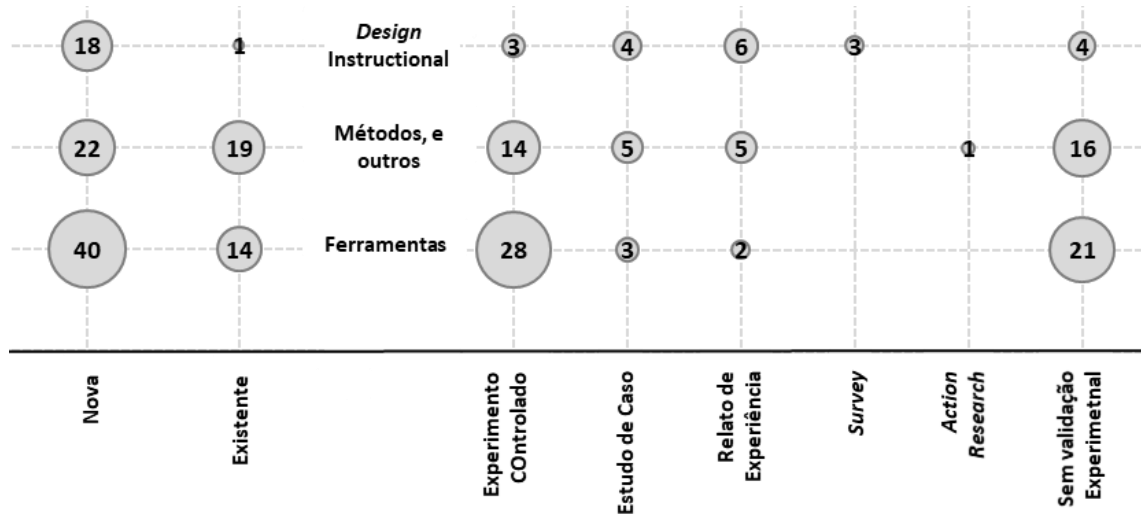


Figura 3.8. Mapeamento dos resultados obtidos a partir da combinação das subquestões de pesquisa SQ3, SQ4 e SQ6.

- Para a maioria das tecnologias, algum tipo de avaliação foi realizado. Esses resultados mostraram que os pesquisadores estão conduzindo estudos experimentais com as tecnologias que estão propondo. Isso pode indicar que os pesquisadores estão preocupados em melhorar as tecnologias propostas para ajudar estudantes e professores durante a modelagem.
- Existem algumas tecnologias que não foram avaliadas por meio de experimentos controlados, estudos de caso ou qualquer tipo de estudo experimental. Portanto, é necessário realizar estudos experimentais com essas tecnologias para avaliar como elas se comportam e também para mostrar como elas ajudam os estudantes durante o aprendizado do design de diagramas da UML.

A Figura 3 mostra os resultados obtidos SQ6 (avaliação empírica) em comparação com a SQ7 (Tipo de resultado) e SQ8 (Perfil dos participantes). Esses resultados indicam que:

- A maioria dos estudos empíricos se baseou em dados quantitativos. Os dados quantitativos foram utilizados para medir a viabilidade do uso das tecnologias. Por outro lado, poucos estudos realizam análise qualitativa dos dados. A análise qualitativa é importante porque os pesquisadores podem usar os resultados para obter uma melhor compreensão sobre como os estudantes e professores entendem o processo de modelagem dos diagramas UML (Houston *et al.*, 2001; Wholin *et al.*, 2012).

- Esses resultados mostraram que um número relativamente pequeno de estudos empíricos com professores e profissionais foi realizado nos estudos publicados, em contraste com um grande número de profissionais que usam a UML durante o processo de desenvolvimento (Hainey *et al.*, 2011).

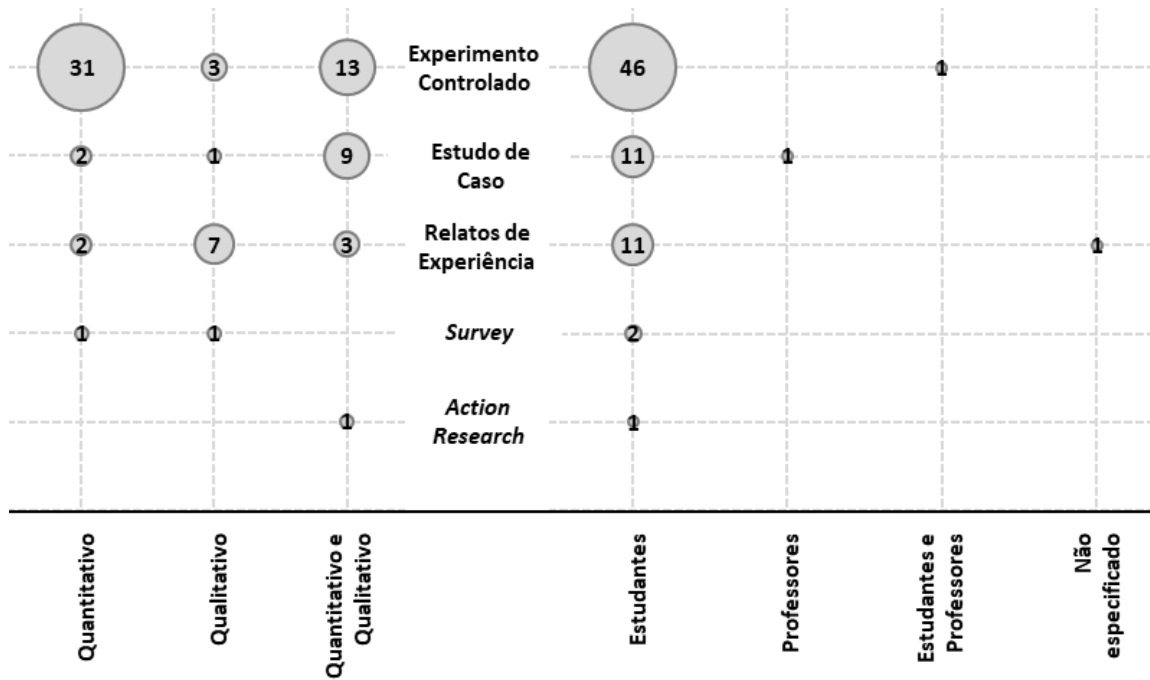


Figura 3.9. Mapeamento dos resultados obtidos a partir da combinação das subquestões de pesquisa SQ6, SQ7 e SQ8.

3.3 DISCUSSÃO DOS RESULTADOS DO MSL

O corpo de conhecimento deste trabalho é baseado na análise de 181 artigos selecionados. Cada trabalho foi selecionado de acordo com critérios de inclusão e exclusão previamente definidos de forma imparcial. Mais especificamente, cada artigo selecionado é um artigo científico revisado por pares, no qual várias partes interessadas gerenciaram, colaboraram e tomaram conhecimento do trabalho uma da outra.

Como resultado desse mapeamento sistemático, foram identificados artigos que relatam as dificuldades percebidas pelos estudantes ao projetar os diagramas UML. Esse é um tópico que pode merecer uma pesquisa mais focada. Com relação às dificuldades relacionadas aos diagramas, não surpreende que os diagramas de classe, diagramas de sequência e diagramas de casos de uso tenham sido os mais diagramas com mais dificuldades relatadas. Guo *et al.* (2017), Petre (2014) e Boberic-Krstićev e Tešendić (2013) relataram que esses três diagramas são os mais frequentemente usados para a modelagem de sistemas. Isso pode ocorrer porque esses diagramas representam as três visualizações UML mais comuns (Ma, 2017): visualização

funcional (diagrama de casos de uso), visão estrutural (diagrama de classes) e visão comportamental (diagrama de sequência).

Utilizando a estrutura proposta por Lindlland *et al.* (1994), que avalia a qualidade dos modelos conceituais de maneira sistemática, percebeu-se que as dificuldades identificadas estão mais relacionadas às dificuldades semânticas do diagrama. Isso significa que os estudantes têm dificuldade em entender os significados semânticos dos diagramas UML da maneira esperada. Com relação ao diagrama de casos de uso, as dificuldades semânticas estão relacionadas, por exemplo, ao uso correto dos relacionamentos de *include* e *extend* ou para definir corretamente o nível de granularidade dos diagramas de casos de uso. Em relação à essa dificuldade, os estudantes normalmente não seguem estritamente as regras para identificar casos de uso (Knauus *et al.*, 2008), e alguns dos estudantes argumentam que um caso de uso grande pode ser dividido em vários pequenos casos de uso. No que diz respeito ao diagrama de classes, dificuldades semânticas são enfrentadas, por exemplo, quando estudantes identificam incorretamente a multiplicidade entre associações ou associam atributos incorretamente nas classes (Bolloju, 2006). A consequência dessas dificuldades é que os estudantes podem projetar diagramas de classes incompletos que não representam o domínio do problema. As classes são abstrações de coisas objetivas em domínios problemáticos e, portanto, geralmente precisam ter suas próprias propriedades e comportamento mapeados (Knauus *et al.*, 2008). Para o diagrama de sequência, os estudantes têm dificuldade em identificar as mensagens que serão enviadas entre os objetos de classe relacionados e em definir as responsabilidades entre as entidades do diagrama. A principal causa de defeitos semânticos é quando os estudantes definem classes e desenham diagramas de sequência relacionados e modificam os nomes das operações nos objetos ou mensagens nos diagramas de sequência sem ajustar no diagrama de classes (Boberic-Krstićev e Tešendić, 2013; Moisan e Rigault, 2009). Da mesma forma, engenheiros iniciantes geralmente modificam os relacionamentos entre classes ou mensagens entre objetos de classe e esquecem de ajustar as classes relacionadas. Segundo Ma (2017), grande parte dessas dificuldades ocorre, porque a semântica lida com os significados por trás dos elementos do modelo, ou seja, o mapeamento semântico dos diagramas da UML para os domínios modelados.

A partir das dificuldades coletadas, pode-se observar que as dificuldades mais relacionadas à semântica dos diagramas foram relatadas por profissionais. Isso pode estar relacionado ao fato desses profissionais já usarem ferramentas de modelagem orientada a objetos que ajudam a evitar muitos defeitos sintáticos. No entanto, vale ressaltar que estudos anteriores mostraram que, mesmo com a ajuda de ferramentas, alguns diagramas ainda podem

conter defeitos sintáticos (Ma, 2017). Algumas dessas ferramentas não identificam todos os tipos de defeitos sintáticos. Quanto às dificuldades percebidas pelos estudantes, percebeu-se que essas dificuldades estavam relacionadas tanto a sintaxe, quanto a semântica dos diagramas. De acordo com a literatura, os estudantes têm dificuldade em entender a sintaxe dos diagramas, pois isso pode exigir, em alguns casos, experiência ou uma experiência prévia em programação. Às vezes, representar através de um diagrama pode ser tão complexo para os estudantes que uma representação em uma linguagem de programação (ou pseudocódigo) pode ser mais clara e natural (Ma, 2017). Com relação às dificuldades semânticas enfrentadas pelos estudantes, isso ocorreu, pois os estudantes não sabem se certas representações são logicamente corretas. Por exemplo, o uso de uma junção ou agregação depende de como o objeto deve ser armazenado e como o objeto deve ser usado. Não é fácil para iniciantes da UML entender as implicações de certas representações e quando usar certas representações.

Os resultados obtidos no MSL ajudaram a identificar tecnologias que visam apoiar os professores na melhoria do processo de ensino de modelagem, usando diagramas UML. Percebeu-se que as tecnologias foram constantemente modificadas para oferecer melhor apoio ao *design* e ao aprendizado de diagramas UML. Portanto, não existe uma tecnologia única que seja adequada para minimizar todas as dificuldades que os estudantes enfrentam durante a modelagem de diagramas UML. A utilização ou não de uma determinada tecnologia depende da finalidade que os professores almejam e do tipo de artefato que será ensinado (por exemplo, diagrama de classes, diagrama de casos de uso).

Com o intuito de ajudar os professores, a Tabela 3.6 mostra uma relação feita com o objetivo de elencar as tecnologias que podem ser utilizadas para minimizar as principais dificuldades enfrentadas pelos estudantes ao realizar a modelagem dos diagramas UML (dificuldades genéricas percebidas em todos os diagramas), do diagrama de casos de uso, diagrama de classes, diagrama de sequência e diagrama de atividades.

Esses diagramas foram selecionados, pois os resultados mostraram que os estudantes enfrentam mais dificuldades em realizar a modelagem dos mesmos. A partir do mapeamento Dificuldade X Tecnologia, percebeu-se que há diversas tecnologias que estão sendo propostas na literatura, porém as mesmas não estão sendo propostas para minimizar uma dificuldade, específica ou não, para um determinado diagrama. Na maioria das vezes, os pesquisadores estão simplesmente realizando a proposição da nova tecnologia. Além disso, a maioria das tecnologias não há uma evidência experimental mostrando a sua utilidade e facilidade de uso.

Tabela 3.6. Mapeamento entre dificuldades enfrentadas pelos estudantes e as tecnologias propostas na literatura.

Categorias	Dificuldades principais	Tecnologias propostas para minimizar as dificuldades		
		Ferramentas	Métodos, Diretrizes e Abordagens	Design Instrucional
Dificuldades relacionadas aos diagramas UML	<ul style="list-style-type: none"> • Dificuldade em compreender a sintaxe dos diagramas UML • Dificuldade em compreender a semântica dos diagramas UML • Dificuldade para compreender os relacionamentos entre os diagramas da UML • Dificuldade em definir o diagrama mais adequado para o projeto • Definição do nível de detalhe dos diagramas UML • Necessidade de conhecimento prévio de OO 	<p>Ferramentas Case:</p> <ul style="list-style-type: none"> • Papyrus (S08) • Rational Rose S169 • Vision (S169) <p>Sistemas de Avaliação dos diagramas:</p> <ul style="list-style-type: none"> • Method to detect learners' difficult points (S17) • Tool to detect learners' difficult points. (S17) • SmallTEAmsHelper (S179) <p>Sistemas de tutoria de modelagem:</p> <ul style="list-style-type: none"> • diagnostic method + Diagram (S58) • TextOO (S72) • Ideogramic UML (S88) • TeDUB (S90) • QSEE-SuperLite (S101) • Agent-Oriented Collaborative Learning System (S154) • A Prototype Object-Oriented Design Learning Environment (OODLE) (S157) • Online Img2UML Repository (S25) 	<p>Métodos Educacionais:</p> <ul style="list-style-type: none"> • Problem-based learning (S116, S189) • Learning from Erroneous Examples (S180) • Method to detect learners' difficult points (S17) • PBL + LBL Double Track Teaching Method (S116) • Project-based learning with weekly quizzes, tests and active learning tasks (S137) <p>Diretrizes Educacionais:</p> <ul style="list-style-type: none"> • Principles for teaching object-oriented design (S117) • Best Practices for Teaching UML (S69) <p>Abordagens Educacionais:</p> <ul style="list-style-type: none"> • Pantomime (S100, S110) • Blended Learning Approach (S80) • Interactive lecture workshop (S140) • Reverse Engineering Techniques (S100) 	<ul style="list-style-type: none"> • Format of a course (S33, S59, S89, S98, S99) • Process of UML 2 Teaching (S66, S77) • Instructional Design of Exercise-Centric Teaching Materials on UML Modeling (S10) • Modeling Education Program for Novices (S23)
Dificuldades relacionadas ao diagrama de casos de uso	<ul style="list-style-type: none"> • Dificuldades relacionadas ao ator • Dificuldades relacionadas aos Casos de uso • Dificuldades relacionadas às relações entre casos de uso 	<p>Ferramentas Case:</p> <ul style="list-style-type: none"> • IDE for OOAD (S01) • Papyrus (S08) • UMLet (S86) • QuickUML (S96) <p>Sistemas de tutoria de modelagem:</p> <ul style="list-style-type: none"> • StudentUML (S05) • UML-IT (S06, S108) • SPACE-DESIGN (S13) • Tablet PCs e Web Conference Software (S63) • Prototype of adaptive online module based on the student's learning style categories (S142) • Online Learning System (S162, S166, S180) 	<p>Métodos Educacionais:</p> <ul style="list-style-type: none"> • Teaching Methods of Object-Oriented Approach in MIS Curriculum (S107) • Method based on think-pair-square (S16) • Rules-based pedagogical method for identifying defects in models (S49) • Learning Methods using inspection data (S81) <p>Diretrizes Educacionais:</p> <ul style="list-style-type: none"> • Use Case Writing Guidelines (S38) • Guidelines for conducting training (S40) • Best Practices for Teaching UML (S69) • UML-Based Team Project (S118) 	<ul style="list-style-type: none"> • Format of a course (S33, S64, S92, S94, S111) • Elicitation Teaching for Object-Oriented Analysis and Design (S54) • Framework for student projects (S12) • Case Method Approach (S43) • UML modeling template (S48)

<p style="writing-mode: vertical-rl; transform: rotate(180deg);">Dificuldades relacionadas ao diagrama de classes</p>	<ul style="list-style-type: none"> • Dificuldades relacionadas às classes • Dificuldades relacionadas aos atributos • dificuldades relacionadas às associações (relacionamentos) • Dificuldades relacionadas aos métodos 	<p>Ferramentas Case:</p> <ul style="list-style-type: none"> • IDE for OOAD (S01) • LogViz (S04) • UML editor WebUML (S07) • UMLet (S86) • Multi-Touch UML Editor (S135) • Spreadsheet-based modeling approach (S143) <p>Sistemas de Avaliação dos diagramas:</p> <ul style="list-style-type: none"> • Graph matching system for class diagram (S136) • A prototype for the automated assessment of UML class diagrams (S139) • Method to detect learners' difficult points (S17) • Approach and algorithms for automating the selection of the best sub-diagrams (S163) <p>Sistemas de tutoria de modelagem:</p> <ul style="list-style-type: none"> • StudentUML (S05) • UML-IT (S06, S108) • Intelligent Problem-Solving Environment for UML Class Diagrams (S28) • COLLECT-UML (S36, S61, S65, S76, S130) • Courseware for UML/SysML modelling (S44) • UML Checker (S50) • UML Tutor (S50) • PatternCoder (S55) • AUTO-COLLEAGUE (S56, S60, S129) • Diagram UML (S57) • diagnostic method + Diagram (S58) • TextOO (S72) • Ideogramic UML (S88) • Interactive Exercise (S103) • UMLGrader (S112) • CIMEL-ITS (S161) • Tool of Model-Driven Requirements Analysis Using Unified Modeling Language (S29) • MERODE (S122, S123, S125) • Geon (S84) • VisAr3D environment (S141) • Prototype of adaptive online module based on the student's learning style categories (S142) • Online Learning System (S162, S166, S180) • Tool for create the class diagram (S163) 	<p>Métodos Educacionais:</p> <ul style="list-style-type: none"> • Teaching Method for Conceptual Modeling Based on Error Analyses of Class Diagrams (S15) • Teaching Methods of Object-Oriented Approach in MIS Curriculum (S107) • Rules-based pedagogical method for identifying defects in models (S49) • Positive and Negative Examples (S148, S149) • Jigsaw Technique (S153) • Learning Methods using inspection data (S81) <p>Diretrizes Educacionais:</p> <ul style="list-style-type: none"> • Set of Guidelines for Teaching OO Modelling Using Concept Mapping Techniques (S37) • minimalCD — A set of translation rules (S22) • Guidelines for conducting training (S40) • Best Practices for Teaching UML (S69) • Classification catalogue for examples (S160) • Patterns of Cognitive Error (S97) • UML-Based Team Project (S118) <p>Abordagens Educacionais:</p> <ul style="list-style-type: none"> • Antipatterns for addressing correctness problems in modeling (S03) • Educational approach based on peer reflection (S138) • Interactive lecture workshop (S140) • Repository of UML Class Diagrams (S02) 	<ul style="list-style-type: none"> • Format of a course (S92, S94, S95 S111 S119, S121) • Elicitation Teaching for Object-Oriented Analysis and Design (S54) • Case Method Approach (S43)
---	--	---	--	--

Tecnologias propostas para minimizar as dificuldades				
Categorias	Dificuldades principais	Ferramentas	Métodos, Diretrizes e Abordagens	Design Instrucional
		<ul style="list-style-type: none"> • Online Img2UML Repository (S25) 		
Dificuldades relacionadas ao diagrama de sequência	<ul style="list-style-type: none"> • Dificuldades relacionadas aos objetos • Dificuldades relacionadas às mensagens 	<p>Ferramentas Case:</p> <ul style="list-style-type: none"> • Papyrus (S08) • UMLet (S86) • Minimally Adequate Synthesizer (MAS) (S156) <p>Sistemas de tutoria de modelagem:</p> <ul style="list-style-type: none"> • StudentUML (S05) • Courseware for UML/SysML modelling (S44) • TextOO (S72) • Online Learning System (S162, S166, S180) 	<p>Métodos Educacionais:</p> <ul style="list-style-type: none"> • Teaching Methods of Object-Oriented Approach in MIS Curriculum (S107) • Rules-based pedagogical method for identifying defects in models (S49) <p>Diretrizes Educacionais:</p> <ul style="list-style-type: none"> • Set of Guidelines for Teaching OO Modelling Using Concept Mapping Techniques (S37) • Methodology for developing Sequence Diagrams (S53) • Design Pattern in software behavior models (S158) • Guidelines for conducting training (S40) • A set of mapping rules between models (S42) • Best Practices for Teaching UML (S69) • Tabular Analysis Method (S85) • UML-Based Team Project (S118) • SaUML (S52) <p>Abordagens Educacionais:</p> <ul style="list-style-type: none"> • Interactive lecture workshop (S140) 	<ul style="list-style-type: none"> • Format of a course (S64, S94, S95 S119, S121), • Elicitation Teaching for Object-Oriented Analysis and Design (S54) • Case Method Approach (S43)
Dificuldades relacionadas ao diagrama de atividades	<ul style="list-style-type: none"> • Dificuldade relacionada aos fluxos • Dificuldades no nó de união 	<p>Ferramentas Case:</p> <ul style="list-style-type: none"> • Papyrus (S08) <p>Sistemas de tutoria de modelagem:</p> <ul style="list-style-type: none"> • StudentUML (S05) • UML-IT (S06, S108) 	<p>Diretrizes Educacionais:</p> <ul style="list-style-type: none"> • Guidelines for conducting training (S40) • UML-Based Team Project (S118) 	<ul style="list-style-type: none"> • Format of a course (S33, S39, S94) • Elicitation Teaching for Object-Oriented Analysis and Design (S54)

Ainda sobre às ferramentas identificadas, observou-se que a maioria delas é construída para fins acadêmicos e não está disponível para o público. Contudo, há ferramentas disponíveis sob uma licença (paga). Observou-se que mesmo as poucas ferramentas disponíveis para uso podem aumentar o desempenho, reduzindo a sobrecarga e facilitando o trabalho de professores e dos estudantes durante o processo de ensino e aprendizado dos diagramas da UML.

Com relação aos métodos de ensino identificados, percebeu-se que alguns métodos abordam a aprendizagem individual e também foi observado que há também um número crescente de métodos e ferramentas que propõem o trabalho colaborativo, fazendo com que os estudantes desenvolvam projetos de maneira colaborativa. Os pesquisadores estabeleceram uma série de vantagens para os estudantes que trabalham de maneira colaborativa para construir conhecimento. No trabalho colaborativo, os estudantes precisam de habilidades sobre como estruturar o processo de solução, explorar caminhos alternativos, desenvolver consciência de seus próprios conhecimentos e habilidades de resolução de problemas e desenvolver e adaptar seus comportamentos de aprendizagem para apoiar uma aprendizagem e solução de problemas mais eficazes (Segedy *et al.*, 2015). Em tais situações, o aprendizado colaborativo pode ajudar os estudantes a superar esses obstáculos e a encontrar métodos eficazes para construir conhecimento e desenvolver soluções de problemas (Emara *et al.*, 2018; Jarvela *et al.*, 2015).

3.4 LIMITAÇÕES DO MSL

Apesar dos cuidados tomados na definição de nosso protocolo deste MSL (Kitchenham e Chartes, 2007) e sua aplicação sistemática, este trabalho também sofre com algumas limitações e ameaças bem conhecidas à validade que serão discutidas a seguir, além de como elas foram abordadas para minimizar seu impacto sobre este MSL.

A classificação dos resultados deste mapeamento. Uma limitação está relacionada aos resultados encontrados, uma vez que a classificação foi realizada por um processo humano e foi baseada em alguns critérios subjetivos. As dificuldades identificadas foram classificadas com base nos procedimentos de codificação (Strauss e Corbin, 1998), que também depende da classificação manual. O uso de tal abordagem já foi feito em outros estudos (ver Steinmacher *et al.*, 2015). Esses procedimentos foram adotados porque o objetivo deste estudo foi identificar e categorizar as dificuldades percebidas por estudantes e profissionais e categorizar os conceitos é um dos principais componentes do processo de codificação. Para reduzir o viés relacionado a essas preocupações, esta revisão envolveu outros dois pesquisadores que verificaram cada

artigo quanto à inclusão e o processo de codificação e um terceiro pesquisador responsável por revisar e discutir as informações geradas após cada etapa.

Os estudos selecionados não representam toda a pesquisa. Outra limitação é ter um conjunto de estudos que não representem toda a pesquisa sobre as dificuldades de estudantes e profissionais na aprendizagem de diagramas UML. Essa ameaça em potencial foi minimizada seguindo uma estratégia de pesquisa que inclui uma pesquisa automática dos estudos selecionados. Além disso, a definição, o refinamento e a pesquisa piloto de maneira iterativa e com a validação de um conjunto bem definido de critérios de inclusão e exclusão contribuíram para reforçar a validade externa do nosso estudo. Aplicou-se métodos e processos sistemáticos bem aceitos em todo o estudo e documentou-se todos eles em nosso protocolo de pesquisa para que este estudo possa ser replicado por outros pesquisadores interessados sobre dificuldades e tecnologias relacionadas à UML. Além disso, sabe-se que outros pesquisadores podem identificar conceitos e atributos diferentes daqueles em nossa estrutura de classificação. Atenuou-se essa ameaça potencial (i) deixando os conceitos e atributos emergirem dos estudos-piloto e refinando-os durante toda a atividade de extração de dados, (ii) realizando uma avaliação externa por pesquisadores independentes que não estavam envolvidos em nossa pesquisa e no processo de extração de dados foi realizado por três pesquisadores. Também evitou-se discutir resultados que podem não estar diretamente relacionados aos dados extraídos.

3.5 CONSIDERAÇÕES SOBRE O CAPÍTULO

Nesse mapeamento sistemático, foram identificadas e organizadas as evidências empíricas disponíveis sobre as dificuldades enfrentadas pelos estudantes ao aprender modelagem de software com o uso de diagramas da UML. As dificuldades evidenciadas na literatura foram agregadas em um único local e, para isso, utilizou-se uma abordagem de codificação inspirada nos procedimentos de codificação aberta e axial do método *Grounded Theory* para realizar a classificação das dificuldades.

Este capítulo apresentou um catálogo estruturado e consolidado de evidências, que raramente é documentado com precisão. Além disso, este trabalho pode fornecer uma base para pesquisas adicionais relacionadas ao ensino de modelagem e servir como uma referência rápida para professores e pesquisadores em diagramas da UML. Além disso, o entendimento das dificuldades pode ajudar os professores a entender como isso afeta a qualidade dos modelos resultantes e que o entendimento das relações entre esses defeitos e dificuldades pode ser benéfico para a prática e a pesquisa. Com relação às tecnologias identificadas, notou-se também

que há um grande número de tecnologias propostas para auxiliar o ensino desses diagramas, além disso, a maioria das tecnologias propostas é validada experimentalmente. Os estudos realizados são em sua maioria realizados com estudantes de graduação e pós-graduação. Isso mostra que o foco dos estudos é do ponto de vista do estudante. Com isso, é necessário o desenvolvimento de novas tecnologias para apoiar os professores no processo de ensino desses diagramas. Com esse tipo de tecnologia, o professor pode ajudar os estudantes de maneira mais proativa no processo de ensino da modelagem de software.

CAPÍTULO 4 – CICLO DE RELEVÂNCIA (PARTE II) - ESTUDOS EXPERIMENTAIS EXPLORATÓRIOS SOBRE O ENSINO DE MODELAGEM DE SOFTWARE COM DIAGRAMAS DA UML

Este capítulo apresenta os resultados do segundo ciclo de relevância. Neste ciclo será apresentado o planejamento, execução e resultados de estudos experimentais realizados com o objetivo de investigar a influência do processo de aprendizagem de modelagem de software (diagrama de casos de uso, especificação de casos de uso, diagrama de atividades, diagrama de classes e diagrama de sequência).

4.1 INTRODUÇÃO

Conforme visto no Capítulo 3, os estudantes apresentam diferentes dificuldades enquanto estão aprendendo a modelagem de software empregando os diagramas da UML. Isso pode estar relacionado a forma como a UML está sendo ensinada pelos professores (Szmurlo e Śmiałek, 2006), uma vez que a abordagem tradicional centrada no professor ainda é dominante em cursos de Computação (Chen *et al.*, 2009). Desta forma, observa-se uma lacuna entre os conceitos abstratos ensinados na sala de aula e as habilidades necessárias para que os estudantes tenham sucesso na indústria (Holmes *et al.*, 2018). De acordo com Al-Tahat (2014) e Chen *et al.* (2019), essas dificuldades estão relacionadas a falta de envolver os estudantes em um ambiente de aprendizagem ativa em sala de aula, tornando-os apenas como agentes passivos durante o processo de ensino-aprendizagem. O currículo de ES não é uma exceção a isso. Portanto, é necessário explorar novas estratégias pedagógicas, que proporcionem aos estudantes um ambiente desafiador, de forma que os estudantes estejam ativamente envolvidos no processo de aprendizagem (De Freitas *et al.*, 2016).

Desta forma, este capítulo apresenta estudos experimentais conduzidos com o objetivo de investigar sobre os como os estudantes compreendem as atividades realizadas pelos professores em sala de aula, durante o processo de ensino de modelagem de software ao adotarem os diagramas da UML. O primeiro estudo teve como objetivo compreender qual a melhor forma de organizar as atividades de modelagem em sala de aula. O segundo visou entender a percepção dos estudantes sobre o uso de duas estratégias que são comumente empregadas pelos professores no ensino de modelagem. Por fim, o terceiro estudo ajudou a

compreender a percepção dos estudantes sobre como um jogo educacional por apoiar no processo de ensino de modelagem.

4.2 ESTUDO EXPERIMENTAL COMPARATIVO SOBRE A MODELAGEM REALIZADA INDIVIDUALMENTE E EM GRUPO

Para auxiliar os estudantes durante as aulas, os professores realizam exercícios individualmente e/ou em grupos para apoiar o processo de ensino e aprendizagem dos diagramas da UML. Chi e Wylie (2014) afirmam que, quando os estudantes são envolvidos em atividades práticas realizadas individualmente ou em grupo, eles tendem a se envolver em processos de aprendizagem específicos e isso pode impactar na construção do conhecimento e nos tipos de habilidades que os estudantes adquirem. Durante a resolução de problemas, atividades realizadas individualmente podem ser mais benéficas do que em grupo (Chi e Wylie, 2014). Trabalhar individualmente possibilita que os estudantes sejam mais práticos, pois não terão que compartilhar tarefas com os outros membros da equipe e não precisarão realizar pausas para explicar as suas ações/decisões (Chi e Wylie, 2014). Por outro lado, as atividades realizadas em grupo possibilitam aos estudantes construir o conhecimento junto com os outros membros da equipe, ajudando e desenvolvendo uma compreensão conceitual mais profunda sobre o conteúdo que está sendo ensinado (Hausmann, 2004). Com isso, espera-se que, ao realizar os trabalhos em grupo, os estudantes tenham maior capacidade para relacionar uma determinada informação e construir esquemas cognitivos de maior qualidade do que ao realizar os trabalhos individualmente (Kirschner, 2009). Portanto, acredita-se que os resultados de aprendizagem alcançados por estudantes que trabalham em equipe sejam melhores do que aqueles alcançados por estudantes que realizam atividades práticas individualmente (Kirschner, 2009). Embora as atividades realizadas em grupo e individualmente sejam utilizadas com frequência em sala de aula (Chi e Wylie, 2014), poucos estudos experimentais foram realizados visando comparar e demonstrar o desempenho dos estudantes e identificar os pontos positivos e negativos das atividades práticas realizadas individualmente e em grupo (Dennis *et al.*, 1994; Dennis *et al.*, 1994. Silva *et al.*, 2018b).

Dada a importância dos diagramas da UML, que são os principais artefatos usados nas diferentes atividades do processo de desenvolvimento de software, essa subseção descreve um estudo experimental realizado com o objetivo de avaliar os efeitos do uso da modelagem em grupo e individual durante a modelagem dos seguintes diagramas: diagrama de casos de uso, diagrama de classes e diagrama de sequência. Esses diagramas foram selecionados por

representar as três visões mais comuns da UML (Ma, 2017; Misbhauddin e Alshayeb, 2017): visão funcional (diagrama de casos de uso); visão estrutural (diagrama de classes) e a visão comportamental (diagrama de sequência). Os estudantes também representaram o comportamento dos casos de uso utilizando a especificação de casos de uso de acordo com o *template* proposto por Cockburn (2000), pois tal especificação auxilia na elaboração do diagrama de sequência (Ma, 2017). Neste estudo, os diagramas elaborados individualmente e em grupo pelos estudantes foram avaliados em termos de corretude e completude. Além disso, com objetivo de obter dados relacionados às dificuldades e facilidades dos estudantes ao realizar atividades de modelagem em grupo e individualmente, foi conduzida uma sessão de Grupo Focal (De França *et al.*, 2015; Kontio *et al.*, 2004). Os dados coletados foram analisados empregando alguns procedimentos de codificação (Strauss e Corbin, 1998). A versão completa do pacote experimental encontra-se disponível no relatório técnico Silva *et al.* (2018).

4.2.1 Planejamento e Execução do Estudo

As atividades que compõem o processo de realização do estudo estão descritas a seguir.

Contexto de Experimento. Este estudo foi conduzido com 18 estudantes voluntários do curso de Sistemas de Informação da Universidade Federal do Amazonas (UFAM). Os estudantes estavam cursando a disciplina de Análise e Projeto de Sistemas, que é ofertada anualmente e tem como objetivo: (a) apresentar os fundamentos sobre os conceitos de análise e projeto orientado a objetos; e (b) projetar diagramas dos sistemas através da UML.

Perfil dos pesquisadores. Quatro pesquisadores (Pesq1, Pesq2, Pesq3 e Pesq 4) estavam envolvidos no experimento. Com relação à experiência em modelagem, o pesquisador Pesq1 possui quatro anos de experiência e os demais pesquisadores possuem mais de dez anos de experiência em ensino de modelagem. Sobre a experiência em análise qualitativa, todos os pesquisadores envolvidos tinham mais de quatro anos de experiência nesse tipo de análise.

Planejamento do experimento. Nesta etapa realizou-se a preparação dos materiais utilizados no experimento: elaboração do termo de consentimento, definição dos cenários e as instruções para o experimento. Todos os artefatos do estudo foram elaborados pelo primeiro autor (Pesq1) e revisados por outros dois pesquisadores (Pesq2 e Pesq3) mais experientes. Durante a revisão, quando os pesquisadores identificavam algum problema, uma nova versão do artefato era elaborada.

Execução do experimento. O experimento foi realizado como uma atividade prática avaliativa, já prevista na ementa da disciplina. No início do experimento, os pesquisadores Pesq1 e Pesq2 atuaram como moderadores, sendo responsáveis por transmitir as informações

sobre as atividades para os estudantes. Quando os moderadores entregaram o termo de consentimento, foi explicado aos estudantes que eles poderiam não disponibilizar seus dados para o estudo, mas deveriam realizar a atividade. Todos os estudantes (18 estudantes) concordaram e assinaram o termo de consentimento. Após isso, os estudantes foram divididos aleatoriamente em dois grupos, que ficaram alocados em salas diferentes. Em cada sala havia um moderador. Considerou-se que o conhecimento em modelagem dos estudantes era homogêneo, pois todos estavam aprendendo modelagem de software com os diagramas e nenhum possuía experiência prévia na indústria. Além disso, ambos os grupos realizaram atividades similares em momentos diferentes do estudo. A Figura 4.1 mostra a composição dos grupos de estudantes e o procedimento seguido no estudo. O estudo foi realizado em três sessões, cada sessão teve duração de duas aulas (1h40m). Todas as sessões do estudo foram realizadas na mesma semana. Nas duas primeiras sessões, os estudantes realizaram a modelagem para cenários específicos e na terceira sessão realizou-se o Grupo Focal.

Na primeira sessão, os estudantes do Grupo 1 (G1) realizaram a modelagem individualmente. Enquanto isso, os estudantes do Grupo 2 (G2) realizaram a modelagem em grupo. Antes de realizar a modelagem com o G2, solicitou-se aos estudantes que formassem novos subgrupos que deveriam ser compostos por, no mínimo, três estudantes. Os estudantes decidiram quais estudantes fariam parte do seu grupo. Na segunda sessão, o processo foi invertido, isto é, os estudantes do G1 realizaram a modelagem em grupo e os estudantes do G2 individualmente. Desta forma, todos os estudantes realizaram as atividades de modelagem individualmente e em grupo. Na terceira sessão, os estudantes participaram de uma sessão de Grupo Focal (De França *et al.*, 2015; Kontio *et al.*, 2004). Grupo Focal é uma técnica de coleta de dados qualitativos realizada por meio de entrevistas organizadas em grupo (De França *et al.*, 2015). O Grupo Focal permite que seja dado um foco em um tópico específico, mediado por um moderador, e que os participantes desenvolvam as suas respostas e discutam a partir das ideias dos outros, enriquecendo as informações obtidas (Kontio *et al.*, 2004).



Figura 4.2. Sessão do Grupo Focal.

Análise dos Resultados. Os pesquisadores analisaram os diagramas e as especificações elaboradas. Nesse estudo, foram utilizados os conceitos de corretude e completude representando as métricas objetivas adotadas para avaliar a qualidade dos diagramas e da especificação elaborados pelos estudantes (Granda *et al.* 2015; Medeiros *et al.*, 2007). Segundo Granda *et al.* (2015), o termo corretude define o quanto um diagrama/especificação emprega corretamente os elementos e os relacionamentos, de acordo com a sintaxe da notação, sendo afetada por defeitos de inconsistência, informação estranha, fato incorreto e ambiguidade. Já a completude define o quanto um diagrama apresenta as informações necessárias de acordo com o propósito da modelagem. A completude está relacionada à semântica, em que o diagrama/especificação representa todos os requisitos do domínio do problema. Com isso, a omissão de informações reduz a completude do artefato (Granda *et al.* 2015). Devido à isso, a corretude e a completude foram calculadas separadamente: corretude para avaliar o quão correto estavam os elementos dos diagramas/ especificações; e completude para avaliar se o diagrama/especificações estava de acordo com o domínio do problema.

A Tabela 4.1 mostra as fórmulas utilizadas para calcular a corretude e a completude dos

diagramas e da especificação, baseada na fórmula proposta por Silva *et al.* (2017a).

Tabela 4.1. Fórmula utilizada para as métricas.

Fórmula
$\text{Corretude} = \frac{EC}{EC + \frac{(1 \times EI_{\text{Leve}}) + (2 \times EI_{\text{Médio}}) + (3 \times EI_{\text{Grave}})}{6}}$
<p>em que, EC: Elementos Corretos (número de elementos modelados corretamente no diagrama/especificação). EI: Elementos Incorretos (número de elementos modelados incorretamente no diagrama/especificação).</p>
$\text{Compleitude} = \frac{IR}{IR + \frac{(1 \times FI_{\text{Leve}}) + (2 \times FI_{\text{Médio}}) + (3 \times FI_{\text{Grave}})}{6}}$
<p>em que, IR: Informações Relevantes (número de informações necessárias e relevantes do cenário que estão presentes no diagrama/ especificação). FI: Falta de Informações (número de informações necessárias e relevantes que não estão presentes no diagrama/especificação).</p>

Para categorizar os defeitos, definiu-se um grau de severidade para cada defeito identificado:

- **Leve** – quando o defeito não prejudica a compreensão e entendimento do diagrama/especificação;
- **Médio** – quando as informações/elementos estão incompletos/ambíguos no diagrama/especificação e/ou uso de elementos de forma incorreta;
- **Grave** – defeitos de notação ou omissão dos elementos/requisitos que comprometem o entendimento do diagrama/especificação.

Dessa forma, a completude e a corretude podem variar de 0,00 (pior resultado possível) a 1,00 (melhor resultado possível). Para avaliar cada diagrama (diagrama de casos de uso, diagrama de classes diagrama e diagrama de sequência) e as especificações utilizou-se itens de verificação, que estão melhor descritos em Silva *et al.* (2018). A avaliação dos diagramas/especificações foi realizada pelo Pesq1, que identificou e classificou os defeitos, e foi revisada e discutida com mais outros dois pesquisadores (Pesq3 e Pesq4).

Realizou-se também uma análise específica sobre os comentários dos estudantes coletados no Grupo Focal. Primeiramente, os comentários dos estudantes foram analisados qualitativamente utilizando procedimentos de codificação (Strauss e Corbin, 1998), em seguida, em vez de definir as categorias e discuti-las, optou-se por fazer uso de algumas citações dos estudantes ao longo do texto como forma de enriquecer algumas das descobertas dos resultados quantitativos. A análise foi conduzida pelo Pesq1 e discutida com um segundo pesquisador

(Pesq4), que possui mais de cinco anos de experiência nesse tipo de análise. Isto foi realizado a fim de mitigar o viés eventualmente causado pela participação de um único pesquisador no processo de codificação.

4.2.2 Resultados do Estudo

Nesta seção são apresentados os resultados quantitativos e qualitativos obtidos nesse estudo.

4.2.2.1 Resultados da Análise da Corretude

A Tabela 4.2 apresenta uma visão geral dos resultados da corretude dos diagramas e da especificação elaborados individualmente e em grupo. Na Tabela 4.2, os participantes P3 e P15 estão sinalizados com "--", pois não puderam comparecer ao segundo dia do estudo.

Tabela 4.2. Valores da corretude dos diagramas e da especificação.

Participante	CORRETUDE							
	Diagrama de Casos de Uso		Especificação Caso de Uso		Diagrama de Classes		Diagrama de Sequência	
	Individual	Grupo	Individual	Grupo	Individual	Grupo	Individual	Grupo
P01	0,97	0,97	0,95	0,95	0,98	1,00	0,00	0,96
P02	1,00	0,97	0,95	0,95	0,98	1,00	0,97	0,96
P03	1,00	--	0,99	--	0,91	--	0,92	--
P04	0,00	1,00	0,97	0,95	0,98	0,99	0,98	0,97
P05	1,00	0,97	0,95	0,95	0,92	1,00	0,92	0,96
P06	0,94	1,00	0,92	1,00	1,00	0,98	0,96	0,94
P07	1,00	0,97	0,98	0,95	1,00	1,00	0,00	0,96
P08	0,99	0,97	0,91	0,97	0,00	0,97	0,94	0,95
P09	0,96	0,97	1,00	0,97	0,97	0,97	0,93	0,95
P10	0,92	1,00	0,97	0,97	1,00	0,96	0,95	0,95
P11	0,96	1,00	1,00	0,95	1,00	0,99	0,00	0,97
P12	0,98	1,00	1,00	1,00	1,00	0,98	0,96	0,94
P13	0,83	1,00	0,94	1,00	0,97	0,98	0,00	0,94
P14	0,00	0,97	1,00	0,97	0,98	0,97	0,95	0,95
P15	0,78	--	1,00	--	0,99	--	0,94	--
P16	1,00	1,00	1,00	0,97	1,00	0,96	0,93	0,95
P17	1,00	1,00	1,00	0,97	1,00	0,96	0,98	0,95
P18	0,83	1,00	1,00	0,95	1,00	0,99	0,88	0,97
Média	0,84	0,99	0,97	0,97	0,93	0,98	0,73	0,95
Mediana.	0,97	1,00	0,99	0,97	0,99	0,98	0,94	0,95
Desvio Padrão	0,31	0,02	0,03	0,02	0,23	0,02	0,40	0,01

Devido ao número reduzido de estudantes em cada grupo, não foi possível realizar o teste de hipótese, pois a quantidade de observações foi muito baixa para a aplicação de métodos estatísticos (Koziolek e Firus, 2005). Então, explorou-se a estatística descritiva e apresentou-se graficamente as principais características dos dados coletados. Na Figura 4.3, é possível observar de forma resumida, os valores da corretude dos diagramas e das especificações elaborados pelos estudantes.

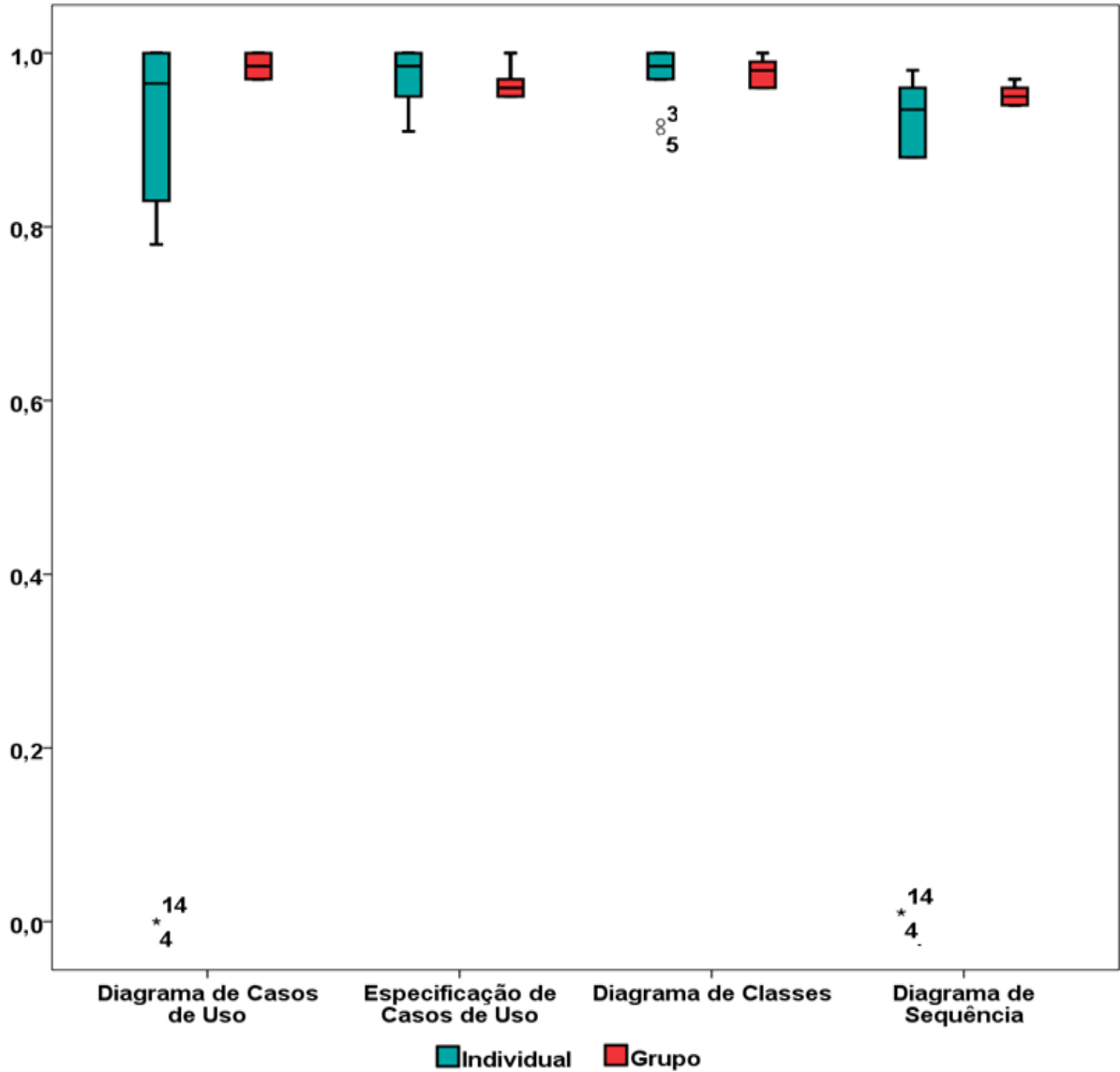


Figura 4.3. Boxplots com os resultados da corretude.

No que diz respeito à corretude dos **diagramas de casos de uso**, os valores das medianas para os diagramas modelados individualmente e em grupo foram próximas, 0,97 e 1,00, respectivamente. Contudo, percebe-se que os valores da corretude dos diagramas de casos de uso elaborados individualmente estão mais dispersos, diferente dos valores da modelagem em grupo. Embora a maioria dos diagramas de casos de uso tenham obtido boa corretude, alguns problemas foram identificados, principalmente no que diz respeito aos diagramas modelados individualmente. Por exemplo, o uso incorreto da direção das setas dos relacionamentos de inclusão (*includes*) e extensão (*extends*), a não descrição do nome do ator e o uso incorreto da direção das setas de especialização e generalização. Com relação ao primeiro problema, este foi relatado pelo estudante P05, enquanto elaborava o diagrama individualmente: “*tive dificuldade no diagrama de casos de uso, porque quando você está sozinho surgem dúvidas (...) em questão*”

de organização, como fazer as setas relacionadas aos includes e extends.” O estudante P10 afirmou que não teve esta dificuldade em grupo, pois o grupo “ajudou a identificar a questão dos includes e extends, a verificar a direção, (...) a gente discute com o grupo sobre isto.”

Sobre a corretude das **especificações de caso de uso**, o valor da mediana da corretude das especificações geradas em grupo e individualmente são bem próximos, 0,99 para as realizadas individualmente e 0,97 em grupo. Além disso, os valores das corretude em ambos os casos estão concentrados ao redor da mediana. A especificação de casos de uso possui poucas regras que estão relacionadas à sua sintaxe. Acredita-se que, por esse motivo, foram identificados poucos defeitos sintáticos: inclusão incorreta das dependências de outros casos de uso na especificação e pré-condições definidas incorretamente. Por exemplo, o estudante P13 disse que ao especificar sozinho sentiu *“dificuldade em determinar as pré-condições do caso de uso e o fluxo principal.”* Além disso, esse estudante comentou: *“eu tenho bastante dificuldade (sozinho) em lembrar os nomes dos campos da especificação de casos de uso, o objetivo e o fluxo principal”*.

Os valores da mediana da corretude dos **diagramas de classes** modelados em grupo e individualmente estão concentrados ao redor da mediana. Isto ocorreu mesmo tendo dois estudantes (P03 e P05) que obtiveram valores abaixo da maioria (0,91 e 0,92, respectivamente) ao realizar a atividade individualmente. Um dos defeitos mais comum é que alguns atributos estavam alocados em classes erradas. Outro defeito identificado foi a utilização incorreta da multiplicidade no diagrama de classes. Com relação à sintaxe desse diagrama, os estudantes não relataram dificuldades ou facilidades ao modelar este diagrama em grupo ou individualmente.

Por fim, no que diz respeito à corretude dos **diagramas de sequência**, é possível perceber na Figura 4.3 que, apesar dos valores da mediana estarem próximos, os diagramas de sequência modelados individualmente encontram-se mais dispersos. Isto ocorreu pois, na maioria dos diagramas, os estudantes não modelaram de forma correta a interação entre objetos – que é realizada através do controlador. Este tipo de defeito também ocorreu quando os estudantes realizaram a modelagem em grupo. Percebeu-se ainda que nos diagramas modelados individualmente e em grupo, alguns dos objetos estavam com nomes incorretos. Isto pode ter sido ocasionado devido ao descuido dos estudantes durante a modelagem.

4.2.2.2 Resultados da Análise da Completude

A Tabela 4.3 apresenta uma visão geral dos resultados da completude dos diagramas e da especificação que foram elaborados individualmente e em grupo. Na Tabela 4.3, os

participantes P3 e P15 estão sinalizados com "--", pois não puderam comparecer ao segundo dia do estudo.

Tabela 4.3. Valores da completude dos diagramas e da especificação.

Participante	COMPLETUDE							
	Diagrama de Casos de Uso		Especificação Caso de Uso		Diagrama de Classes		Diagrama de Sequência	
	Individual	Grupo	Individual	Grupo	Individual	Grupo	Individual	Grupo
P01	1,00	0,89	0,99	0,97	0,96	0,71	0,00	0,68
P02	1,00	0,89	0,96	0,97	0,86	0,71	0,82	0,68
P03	0,89	--	0,96	--	0,46	--	0,32	--
P04	0,00	1,00	0,81	0,90	0,80	0,77	0,73	0,65
P05	0,86	0,89	0,95	0,97	0,62	0,71	0,51	0,68
P06	0,96	0,93	0,92	0,88	0,94	0,69	0,89	0,58
P07	1,00	0,89	0,83	0,97	0,67	0,71	0,45	0,68
P08	1,00	0,94	0,93	0,98	0,67	0,83	0,56	0,66
P09	0,86	0,94	0,77	0,98	0,80	0,83	0,71	0,66
P10	0,89	1,00	0,89	0,96	0,67	0,73	0,54	0,51
P11	0,92	1,00	0,91	0,90	0,80	0,77	0,00	0,65
P12	1,00	0,93	0,88	0,88	0,62	0,69	0,61	0,58
P13	0,98	0,93	0,95	0,88	0,82	0,69	0,76	0,58
P14	0,00	0,94	0,94	0,98	0,78	0,83	0,68	0,66
P15	0,92	--	0,93	--	0,72	--	0,65	--
P16	0,89	1,00	0,98	0,96	0,89	0,73	0,79	0,51
P17	1,00	1,00	0,83	0,96	0,78	0,73	0,55	0,51
P18	0,93	1,00	0,99	0,90	0,86	0,77	0,78	0,65
Média	0,84	0,93	0,91	0,94	0,76	0,74	0,58	0,62
Mediana.	0,93	0,94	0,93	0,96	0,79	0,73	0,63	0,65
Desvio Padrão	0,31	0,07	0,06	0,04	0,13	0,05	0,25	0,06

Na Figura 4.4 é possível observar, de forma resumida, os valores da completude dos diagramas e das especificações elaborados pelos estudantes. Com relação à completude do **diagrama de casos de uso**, os valores das medianas dos diagramas gerados individualmente e em grupo foram bem próximas, 0,93 e 0,94, respectivamente. Vale ressaltar ainda que dois estudantes (P04 e P14) não realizaram a modelagem desse diagrama individualmente. Além disso, dois estudantes (P05 e P09) obtiveram desempenho menor ao realizar a modelagem individualmente (0,86). Em contrapartida, seis estudantes (P01, P02, P07, P08, P12 e P17) tiveram o melhor desempenho (1,00). No que diz respeito aos diagramas elaborados em grupo, os subgrupos SG2_1 (P10, P16 e P17) e SG2_3 (P04, P11 e P18) obtiveram o melhor desempenho (1,00) e o subgrupo SG1_2 (P01, P02, P05 e P07) obteve o pior desempenho (0,89). Sobre os defeitos cometidos pelos estudantes, percebeu-se que os estudantes tiveram dificuldades em identificar os casos de uso e em determinar o nível correto de granularidade dos casos de uso. Isso pode ter ocorrido, pois quando estão com dúvidas, os estudantes tendem a derivar alguns casos de uso em casos de uso menores. Isso afeta na perda da ideia das principais

funcionalidades do sistema e das relações entre elas. Além disso, percebeu-se que houve defeitos relacionados à identificação e às relações entre casos de uso.

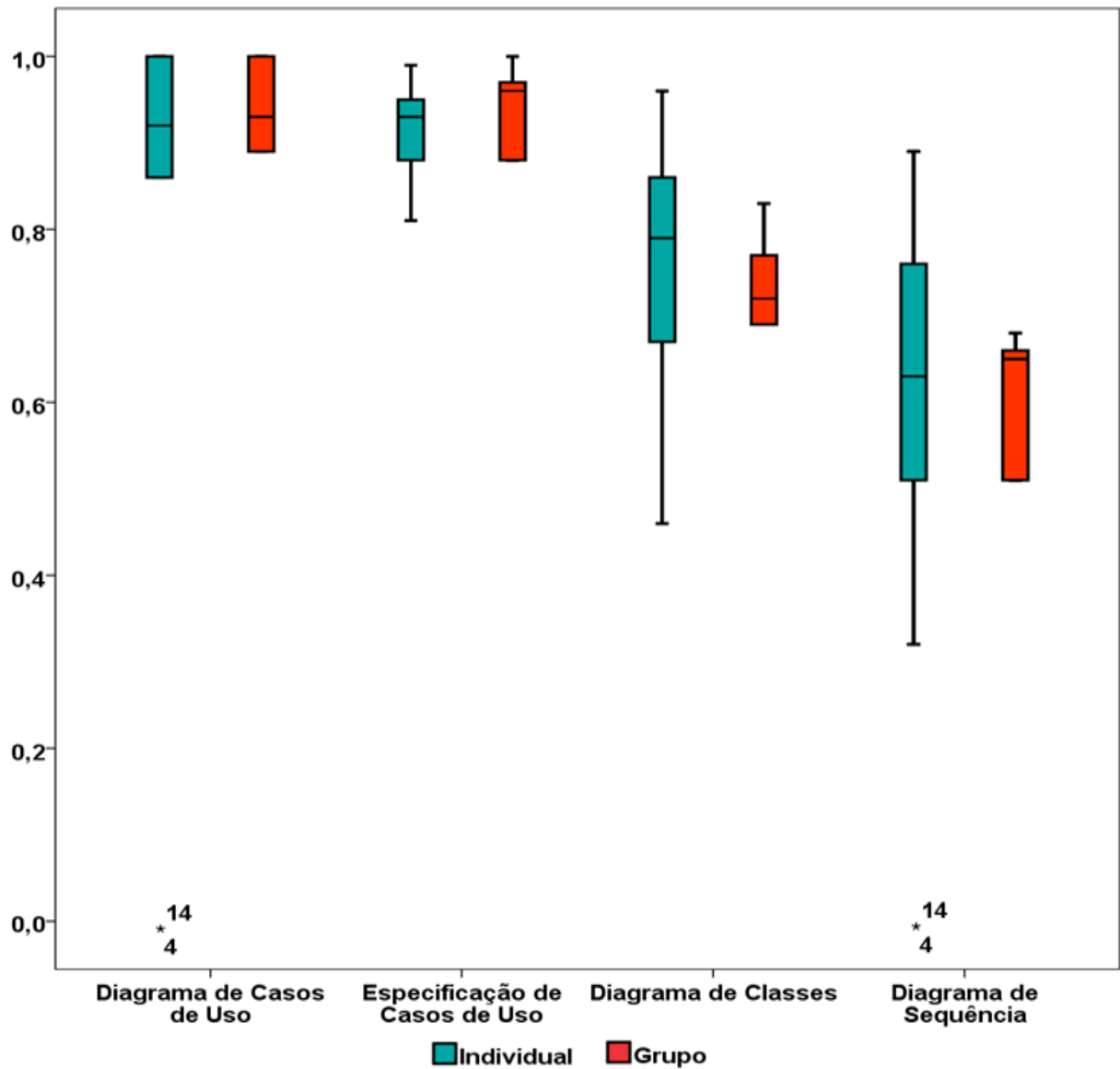


Figura 4.4. Boxplots com os resultados da completude.

Quanto à preferência de realizar a modelagem do diagrama de casos de uso individualmente, o estudante P04 comentou: “*modelar o diagrama de caso de uso individualmente foi bom porque não gastei tempo tentando explicar minha compreensão do problema para a equipe, assim consegui modelar com mais detalhes.*” O estudante P09 concordou com este ponto, pois se torna “*mais fácil modelar o diagrama de casos de uso (individualmente), já que se segue uma única linha de raciocínio.*” Além disso, o estudante P12 ressaltou que: “*é um diagrama (casos de uso) muito importante porque vai basear todos os outros e se muitas pessoas ficarem discutindo acabam entrando em conflito.*” Em relação à

modelagem do diagrama de casos de uso em grupo, o estudante P05 relatou que é *“mais fácil fazer em grupo (...) pelo fato da pessoa conseguir ter mais ideias e também discutir se aquele caso de uso que se está querendo colocar é ou não útil no diagrama.”* O estudante P02 complementou dizendo que: *“é mais fácil trabalhar o diagrama de casos de uso em grupo (...) com uma equipe coesa que tem a capacidade de trabalhar em grupo, que consegue ouvir a opinião do outro e que sabe ceder quando é necessário, no final, a gente vai chegar um diagrama de melhor qualidade (...) essas visões diferentes, essas leituras diferentes acabam contribuindo.”* A partir destes resultados, percebe-se que não houve diferença com relação à completude ao modelar o diagrama de casos de uso em grupo e individualmente. Contudo, alguns estudantes ressaltaram que quando a modelagem foi realizada em grupo, eles podiam discutir alguns pontos que poderiam ou não ser representados no diagrama de casos de uso. Como já previsto, o trabalho realizado em grupo possibilitou que os estudantes melhorassem as suas habilidades de discussão e de trabalho em equipe.

No que diz respeito à completude das **especificações de caso de uso**, o valor da mediana para as especificações elaboradas individualmente foi de 0,93 e em grupo foi de 0,96. No entanto, percebe-se uma maior distribuição nos valores da completude das especificações elaboradas individualmente, quando comparado com os valores das elaboradas em grupo. Alguns dos defeitos cometidos pelos estudantes se relacionam à omissão de regras de negócio e omissão de algum fluxo de evento (alternativo e/ou exceção) necessário à completude do caso de uso. Apesar desses defeitos, na modelagem individual, o valor mínimo da corretude das especificações foi 0,77 (P09) e o valor máximo foi 0,99 (P01 e P18). Enquanto isso, na modelagem em grupo, o valor mínimo foi 0,88 (SG1_1 = P06, P12 e P13) e o valor máximo foi 0,98 (SG2_2 = P08, P09 e P14).

Quanto à percepção dos estudantes sobre a modelagem individual, o estudante P13 comentou o seguinte: *“é mais fácil fazer sozinho a especificação de casos de uso, porque cada um já sabe o fluxo, a sua ideia.”* O estudante P03 concordou dizendo que *“a pessoa fica mais concentrada quando ela vai fazer a especificação de casos de uso.”* Contudo, alguns estudantes comentaram que preferiram modelar em grupo: *“quando fiz em grupo a especificação de casos de uso, eu senti mais facilidade, porque se falta algum certo conhecimento no fluxo alternativo, por exemplo, outro colega pode ir lá e ajudar a fornecer esse conhecimento”* (P12). O subgrupo SG1_1, composto pelos estudantes P06, P12 e P13, obteve o menor desempenho ao especificar o caso de uso em grupo. Sobre isso, o estudante P13 comentou que teve essa dificuldade devido à *“organização, pois cada um tem uma forma de especificar os casos de uso.”* Do ponto de vista

quantitativo, percebeu-se que os estudantes realizaram a especificação de forma mais completa em grupo, uma vez que os valores estão mais agrupados ao redor da mediana. Isto pode ter ocorrido pois, em grupo, os estudantes realizaram uma revisão do que se estava especificando, conforme relatado pelo estudante P02: *“fazer em grupo à especificação foi melhor no sentido de revisão, por esse ponto o trabalho em grupo foi mais produtivo.”*

Quanto à completude do **diagrama de classes**, os valores da completude dos diagramas elaborados em grupo estavam mais concentrados ao redor da mediana, diferentemente da distribuição observada para os diagramas gerados individualmente. Apesar do valor da mediana dos diagramas modelados individualmente (0,79) estar maior que a do grupo (0,73), percebeu-se que teve estudantes que obtiveram bons resultados e outros não tão bons. Por exemplo, ao modelar individualmente, o estudante P01 obteve o melhor desempenho (0,96), seguido pelo estudante P06 (0,94), enquanto que o estudante P03 foi o que teve o pior desempenho (0,46). No que diz respeito aos defeitos cometidos pelos estudantes nesse diagrama, o principal foi a omissão de classes importantes para o domínio do problema e o uso incorreto das classes associativas. Os estudantes tinham dificuldades em entender que quando há uma classe associativa, essa classe deve possuir um atributo que seja uma coleção de objetos resultantes de outras duas classes.

Um ponto interessante identificado nesse resultado é que, mesmo tendo obtido resultados melhores quando trabalharam individualmente, os estudantes preferiram modelar o diagrama em grupo. Em relação à isso, o estudante P03, que teve o pior desempenho individual na modelagem desse diagrama, comentou: *“o diagrama de classes tem que pensar como uma implementação do sistema e em grupo seria melhor de trabalhar para sua construção, pois há dúvidas com relação à comunicação das classes que seria melhor decidir com o grupo.”* Contudo, o estudante P13 disse que *“o diagrama de classes realizado individualmente foi mais fácil por eu seguir um foco”*, no entanto, ele completou dizendo que *“isso pode ser um pouco prejudicial, então é melhor depois que fazer individualmente submeter à apreciação do grupo para melhorar.”* O estudante P01 confirmou dizendo que *“uma pessoa sozinha fazendo um diagrama de classes, às vezes, pode esquecer algum detalhe, alguma coisa mais importante, um método ou um atributo ou pode até criar classes que não existem. Então várias pessoas podem acabar ajudando.”* O estudante P14 disse que modelar em grupo ajuda a *“(…) pensar no problema, uma vez que poder dividir o problema em equipe é muito mais simples e leva a um resultado muito mais rápido.”* O estudante P02 complementou dizendo que *“diferentes analistas têm visões diferentes visões do domínio do problema. Isto acaba enriquecendo o diagrama que a gente produz.”*

Por fim, na completude do **diagrama de sequência**, o valor da mediana (0,63) dos diagramas elaborados individualmente foi abaixo da mediana dos elaborados em grupo (0,65). Contudo, percebeu-se que ocorreu a mesma dispersão apresentada no diagrama de classes, ou seja, houve uma grande quantidade de estudantes que tiveram um bom desempenho modelando individualmente e outros estudantes que tiveram um mau desempenho. Os estudantes P06 e P02 obtiveram o melhor desempenho (0,82). Já os estudantes P03 e P07 obtiveram os piores desempenhos (0,32 e 0,45, respectivamente) modelando individualmente. Com relação à facilidade de realizar a modelagem individualmente, o estudante P02 relatou que *“é mais fácil você manter o foco na sua linha de raciocínio quando você está trabalhando sozinho no diagrama de sequência. Como é uma sequência, (...) às vezes uma interferência, uma opinião de outra pessoa pode acabar te desviando da sua linha de raciocínio e fazendo você cometer algum tipo de engano no diagrama.”* O estudante P09 complementou dizendo que *“o diagrama de sequência é uma parte mais mecânica na modelagem, (...) seria uma coisa mais simples de você fazer individualmente, porque não requer você organizar as ideias como você está fazendo nos diagramas anteriores.”* Com relação à modelagem em grupo, o subgrupo SG1_2 (P01, P02, P05 e P07) realizou o diagrama de sequência de forma mais completa. Já o subgrupo SG1_1 (P06, P12 e P14) foi o que obteve o pior desempenho. O estudante P10 disse que teve dificuldade em realizar a modelagem desse diagrama em grupo: *“a dificuldade de fazer o diagrama de sequência (...) foi que precisava primeiro finalizar a descrição dos casos de uso. Então, ficou assim muito dependente, pois tinha que está muito alinhado com o outro que estava descrevendo o caso de uso.”* Ou seja, os estudantes dividiram o trabalho entre os membros da equipe para tentar tornar a modelagem mais rápida, contudo, essa decisão acabou gerando uma dependência de atividade entre os membros da equipe.

Vale a pena ressaltar que a completude era calculada de forma independente da corretude, ou seja, para cada diagrama, os pesquisadores avaliaram se o mesmo representava completamente o domínio do problema (completude) e se estava de acordo com as regras sintáticas da linguagem (corretude). Baseado nisso, os resultados mostraram que os estudantes tiveram problemas ao elaborar completamente os diagramas de sequência em grupo e individualmente. Isso pode estar relacionado à incompletude dos diagramas de classes previamente elaborados, pois os estudantes não conseguiram representar o domínio do problema completamente através desse diagrama. Esse problema refletiu diretamente na incompletude dos diagramas de sequência. Isso também pode ter sido ocasionado pelo fato de os estudantes não compreenderem que os diagramas da UML, como diagramas de sequência e diagrama de classes,

estão relacionados entre si, embora cada um deles aborde um aspecto particular do mesmo sistema. Sobre esse ponto, o estudante P03, que obteve o pior desempenho ao realizar o diagrama de classes e o diagrama de sequência sozinho, relatou que *“quando estava construindo o diagrama de sequência sozinho, às vezes via que alguma coisa estava faltando no diagrama de classes, (então) voltava para o diagrama de classes e corrigia.”* Na modelagem em grupo, todos os subgrupos realizaram a modelagem do diagrama de sequência e, apesar de terem obtidos baixos resultados de completude, alguns estudantes perceberam que precisavam manter uma rastreabilidade entre os diagramas. Sobre isso, o estudante P10 comentou o seguinte: *“o grupo tinha uma noção do que ia ter no diagrama de classes e a gente fez o diagrama de classes daquele jeito (uma primeira versão). Depois fomos fazer o diagrama de sequência, aí no diagrama de sequência a gente acabou vendo que precisava fazer algumas coisas no diagrama de classes e acabava melhorando-o.”*

4.2.3 Discussão dos Resultados

Baseado nos resultados relacionados à corretude, percebeu-se que os estudantes elaboraram a especificação de casos de uso e o diagrama de classes de forma mais correta individualmente, enquanto que os diagramas de casos de uso e de sequência foram modelados de maneira mais correta em grupo. Contudo, os estudantes apresentaram mais dificuldades em entender a sintaxe modelando individualmente que em grupo. Este já era um resultado esperado, pois os estudantes não podiam tirar as suas dúvidas com outros colegas. Porém, o que se esperava também é que os estudantes elaborassem diagramas mais corretos ao realizar a modelagem em grupo, uma vez que houve um alto nível de participação dos estudantes e as várias pessoas observando a notação poderiam ajudar a melhorar a qualidade sintática dos diagramas, o que, de fato, não ocorreu. Com isso, parte dos resultados desse estudo são semelhantes aos de Dennis *et al.* (1999; 1994). No estudo realizado por Dennis *et al.* (1999; 1994), os resultados mostraram que os diagramas que são construídos por um modelador têm melhor qualidade sintática do que os que são construídos em grupo.

Em relação à completude, percebeu-se que os estudantes elaboraram o diagrama e especificação de caso de uso e o diagrama de sequência de forma mais completa em grupo. No entanto, os mesmos não foram tão completos quanto se esperava. Esse não era o resultado esperado pois, os estudantes podiam discutir com os colegas sobre o problema e sugerir melhorias. Os resultados apenas mostraram que a completude dos diagramas e da especificação elaborados em grupo não foi tão dispersa quanto aos dos realizados individualmente. Os resultados mostraram também que os estudantes modelaram o diagrama de classes de forma

mais completa individualmente. Baseando-se na literatura, com relação à completude, os resultados encontrados nesse estudo são diferentes dos resultados encontrados por Dennis *et al.* (1999; 1994). Dennis *et al.* (1999; 1994) não encontraram diferença significativa entre a completude dos diagramas elaborados em grupo dos construídos individualmente.

Os resultados deste estudo também mostraram que alguns estudantes se destacaram ao realizar a modelagem individualmente, por exemplo, os estudantes P01, P02, P08 e P16. Com isso, percebeu-se que ao realizar a modelagem em grupo, alguns dos membros das equipes, das quais estes estudantes faziam parte, que tinham obtidos resultados inferiores individualmente, conseguiram obter melhores resultados ao realizarem em grupo. Do ponto de vista pedagógico, este é um ponto que merece destaque, pois permite que os estudantes que apresentem dificuldades em modelagem possam aprender com outros estudantes mais experientes, além de melhorar a qualidade da colaboração, incentivando a comunicação e a contribuição ativa dos estudantes.

Portanto, respondendo à pergunta sugerida no título deste estudo “*Quais são as diferenças entre modelar em grupo e individualmente durante a aprendizagem de UML?*”, os resultados deste estudo trazem indícios de que nem sempre as atividades práticas de modelagem em grupo fazem com que os estudantes se envolvam em processos eficazes de aprendizagem e obtenham os melhores resultados. Às vezes, ao realizar atividades de modelagem individuais é possível obter resultados de aprendizagem melhores. Esses resultados são corroborados pelos resultados obtidos por Kirschner *et al.* (2009). Segundo os autores, mesmo os estudantes estando engajados em um ambiente de aprendizagem em que os grupos são efetivamente formados, com processos de aprendizagem cognitivos apoiados com sucesso, não há garantia da melhoria do aprendizado de todos os estudantes Kirschner *et al.* (2009).

4.2.4 Considerações sobre o estudo

Esta subseção apresentou um estudo experimental que avaliou os efeitos de realizar atividades práticas de modelagem em grupo e individualmente. No que diz respeito à corretude dos diagramas e da especificação, os resultados mostraram que, dependendo do diagrama que se está ensinando, em alguns casos a modelagem realizada em grupo possibilitou que os estudantes elaborassem diagramas mais corretos (diagrama de casos de uso e diagrama de sequência), enquanto que em outros a modelagem individual obteve resultados melhores (especificação de casos de uso e diagrama de classes). Quanto à completude, percebeu-se que os estudantes, ao realizar a atividade em grupo, elaboraram o diagrama e a especificação de casos de uso e o

diagrama de sequência de forma mais completa. Contudo, modelar o diagrama de classes individualmente foi melhor que em grupo. Percebeu-se também que devido aos estudantes não conseguirem modelar os diagramas de classes de forma completa, isto refletiu diretamente na completude dos diagramas de sequência, uma vez que este diagrama depende diretamente da modelagem completa do diagrama de classes. Estas dificuldades foram percebidas tanto nas modelagens realizadas em grupo e individualmente.

Embora a prática de exercícios de modelagem em grupo, na maioria das vezes, seja relatada na literatura como mais eficaz que individualmente, os resultados deste estudo indicam que o oposto, às vezes, é verdadeiro. Estes resultados sugerem que se o objetivo do professor for que os estudantes aprendam a projetar os diagramas mais corretos e completos, nem sempre a prática de modelagem em grupo obterá resultados de aprendizagem mais positivos. Em vez disso, em alguns casos, é provável que a modelagem realizada individualmente forneça resultados melhores, conforme demonstram alguns dos resultados deste estudo. Contudo, a percepção dos estudantes em relação à modelagem em grupo foi melhor, pois os estudantes se envolviam mais durante a discussão, permitindo melhorar a qualidade dos diagramas elaborados.

Apesar da ampla discussão, os resultados do estudo inferem que as atividades elaboradas em grupo ainda apoiam melhor a realização de algumas atividades de modelagem de software. Sendo assim, utilizou-se como base esses resultados para realizar o planejamento e execução do novo estudo experimental. Esse estudo será realizado para comparar duas estratégias de ensino e obter um *feedback* dos estudantes sobre essas estratégias.

4.3 ESTUDO EXPERIMENTAL COMPARATIVO SOBRE *LEARNING FROM ERRONEOUS EXAMPLES* E *PROBLEM-BASED LEARNING*

Em sala de aula, os professores desejam aplicar novas estratégias de ensino, que proporcionem aos estudantes um ambiente desafiador, de forma que os estudantes estejam ativamente envolvidos no processo de aprendizagem (de Freitas *et al.*, 2016), ou seja, estratégias de aprendizagem ativa. Dentre os métodos implementados com aprendizagem ativa, pode-se citar incluir *Problem-Based Learning* (PBL) (Prince, 2004; Boud e Feleti, 1997) e *Learning from Erroneous Examples* (ErrEx) (Kopp, 2008; Bık *et al.*, 2013). O PBL é um método pedagógico que enfatiza o papel de um problema, no qual os alunos são responsáveis por sua aprendizagem (Boud e Feletti, 1997). Este método ajuda os alunos a desenvolver estratégias e construir conhecimento. A formação de equipes e a distribuição de papéis são essenciais para o uso de PBL (Richardson e Delaney). No ErrEx, os alunos tentam resolver um

problema no qual um ou mais exemplos estão incorretos. Este método ajuda principalmente na aquisição inicial de habilidades no processo de aprendizagem (Kopp, 2008; Båk *et al.*, 2013). Além disso, o ErrEx ajuda os alunos a (a) estudar mais cuidadosamente os exemplos e (b) a lembrar e evitar erros encontrados e aprendidos em uma atividade futura (Kopp, 2008; Båk *et al.*, 2013).

Desta forma, esta subseção apresenta um estudo experimental com o objetivo de avaliar os efeitos do uso dos métodos de ensino: uma abordagem baseada em princípios de *Problem Based Learning* (PBL) e *Erroneous Examples Learning* (ErrEx). Vale ressaltar que para fins desse estudo não foi empregado o PBL por completo. Foram utilizados apenas alguns dos principais passos. Esses métodos foram selecionados pois: (a) fornecem contexto para recuperação subsequente e uso apropriado de novas informações para os e (Prince, 2004; Boud e Feleti, 1997); (b) são métodos eficazes de aprendizagem (Renkl, 2014; Richardson, 2009); e (c) os estudantes participam ativamente do processo de aprendizagem (Kopp, 2008; Båk *et al.*, 2013). Os métodos foram avaliados em termos de corretude e completude dos diagramas criados. Ao medir estes indicadores, espera-se entender como os métodos auxiliaram os estudantes na modelagem de diagramas mais corretos e completos. Além disso, os estudantes forneceram feedback sobre o aprendizado percebido sobre os métodos. A análise dos dados sobre a percepção dos estudantes foi feita qualitativamente empregando alguns procedimentos de codificação (Strauss e Corbin, 1998), para extrair os benefícios e dificuldades percebidos pelos estudantes após o uso dos métodos.

4.3.1 Planejamento e Execução do Estudo

As atividades que compõem o processo de realização do estudo estão descritas nas subseções a seguir.

Contexto do Estudo. Este estudo foi conduzido com 44 estudantes do curso de Ciência da Computação na Universidade Federal do Amazonas (UFAM) no Brasil. Neste estudo, optou-se por trabalhar com esses estudantes porque são os futuros engenheiros de software. Acredita-se que, aplicando os métodos apropriados durante seu processo de aprendizagem, a indústria pode se beneficiar de melhores profissionais. Os estudantes estavam cursando a disciplina de Análise e Projeto de Sistemas, que é ofertada semestralmente e tem como objetivo: (a) apresentar os fundamentos de modelagem e projeto de sistemas e os conceitos de análise e projeto orientado a objetos; e (b) exercitar os conceitos usando a UML. Para isso, a ementa da disciplina é dividida em três partes: (1) Fundamentos de análise e processo de software,

gerência do processo de desenvolvimento de software; (2) processo de identificação de componentes e artefatos de um sistema, elaboração de modelos de sistemas através da UML, projeto de sistemas com base no modelo de especificação; e (3) modelagem de processos de negócios, componentes, arquitetura, linhas de produto de software.

Planejamento do estudo. Nesta etapa, realizou-se a preparação dos materiais que foram utilizados no estudo: elaboração do termo de consentimento, definição dos cenários, as instruções para o estudo, elaboração dos questionários pós-modelagem e questionários pós-estudo. Todos os artefatos do estudo foram validados por outros dois pesquisadores.

Execução do estudo. Inicialmente, os estudantes foram divididos em equipes definidas por conveniência, em que eles decidiram quais estudantes fariam parte do seu grupo. As atividades foram realizadas em grupo visando o engajamento entre os estudantes e uma colaboração ativa dentro de seus grupos. Ao todo, foram formados nove grupos: oito grupos de cinco estudantes e um grupo de quatro estudantes (ver Tabela 4.4). Em seguida, todos os estudantes matriculados (44 estudantes) assinaram o termo de consentimento cedendo os dados de seus trabalhos para análise. Vale ressaltar que os estudantes não possuíam experiência prévia em modelagem.

Tabela 4.4. Distribuição dos estudantes por grupo.

Grupos	Estudantes
G1	S19; S20; S23; S39; S41;
G2	S26; S32; S33; S35; S36;
G3	S12; S16; S30; S38; S42;
G4	S34; S11; S08; S10. S24;
G5	S28; S27; S15; S25; S31;
G6	S01; S37; S29; S21; S02
G7	S40; S04; S07; S05;
G8	S14; S06; S22; S03; S09;
G9	S17; S13; S43; S18; S44;

A Figura 4.5 mostra o procedimento seguido no estudo. Antes de cada sessão, os estudantes assistiam a uma aula teórica sobre os diagramas que iriam modelar. As aulas enfatizavam os conceitos que seriam necessários para que os estudantes entendessem os objetivos e a notação do diagrama ensinado. Além disso, foram realizados exercícios com o objetivo de analisar se os estudantes tinham entendido os que fora apresentado na aula teórica. Cada sessão teve duração de duas aulas (duração de 1h40 por aula). Na primeira sessão, os estudantes realizaram a modelagem do diagrama de atividades e na segunda sessão realizaram a modelagem do diagrama de sequência.

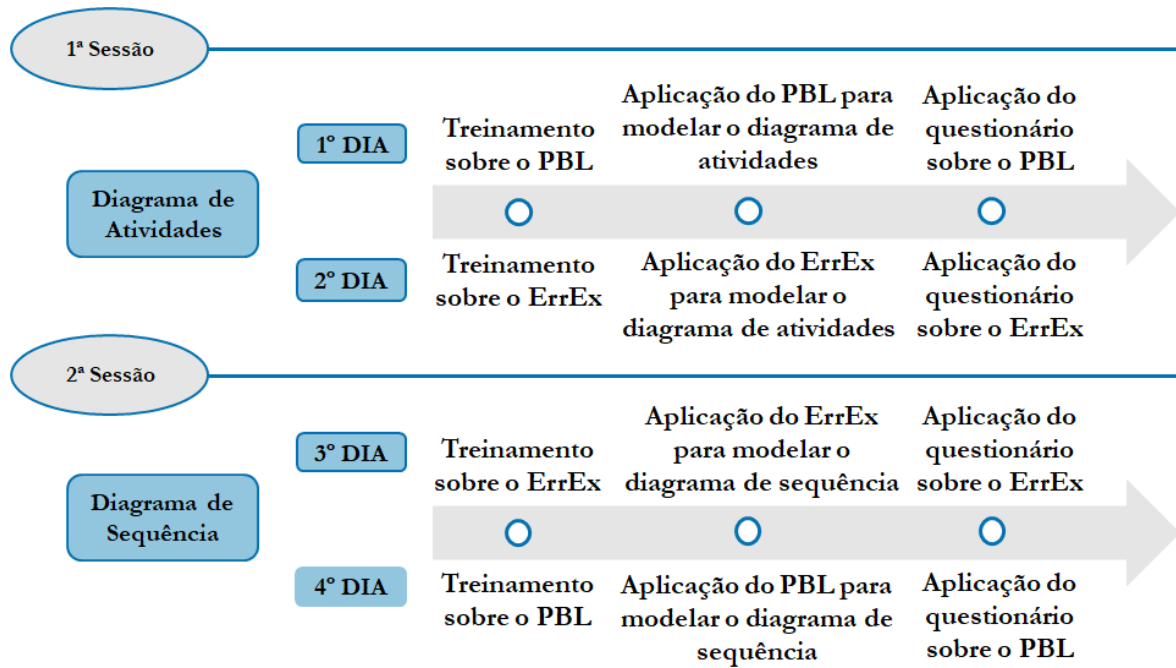


Figura 4.5. Procedimento seguido no estudo.

No primeiro dia do estudo, os grupos receberam um treinamento sobre como o método baseado em princípios de PBL deveria ser utilizado. Após isso, cada grupo recebeu um cenário com um problema descrito e, aplicando o método PBL, os estudantes deveriam modelar o diagrama de atividades. Ao finalizar a modelagem, os estudantes responderiam um questionário que tinha como objetivo verificar a percepção dos estudantes sobre o PBL.

No segundo dia, o mesmo procedimento foi realizado: (a) treinamento com os estudantes sobre como aplicar o método ErrEx; (b) entrega de um cenário com um problema descrito e um diagrama de atividades com vários defeitos sintáticos e semânticos para os estudantes modelarem um novo diagrama de atividades; e (c) aplicação de um questionário sobre as percepções dos estudantes sobre o ErrEx. Na segunda sessão, os métodos foram invertidos e foram utilizados para a modelagem do diagrama de sequência. Neste caso, os estudantes utilizaram primeiro o ErrEx para auxiliar na modelagem do diagrama de sequência e, em seguida, utilizaram o PBL.

Análise dos Resultados do estudo. Por fim, nesta etapa foram analisados os resultados obtidos no estudo. Nesta etapa, os pesquisadores analisaram os diagramas modelados e as respostas dos questionários. Para a análise dos diagramas, foram utilizadas métricas objetivas sobre a qualidade dos diagramas elaborados pelos estudantes: (a) corretude e (b) completude. Estas métricas foram as mesmas definidas na Subseção 4.2.1. A avaliação dos diagramas foi

realizada por um pesquisador e, posteriormente, conferida e discutida por outros dois pesquisadores mais experientes em modelagem de software.

4.3.2 Resultados do Estudo Experimental

Nesta seção são apresentados os resultados quantitativos e qualitativos obtidos no estudo experimental.

4.3.2.1 Resultados da Análise da Corretude e Completude dos diagramas elaborados

A Tabela 4.5 apresenta uma visão geral dos resultados da corretude e completude dos diagramas elaborados a partir do uso dos dois métodos (PBL e ErrEx). A completude e a corretude de um diagrama podem variar de 0,00 (pior resultado possível) a 1,00 (melhor resultado possível). Vale ressaltar que durante a aplicação do método ErrEx, os grupos G7 e G8 não puderam comparecer no dia do estudo, por isto, na tabela estão marcados como "--". Um resumo dos resultados é apresentado usando o gráfico de *boxplot*.

Tabela 4.5. Resumo do Resultado da Avaliação dos Diagramas por Grupo.

Grupos	PBL				ErrEx			
	Corretude		Completude		Corretude		Completude	
	DA	DS	DA	DS	DA	DS	DA	DS
G1	1,00	0,97	0,95	0,84	0,87	1,00	1,00	0,30
G2	1,00	1,00	0,86	0,95	0,96	0,97	1,00	0,83
G3	0,87	1,00	0,88	0,80	0,96	0,71	0,93	0,73
G4	0,84	0,97	0,93	0,87	0,95	0,94	0,91	0,81
G5	0,98	1,00	0,96	0,90	0,96	0,98	1,00	0,90
G6	0,80	1,00	0,98	0,87	0,86	1,00	0,90	0,83
G7	0,84	0,73	0,87	0,86	0,88	--	1,00	--
G8	0,97	1,00	0,82	0,58	0,88	0,93	1,00	0,50
G9	0,98	1,00	0,77	0,30	--	0,80	--	0,60
M	0,92	0,96	0,89	0,77	0,91	0,90	0,96	0,69
Md.	0,97	1,00	0,88	0,86	0,91	0,90	1,00	0,77
DP	0,08	0,08	0,07	0,20	0,05	0,05	0,05	0,20

Legenda: G – Grupos; DA – Diagrama de Atividades; DS – Diagrama de Sequência; M – Média; Md. – Mediana; DP – Desvio Padrão.

Com relação à corretude, através da Figura 4.6 é possível observar que a mediana dos diagramas, nos métodos PBL e ErrEx, está acima de 0,90. A corretude dos diagramas de atividades para ambos os métodos está dispersa, no entanto, a corretude dos grupos estão acima de 0,80. Esta dispersão poderia ter sido causada por dificuldades na representação correta da notação do diagrama de atividade. Por exemplo, um dos defeitos mais identificados foi a não representação da barra de sincronização de forma correta no diagrama. No que diz respeito a corretude dos diagramas de sequência elaborados com o uso do método PBL, apesar do grupo G7 ter tido um baixo valor, a corretude dos diagramas criados ainda obteve valores próximos

ao da mediana. Por outro lado, a corretude dos diagramas de seqüências criados utilizando o método ErrEx mostrou que os valores das médias foram bem dispersos. Alguns estudantes também tiveram dificuldades em representar corretamente o diagrama de seqüência. Por exemplo, os estudantes não souberam representar que a interação entre os objetos é realizada por meio do controlador. No entanto, ao usar o método PBL, os estudantes conseguiram modelar diagramas de seqüência mais corretos.

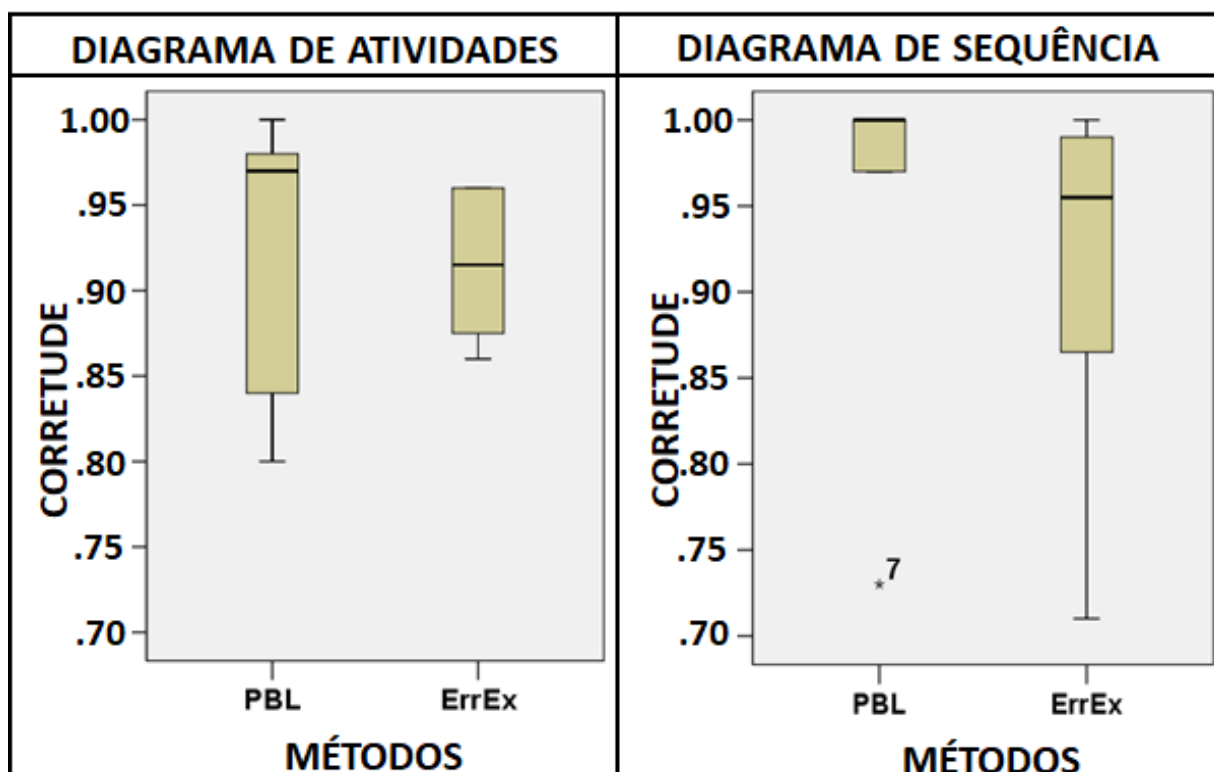


Figura 4.6. Boxplot para a corretude dos diagramas.

Na Figura 4.7, é possível observar que o valor da mediana para a completude dos diagramas de atividades modelados utilizando o método PBL foi 0,88, enquanto que utilizando ErrEx foi de 1,00. Além disso, os valores da completude para os diagramas modelados usando ErrEx são muito próximas da mediana, diferentemente da distribuição observada para PBL. Assim, supõe-se que, ao empregar o método ErrEx, os estudantes conseguiram modelar diagramas de atividades mais completos. Sobre à completude dos diagramas de seqüência modelados utilizando o método PBL, houve dois grupos que tiveram resultados bem abaixo da mediana (G8 e G9). Apesar disto, os resultados para este método também estão bem concentrados em torno da mediana (0,86). A completude do diagrama de seqüência usando o método ErrEx está muito dispersa. Nesse caso, a completude varia entre 0,30 (mínimo) e 0,90 (máximo).

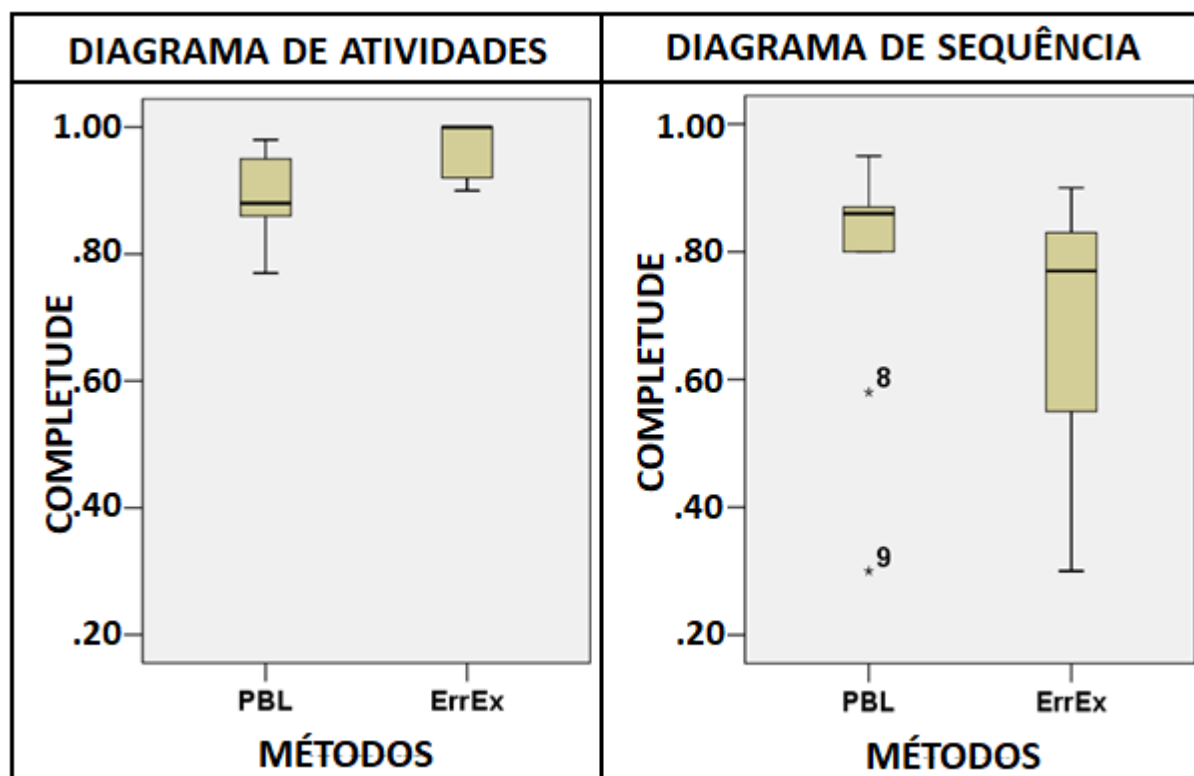


Figura 4.7. Boxplot para a completude dos diagramas.

4.3.2.2 Percepção dos estudantes sobre a preferência dos métodos

Para uma análise comparativa sobre os métodos, os estudantes também responderam a quatro questões, apresentadas na Tabela 4.6.

Tabela 4.6. Preferência dos Métodos para Aprender a modelar.

Objetivo	Questão
Método mais fácil	Qual dos métodos de ensino você considerou mais fácil para aprender a modelagem dos diagramas?
Método mais difícil	Qual dos métodos de ensino você considerou mais difícil para aprender a modelagem dos diagramas?
Preferência	Dentre os métodos utilizados para ensinar diagramas, qual a sua preferência?
Nível de Engajamento	Classifique seu nível de engajamento após utilizar cada um dos métodos (Método Tradicional, PBL e ErrEx)

Na Figura 4.8 pode-se notar que os estudantes consideraram o método PBL mais fácil para aprender a modelar (18 estudantes), seguido pelo Método Tradicional (MT) (13 estudantes) e pelo ErrEx (6 estudantes).

Os estudantes comentaram que é mais fácil aprender utilizando PBL, devido ao método estimular a pensar mais sobre o problema e permitir uma familiaridade com os problemas do mundo real, como mostrado nas citações abaixo:

“O método estimula a pensar mais (...) e aborda a completude da teoria.” – S17

“Eu tive mais facilidade em criar alternativas de solução para os problemas utilizando o PBL.” – S01

“[PBL] ajuda a familiarizar com os problemas do mundo real.” – S26

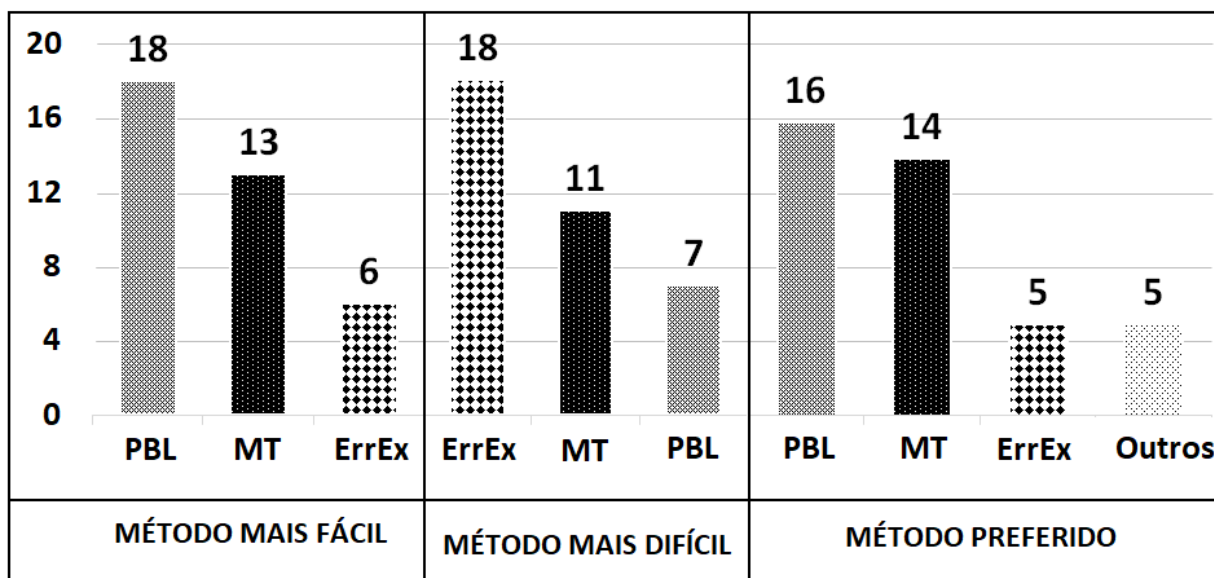


Figura 4.8. Resultados das Percepções dos estudantes sobre os métodos.

De acordo com a Figura 4.8, o método ErrEx foi considerado o mais difícil para aprender a modelar (18 estudantes), seguido pelo Método Tradicional (11 estudantes) e pelo PBL (7 estudantes). O método ErrEx pode ter sido considerado mais difícil devido a dois fatores: (a) a adição de mais uma etapa durante o processo de ensino; e (b) a dificuldade em identificar os erros nos diagramas, como mostrado nas citações abaixo:

“O ErrEx adiciona mais uma etapa [a busca pelos erros], o que acaba quebrando o raciocínio que construímos ao ler a especificação.” – S32

“o método ErrEx não contribuiu tanto quanto os outros, pois tínhamos dificuldades para encontrar os erros.” – S33

Com relação à preferência do método para aprender a modelar (Figura 4.8), os resultados mostraram que os estudantes preferem aprender a modelar utilizando o PBL (16 estudantes), seguido pelo Método Tradicional (14 estudantes), o ErrEx (5 estudantes) e Outros Métodos (5 estudantes). Com relação à preferência pelo método PBL, os estudantes comentaram que o método proporciona uma aprendizagem mais concreta, como mostrado na citação abaixo:

“Acredito que o PBL oferece uma aprendizagem mais concreta quando aplicamos e discutimos conceitos.” – S32

O MT foi o segundo método escolhido pelos estudantes. Isso deve-se ao fato dos estudantes se sentirem mais seguros quando o professor está explicando o assunto (ver citações abaixo):

“com o direcionamento do professor, fica mais fácil de evitar erros e sanar dúvidas.” – S07

“Sinto-me muito motivado a aprender e tentar aplicar corretamente os conceitos ensinados.”
– S29

O ErrEx foi o terceiro método preferido pelos estudantes. Isto porque o ErrEx é prático e por ser possível aprender o que não fazer durante a modelagem, como visto a seguir:

“Eu considero mais rápido aprender o que não fazer utilizando o ErrEx.” – S08

Além dos métodos mencionados acima, alguns estudantes preferem aprender a modelar utilizando “Outros Métodos” (5 estudantes). Infelizmente, dois estudantes não informaram o nome de quais seriam esses outros métodos, um estudante ressaltou que prefere *“vídeo aula/ ensino à distância, com algumas aulas práticas para tirar dúvidas”* (S02). Já outro estudante informou que prefere aprender sozinho. Um estudante relatou que prefere a combinação dos três métodos (MT, PBL e ErrEx). Percebeu-se que alguns estudantes relataram que *“os três métodos são necessários para aprender a modelos”* e que poderia ser feita a combinação do MT com os métodos PBL e ErrEx. Segundo os estudantes, esta combinação auxiliaria na obtenção de diferentes visões do mesmo problema, conforme pode ser visto nas citações abaixo:

“os três são necessários, o PBL e o ErrEx combinado com o (método) tradicional seria melhor.” – S20

“Prefiro um método que possa englobar todos, para termos diversas visões acerca do mesmo problema.” – S23

“Tem que ter o método tradicional obviamente. Muito do PBL e alguns do ErrEx para detectarmos possíveis erros.” – S22

4.3.2.3 Percepção dos estudantes sobre o método PBL e ErrEx

Questionários pós-modelagem foram aplicados e analisados com o objetivo de avaliar a aprendizagem percebida pelos estudantes quanto ao uso dos métodos PBL e ErrEx. Os estudantes forneceram suas respostas em uma escala Likert de cinco pontos, com alternativas de resposta variando de discordo fortemente (-2) a concordo fortemente (2). Os itens dos questionários, apresentados na Tabela 4.7, foram baseados nas dimensões de aprendizagem (Sindre e Moody, 2003; Wangenheim *et al.*, 2013). Os itens 01 e 02 visaram avaliar se os

métodos contribuíram para a aprendizagem na disciplina e os itens 03 ao 08 foram elaborados de acordo com os níveis de aprendizagem da Taxonomia de Bloom (Bloom, 1956).

Tabela 4.7. Itens avaliados nos questionários pós-modelagem.

Item	Descrição do Item
Item 01	O método PBL/ ErrEx contribuiu para a minha aprendizagem na disciplina.
Item 02	O método PBL/ ErrEx foi eficiente para minha aprendizagem, em comparação com outras atividades da disciplina.
Item 03	O método PBL/ ErrEx auxiliou a relembrar os conceitos aprendidos sobre diagrama.
Item 04	O método PBL/ ErrEx contribuiu para interpretar como os conceitos aprendidos podem ser utilizados na modelagem do diagrama.
Item 05	O método PBL/ ErrEx contribuiu para aplicar os conceitos do diagrama durante a resolução dos problemas.
Item 06	O método PBL/ ErrEx contribuiu para organizar o diagrama durante a modelagem.
Item 07	O método PBL/ ErrEx contribuiu para verificar se o diagrama foi modelado corretamente.
Item 08	O método PBL/ ErrEx contribuiu para criar o diagrama durante a modelagem.

A Figura 4.9 apresenta a percepção dos estudantes com relação ao aprendizado dos diagramas de atividades e sequência utilizando os métodos PBL e ErrEx. No que diz respeito ao aprendizado do diagrama de atividades, percebeu-se que a maioria dos estudantes concordou que os métodos contribuem positivamente para a aprendizagem. Para o item que avalia se o método contribuiu para a aprendizagem (item 01) obteve-se um bom nível de concordância para ambos os métodos (87% de concordância para o PBL e 95% de concordância para o ErrEx). No item que avalia se o método ajuda a modelar o diagrama corretamente (item 07), obteve-se a menor taxa de concordância para o método PBL (64% de concordância). Além disso, com relação ao método PBL, este item foi o que teve a maior taxa de discordância (18% de discordância). Estas discordâncias podem estar relacionadas ao fato de o método ter sido utilizado apenas com o objetivo de criar os diagramas. Contudo, a percepção geral dos estudantes ao utilizar os dois métodos para aprender diagrama de atividades foi classificada como muito positiva. Isso demonstra que o uso de PBL e ErrEx proporcionou uma boa percepção para os estudantes, especialmente em termos de relembrar, interpretar e aplicar conceitos.

No que diz respeito ao aprendizado do diagrama de atividades, percebeu-se que a maioria dos estudantes concordou que os métodos contribuem positivamente para a aprendizagem. Para o item que avalia se o método contribuiu para a aprendizagem (item 01) obteve-se um bom nível de concordância para ambos os métodos (87% de concordância para o PBL e 95% de concordância para o ErrEx).

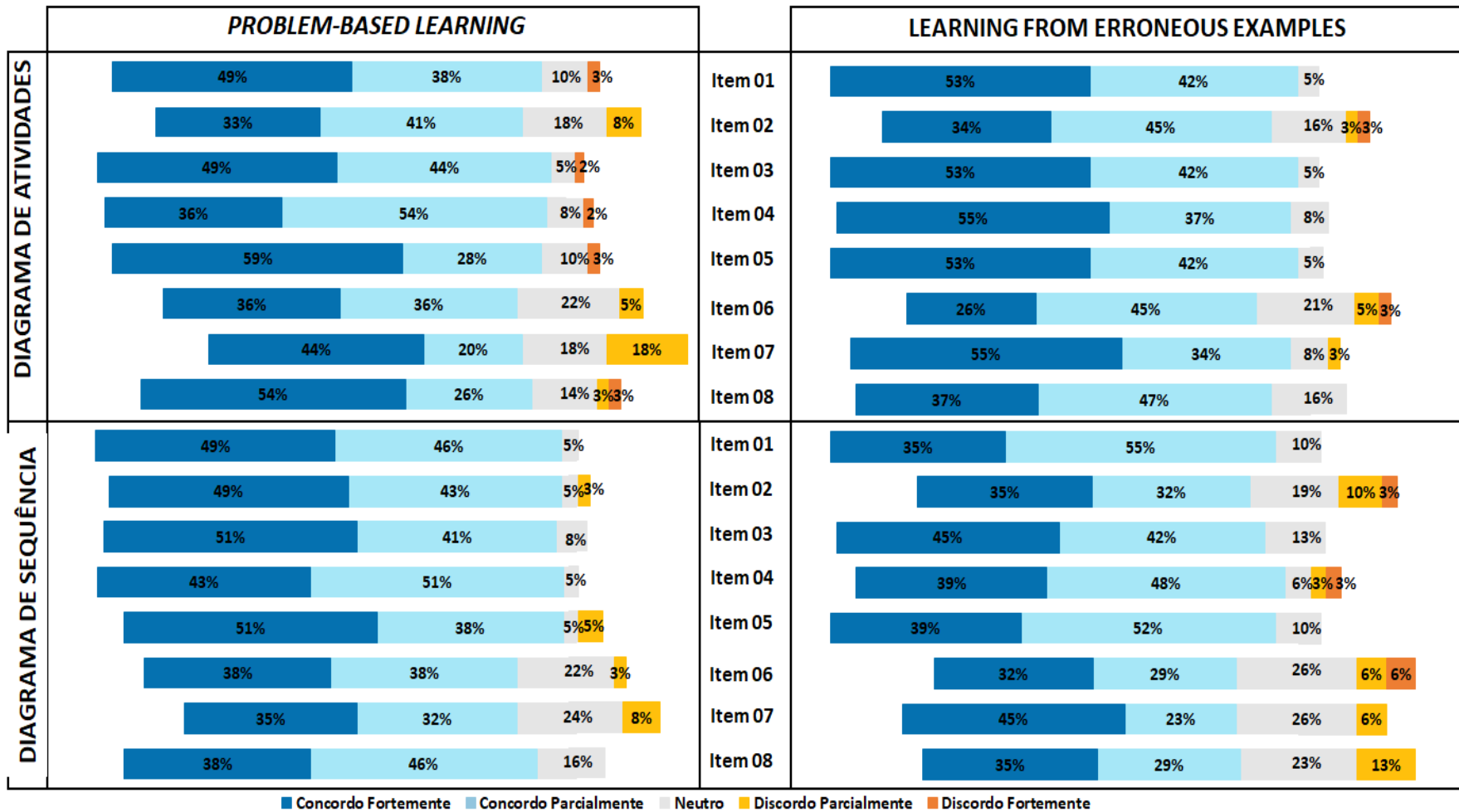


Figura 4.9. Percepção dos estudantes sobre os métodos PBL e ErrEx.

No item que avalia se o método ajuda a modelar o diagrama corretamente (item 07), obteve-se a menor taxa de concordância para o método PBL (64% de concordância). Além disso, com relação ao método PBL, este item foi o que teve a maior taxa de discordância (18% de discordância). Estas discordâncias podem estar relacionadas ao fato de o método ter sido utilizado apenas com o objetivo de criar os diagramas. Contudo, a percepção geral dos estudantes ao utilizar os dois métodos para aprender diagrama de atividades foi classificada como muito positiva. Isso demonstra que o uso dos métodos PBL e ErrEx proporcionou uma boa percepção para os estudantes, especialmente em termos de relembrar, interpretar e aplicar conceitos.

No que diz respeito ao aprendizado de diagrama de sequência utilizando os métodos PBL e ErrEx, percebeu-se que o item 01 (contribuição para a aprendizagem) foi o mais bem avaliado pelos participantes. De acordo com a percepção dos estudantes, ambos os métodos contribuem para a aprendizagem dos diagramas na disciplina. No item 02 (eficiência do método), percebeu-se um alto nível de concordância para o método PBL (92% de concordância). Porém, houve 67% de concordância dos estudantes em relação ao método ErrEx. Ainda em relação ao método ErrEx neste item, percebe-se que houve estudantes discordando (13%) ou permanecendo neutros (19%). Percebeu-se também que no item 08 (criar o diagrama), 84% dos estudantes concordaram que o método PBL auxilia na criação do diagrama, enquanto 64% dos estudantes concordaram com este item para o método ErrEx. Ainda neste item, com relação ao método ErrEx, percebeu-se que houve uma grande taxa de discordância (13%) e de respostas neutras (23%). Analisando a percepção dos participantes em relação à aprendizagem do diagrama de sequência, pode-se perceber que o método ErrEx, quando comparado ao PBL, obteve uma grande quantidade de participantes discordando e permanecendo neutro nos itens avaliados. Isso demonstra que o método PBL mostrou-se mais eficiente na criação dos diagramas, auxiliando na aprendizagem do diagrama de sequência.

No que diz respeito ao nível de engajamento dos estudantes em relação aos métodos, na Figura 4.10 é possível perceber que os estudantes se sentiram mais engajados utilizando o método PBL (76% de concordância). Além disso, a menor taxa de discordância foi para este método (10%). Em relação aos métodos PBL e MT, os estudantes concordaram que estes auxiliam no engajamento, respectivamente 60% e 55% de concordância. Apesar de não haver muitos participantes discordando, é possível perceber que houve mais discordância com relação ao método ErrEx (25%).

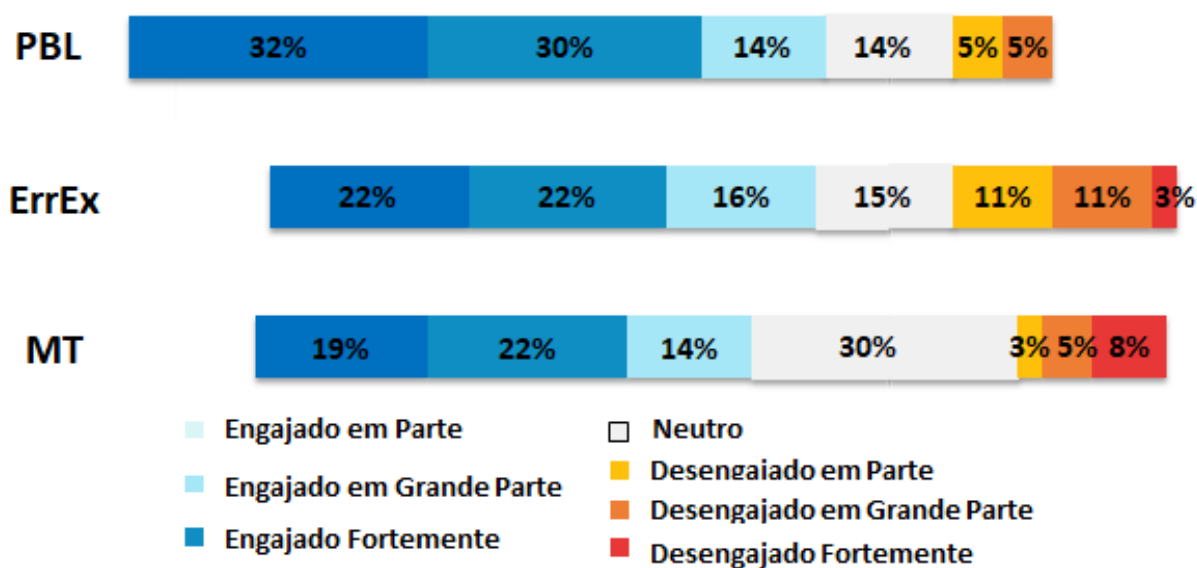


Figura 4.10. Nível de engajamentos dos estudantes em relação aos métodos.

4.3.2.4 Resultados qualitativos

Assim como nos estudos anteriores, foi realizado procedimentos de codificação com o intuito de identificar códigos que descrevessem as dificuldades e os benefícios percebidos pelos estudantes após utilizar os métodos (PBL e ErrEx). A análise foi conduzida por um pesquisador e foi discutida com outros dois pesquisadores. Isto foi realizado visando mitigar o viés eventualmente causado pela participação de um único pesquisador no processo de codificação. As subseções seguintes apresentarão mais detalhes sobre os resultados identificados na análise qualitativa.

4.3.2.4.1 Percepção dos estudantes com relação ao método PBL

A partir da análise dos dados qualitativos, foram três principais categorias: (1) PBL ajudou na aprendizagem dos diagramas, (2) A discussão em grupo utilizando PBL melhorou o aprendizado do diagrama e (3) PBL não ajudou na aprendizagem durante a modelagem.

A Categoria “**PBL ajudou na aprendizagem dos diagramas**” apontou três subcategorias sobre facilidades em aprender a modelar utilizando o PBL. Com relação ao primeiro código (PBL estimulou a compreender os conceitos aprendidos sobre os diagramas), o estudante S01 comentou que “*PBL com certeza ajudou a lembrar e aplicar os conceitos aprendidos sobre o diagrama.*” Além disso, o estudante S07 ressaltou que com o uso do PBL “*os conceitos que não foram bem assimilados por alguns integrantes foram lembrados e aplicados por outros que compreenderam bem.*” Outros estudantes ressaltaram que o PBL ajudou a modelar o problema de forma prática. O estudante S07 comentou que o “*problema*

ajudou a ter uma ideia melhor de como funciona na prática o sistema e, dessa forma, auxiliou a modelar o diagrama.” O estudante S27 complementou dizendo que a *“apresentação de um cenário real instiga a explorar os conceitos e possibilidades para solucionar o problema.”* Além disso, PBL ajudou a mostrar a relação entre os diagramas, ou seja, os estudantes conseguiram entender a relação de dependência entre os diagramas. Com relação a isto, S19 disse que *“conseguiu ver a ligação direta que o diagrama de classes tem como diagrama de sequência e como isso torna mais fácil o desenho do diagrama de sequência.”* Isto mostra que o PBL ajudou a compreender a importância de um diagrama para a modelagem de outro.

A Categoria **“A discussão em grupo utilizando PBL melhorou o aprendizado do diagrama”** apresenta alguns dos benefícios de discutir com o grupo utilizando PBL na modelagem dos diagramas. Evidenciou-se que PBL melhorou a interação entre os membros da equipe. Para ilustrar esta subcategoria, o estudante S40 comentou que *“o método ajudou a melhorar a aprendizagem pelo fato de estimular bastante o contato com outras pessoas, estimulando também o diálogo, facilitando a retirada de informações do problema.”* Para complementar, o estudante S41 ainda comentou que *“debater em equipe permitiu tirar ‘dúvidas bobas’ que às vezes são constrangedoras fazer durante uma aula.”* Isto demonstra que o PBL permite que os estudantes alcancem uma aprendizagem coletiva.

Outro benefício é que PBL permitiu os estudantes obterem diferentes perspectivas sobre o problema. O estudante S37 confirmou que a partir da *“interação com os outros colegas foi possível verificar outros pontos de vista sobre o mesmo problema e diferentes formas de elaborar soluções.”* O estudante S04 ainda ressaltou que com esta interação *“cada integrante [da equipe] deu sua opinião e assim tiramos uma melhor conclusão do problema.”* Por fim, PBL ajudou a modelar corretamente o diagrama. O estudante S01 disse que *“a discussão em grupo foi fundamental para a criação de diagramas mais corretos e mais completos.”* Além disso, de acordo com o estudante S33, o método PBL *“fez fluir a discussão entre os membros da equipe para fazer o diagrama corretamente.”*

Os códigos relacionados às categorias acima mostram indícios de que os estudantes tiveram uma boa percepção sobre o método PBL. Contudo, foram identificadas algumas dificuldades percebidas pelos estudantes, em que estes enfatizam que o **“PBL não ajudou na aprendizagem durante a modelagem”**. Uma das dificuldades percebidas pelos estudantes é que o tamanho da equipe influenciou negativamente a modelagem. Com relação a isto, o estudante S14 comentou que *“como o grupo ficou muito grande (5 pessoas), houve muita divergência de opinião, o que dificultou um pouco a modelagem.”* O estudante S14 ressaltou

que “*com o grupo menor, o diagrama sai mais facilmente*”. Outra dificuldade é sobre as diferentes visões dos membros da equipe dificultam a modelagem. Conforme relatado pelo estudante S26, durante a modelagem ficou “*complicado conciliar as diferentes visões e abstrações dos membros*.” O estudante S22 ainda complementou dizendo que a modelagem “*demorou mais pelas diferentes visões [...] e isso tornou a modelagem mais complicada*.” Além disso, foram identificadas dificuldades relacionadas a problemas de interação na equipe. Segundo o estudante S23, utilizar o PBL “*acabou dificultando a execução do trabalho*.” Este estudante ainda citou alguns dos motivos para estes problemas: “*ideias contraditórias, muita gente, muita confusão na aplicação de alguns conceitos aprendidos em sala*.”

Outra dificuldade percebida foi em relação a dúvidas sobre a aplicação correta do método PBL. Sobre essa dificuldade, o estudante S29 ressaltou que “*no contexto mais específico da discussão, acredito que falhamos um pouco na aplicação do método*.” Uma hipótese para esta dificuldade ter ocorrido é que este estudante não entendeu claramente o objetivo da atividade. Além disso, os estudantes sentiram a necessidade da orientação do professor, pois, segundo o estudante S10, o grupo “*perdeu muito tempo discutindo, às cegas*.” Já o estudante S20 comentou que isto pode ter ocorrido por que “*em vários momentos a mesma coisa a qual eu tinha dúvidas/não sabia, meus colegas também não sabiam ou tinham dúvidas*.” Segundo esse estudante, faltou o professor ou o monitor da disciplina para direcioná-lo de maneira mais adequada.

4.3.2.4.2 Percepção dos estudantes com relação ao método ErrEx

Na análise sobre a percepção dos estudantes, foram identificadas também três categorias: (1) ErrEx ajudou na aprendizagem do diagrama, (2) a discussão em grupo utilizando ErrEx melhorou o aprendizado do diagrama e (3) ErrEx não ajudou na aprendizagem durante a modelagem.

Com relação à categoria **ErrEx ajudou na aprendizagem do diagrama**, foi possível identificar algumas subcategorias com benefícios de utilizar o método ErrEx. Segundo os estudantes, o ErrEx permite identificar defeitos para realizar a modelagem. Com relação a esta subcategoria, o estudante 26 comentou que o método ErrEx “*mostra exemplos incorretos que devem ser corrigidos e estes erros podem reaparecer em uma modelagem real*.” Além disso, este estudante afirmou que “[...] *verificando os erros eu consigo aprender melhor*.”

Identificar os erros evitou que estes fossem repetidos durante a nova modelagem foi mencionado pelo estudante S40, que ressaltou que “*por ter mostrado um diagrama com defeitos,*

fica mais fácil modelar o nosso sabendo o que não fazer.” Concordando com esta afirmativa, o estudante S28 ainda ressalta que *“quando vemos os erros, aprendemos mais para não errar em nossas modelagens.”* Além disso, o método ErrEx ajudou a compreender os conceitos aprendidos. Assim como no PBL, os estudantes conseguiram compreender os conceitos que foram ensinados em sala utilizando o ErrEx. Com relação a este benefício do método, o estudante S29 ressaltou que o método ErrEx *“é muito eficiente para mim, pois o conceito e o conteúdo ficam mais fixos em minha mente.”* Já o estudante S23 ressaltou que o ErrEx *“permite relembrar os conceitos aplicando a correção sobre o artefato.”* Foi possível perceber ainda que no ErrEx, similarmente ao método PBL, as diferentes visões auxiliaram a modelar o diagrama. O estudante S29 disse que o método *“contribuiu para percepção de como diferentes visões e diferentes soluções podem nos fazer nos observar melhor as soluções, antes despercebidas.”* Com relação a disponibilizar um diagrama ajuda na modelagem, segundo o estudante S31, *“ter um diagrama como ponto de partida, mesmo que contenha erros, também ajuda na hora de criar um novo com as correções.”* Isto demonstra que apesar do diagrama apresentar diferente tipos de defeitos, isto ajuda os estudantes no início da modelagem.

Evidenciou-se ainda que **A discussão em grupo utilizando ErrEx melhorou o aprendizado do diagrama**. As evidências relativas a essa categoria foram agrupadas em duas subcategorias que evidenciam isto. Uma das subcategorias é ErrEx melhorou a interação entre os membros da equipe. O estudante S02 mencionou que *“a discussão com os colegas esclarece algumas dúvidas e ajuda a identificar problemas que às vezes não conseguimos identificar.”* O estudante S03 complementou que *“debatendo e vendo a opinião do grupo, como eles fariam algumas coisas, pode melhorar meu entendimento do diagrama.”* A segunda subcategoria é denominada ErrEx nivelou o conhecimento entre os membros da equipe. Sobre isto, o estudante S12 afirmou que *“as discussões são sempre muito ricas para nivelarmos nossos conhecimentos, sendo essa a parte que julgo fundamental para alinhar, criar esboço e modelar.”* Já o estudante S41 ressaltou que *“o fato de ser em equipe, (isto) nivelou nossos conhecimentos.”*

Por fim, os estudantes perceberam algumas dificuldades indicando que o método **ErrEx não ajudou na aprendizagem durante a modelagem**. Eles notaram que ErrEx não auxiliou como modelar corretamente o diagrama. O estudante S18 apontou que *“(...) descobriu uma nova forma de fazer diagramas incorretamente, ao invés de aprender como fazer corretamente de fato.”* Já o estudante S11 disse que *“no geral ajudou afinal se trata de uma outra maneira de encarar o problema e apesar de não ajudar muito na modelagem e criação do diagrama, este*

método ajuda a verificar se o diagrama está bem certo ao mostrar os erros, ou seja, o que não fazer.”

Os estudantes relataram ainda o fato de o ErrEx possuir um passo de identificação de erros que dificulta a modelagem. Segundo o estudante S36, *“você perde tempo procurando os erros em vez de fazer o diagrama.”* Além disso, o estudante S32 ressaltou que *“o fato de primeiro procurar erros para depois corrigi-los e fazer um novo diagrama faz com que meu pensamento ficasse um pouco bagunçado.”* Este estudante ainda disse que prefere *“ler o problema e já ir modelando, sem esse passo adicional no meio.”* Ainda segundo os estudantes o ErrEx diminui a interação entre os membros da equipe, uma vez que os estudantes não começam o diagrama do zero. Nesse sentido, o estudante S15 disse que ter um diagrama incompleto diminui a interação, uma vez que *“criar um diagrama do zero força uma discussão maior com a equipe.”*

Assim como no PBL, os estudantes comentaram que equipes com muitos membros atrapalham a modelagem. De acordo com o estudante S14, *“muitas pessoas atrapalham no desenvolvimento do diagrama, recomendo o número de 3 pessoas escolhidas de forma sortida.”* Vale a pena ressaltar que esta dificuldade foi relatada pelo mesmo estudante em ambos os métodos, demonstrando que, talvez, este estudante não goste de realizar atividades em grupo.

4.3.3 Discussão dos resultados do estudo

Os resultados qualitativos mostram que ambos os métodos (a) ajudam a compreender os conceitos dos diagramas, (b) melhoram a interação entre a equipe e (c) o tamanho da equipe dificulta no aprendizado dos diagramas. Com relação a compreender conceitos, no método PBL percebeu-se que os estudantes conseguiram compreender os conceitos devido a interação e a discussão que houve entre os membros durante a modelagem do diagrama. Porém, no método ErrEx, os estudantes conseguiram compreender os conceitos devido ao diagrama com defeitos que estes possuíam. Com relação à interação (b), ambos os métodos ajudaram na melhoria deste fator. Percebeu-se que a interação do método PBL ajudou os estudantes a discutirem os seus pontos de vista sobre o problema e também na modelagem correta do diagrama. Já no ErrEx, a interação fez com que os estudantes nivelassem o nível de conhecimento dos membros da equipe que ajudou a modelar o diagrama. Com relação ao tamanho da equipe dificultar no aprendizado dos diagramas (c), um estudante sugeriu que os grupos poderiam ser formados por duas ou três pessoas, assim várias discussões não atrapalhariam na modelagem do diagrama.

Também foram identificados alguns fatores que foram considerados tanto positivos quanto negativos. No PBL, o fator *“diferentes perspectivas do problema”* pode ajudar os

participantes a discutir, observar e decidir qual o ponto de vistas mais adequado para realizar a modelagem do diagrama. Contudo, alguns participantes consideraram que discutir os diferentes pontos de vistas tornou a modelagem do diagrama uma atividade complexa e demorada. Já com o método ErrEx, houve divergência sobre o diagrama com defeitos. Alguns estudantes afirmaram que o diagrama, mesmo errado, ajuda a iniciar a modelagem e que identificar os defeitos evita repetir tais erros, além de ajudar a compreender alguns conceitos. Contudo, alguns participantes ressaltaram que este passo apenas identifica defeitos e não auxilia a construir o diagrama. Além disso, os estudantes ressaltaram que o este passo diminui a interação entre a equipe, pois os estudantes iniciam a modelagem a partir do diagrama errado, sem discutir direito o cenário.

4.3.4 Considerações sobre o estudo

Esta subseção descreveu um estudo experimental teve como objetivo comparar dois métodos de ensino no contexto do ensino de diagramas UML, *Problem Based Learning* (PBL) e *Learning from Errouneous Examples* (ErrEx). A análise dos resultados quantitativos mostrou que a mediana da corretude dos diagramas (Atividades e Sequência) gerados a partir do uso do PBL e ErrEx estão acima de 0.91. Esses resultados demonstram que adotar PBL e ErrEx auxilia os estudantes a modelarem os diagramas satisfatoriamente corretos. Em relação à completude, a mediana dos resultados que utilizaram o método PBL está acima de 0,85 e que empregaram o método ErrEx está acima de 0,76. No entanto, os resultados quantitativos mostraram que os estudantes modelaram os diagramas de atividades mais completos usando o método ErrEx (mediana igual a 1,00) e um diagrama de sequência mais completo usando o método PBL (mediana acima de 0,85).

Com relação à percepção dos estudantes sobre métodos, o PBL foi considerado o método mais fácil e método preferido para aprender a modelar os diagramas. Os motivos desta preferência podem estar relacionados ao fato de PBL estimular a pensar sobre o problema de maneira mais crítica. Já o ErrEx foi considerado o método mais difícil. Uma das razões é o fato do método de fazer o estudante primeiro identificar os problemas e depois realizar a modelagem. Contudo, vale ressaltar que houve estudantes que preferiram este método apenas por ele proporcionar um aprendizado já na etapa de identificação dos defeitos. No que diz respeito à aprendizagem percebida, é possível observar que, no geral, muitos estudantes concordaram com as questões relacionadas à estas percepções para ambos os métodos, ou seja, o método PBL e o ErrEx auxiliam no aprendizado de diagramas de atividades e de sequência.

A análise qualitativa permitiu a identificação de benefícios e dificuldades que influenciam positivamente e negativamente no aprendizado empregando os métodos. Os benefícios e dificuldades identificados podem ajudar os professores na seleção de um dos métodos para apoiar os estudantes a alcançar um determinado objetivo de aprendizagem. Por exemplo, se o professor quiser que os estudantes aprendam algo de forma mais prática e que ajude a relembrar alguns conceitos, o método a ser adotado é o PBL. Porém, o professor deve se fazer mais presente durante este processo e adequar os grupos para que não tenham problemas de interação devido ao tamanho da equipe ou as diferentes visões. Mas se o professor desejar que os estudantes alcancem as primeiras habilidades sobre o entendimento de um objetivo de aprendizagem específico, por exemplo, o método mais adequado é o ErrEx. Ressaltando que ambos os métodos melhoram a interação entre os membros da equipe. Por fim, acredita-se que a criação de uma abordagem no qual combine PBL e ErrEx com o método tradicional deve ser um ponto a ser discutido. A partir dos resultados deste estudo, percebeu-se que os estudantes justificaram suas escolhas baseados na forma como aprendem.

Com base no resultado do estudo apresentado acima, percebeu-se que os professores podem utilizar diferentes estratégias de ensino como forma de melhorar o engajamento dos estudantes e melhorar o processo de ensino e aprendizagem em sala de aula. Além dessas estratégias, professores tendem também a utilizar recursos educacionais em que os estudantes se divertem enquanto aprendem o conteúdo, assim, estimulando a curiosidade e provendo uma motivação para o aprendizado. Na área de Educação em Engenharia de Software, esses recursos educacionais vêm ganhando bastante destaque e sendo estudados constantemente. Um destes recursos são os jogos, que podem ser agregados à estratégia de ensino-aprendizagem de Engenharia de Software. Portanto, a motivação do próximo estudo partiu da adoção de um jogo educacional para apoiar o ensino de modelagem de software.

4.4 ESTUDO EXPERIMENTAL DE UM JOGO EDUCACIONAL PARA APOIAR O ENSINO DE DIAGRAMA DE ATIVIDADES

Os professores adotam diversos recursos educacionais como forma de melhorar o engajamento dos estudantes. Um dos recursos pedagógicos disponíveis aos professores são os jogos educacionais, que têm por objetivo despertar os interesses dos estudantes pelo conteúdo que está sendo ensinado (Souza e França, 2016). Contudo, Petri e Chiavegatti (2015) afirmam que os jogos existentes na literatura objetivam apenas relembrar os conceitos aprendidos na

aula. Assim, os estudantes não conseguem aplicar tais conteúdos de forma mais prática, como, por exemplo, na elaboração de um diagrama da UML.

Assim, esta subseção apresenta um estudo experimental realizado com um jogo educacional chamado *ActGame (Activity Diagram Game)*. O *ActGame* é um jogo baseado em cartas e tem por objetivo ensinar os estudantes a construir um dos modelos ensinados nos cursos de computação, o Diagrama de Atividades (DA) da UML (*Unified Modeling Language*) (OMG, 2010). O DA é utilizado para representar, de forma mais detalhada, a lógica das atividades que podem ser realizadas no software (Booch *et al.*, 2005). Este jogo utiliza alguns elementos de gamificação (Deterding *et al.*, 2011), visando incentivar a competição entre os estudantes durante a modelagem do DA. Para analisar a viabilidade do jogo em um ambiente acadêmico, foi realizada uma avaliação experimental utilizando um modelo específico para avaliação de jogos educacionais, o MEEGA (*Model for the Evaluation of Educational Games*) (Savi *et al.*, 2011). O *ActGame* foi utilizado por estudantes de um curso de Ciência da Computação para avaliar o grau de aprendizado de DA após a utilização do jogo.

4.4.1 Jogo Educacional *ActGame*

ActGame (Activity Diagram Game) é um jogo educacional baseado em cartas, que visa ajudar os estudantes durante a modelagem de DA. A fim de ajudar os estudantes, o jogo auxilia na identificação dos elementos do DA e também na construção do diagrama a partir de uma descrição textual. Com o intuito de facilitar o entendimento dos estudantes em relação à notação do DA, as cartas do jogo foram criadas tomando como base os principais elementos do DA (Booch *et al.*, 2005). O *ActGame* foi projetado para ser utilizado individualmente e em grupo. Individualmente, apenas um jogador irá interagir com o jogo seguindo suas etapas. Em grupo (equipes de três a cinco jogadores), os jogadores devem discutir livremente o problema e em seguida realizar a modelagem seguindo as etapas do jogo. O jogo possui cinco etapas: Identificação das Swimlanes (Etapa 1), Identificação do Nó de Início (Etapa 2), Identificação das atividades (Etapa 3), Agrupamento das atividades (Etapa 4), Transição das Atividades (Etapa 5), Identificação do Nó Final (Etapa 6). Cada etapa possui 4 cartas: Passo da Modelagem, Perguntas-Guia, Dicas e Exemplo. A descrição das cartas do jogo está na Tabela 4.8.

Tabela 4.8. Descrição das Cartas utilizadas no jogo.

Tipo de Carta	Descrição
Carta Passo da Modelagem	Esta é a primeira carta a ser utilizada em cada uma das etapas do jogo. Nessa carta há: uma heurística de construção que ajuda a identificar os elementos do diagrama a partir da descrição textual (Figura 4.11 Item A);Instruções de Construção que ajudam a inserir esses elementos no diagrama (Figura 4.11 Item B);um exemplo do elemento

	(Figura 4.11 Item C); e um exemplo de aplicação que mostra de forma ilustrativa como extrair o elemento do diagrama de atividades a partir de uma descrição textual (Figura 4.11 Item D).
Carta Perguntas-Guia e Carta Dicas	A carta Perguntas-Guia e Dicas apresentam, respectivamente, perguntas (Figura 4.11 Item E) e dicas (Figura 4.12 Item A) que direcionam os jogadores na identificação dos elementos e na construção do DA. Além disso, cada carta possui um exemplo de como identificar o elemento a partir de uma descrição textual (Figura 4.11 Item F e Figura 4.12 Item B).
Carta Exemplo	Esta carta apresenta mais um exemplo de como o elemento deve ser modelado no DA (Figura 4.12 Item C).

Carta Passo da Modelagem

Passo 01 – Identificação da Swimlanes

Heurística 01 A

Papéis dos atores envolvidos na descrição

Os atores podem ser identificados através das responsabilidades que podem assumir na descrição. São os responsáveis pela execução das atividades.

B B **Instruções para a construção**

Instrução 01 - Identifique quem são os usuários envolvidos na descrição

Instrução 02 - Identifique quais os papéis que estes usuários assumem na descrição textual

Instrução 03 - Transforme os papéis dos atores em swimlanes, conforme está apresentado no Exemplo do Elemento

- Caso o ator realize alguma interação com a aplicação/sistema, transforme a aplicação/sistema em uma swimlane

Exemplo do Elemento C

Swimlane 1	Swimlane 2	Swimlane 3


Swimlane

Exemplo de Aplicação D

Parte de uma descrição textual

José necessita realizar uma viagem para São Paulo e precisa fazer as reservas de um voo. Para isso, ele precisa executar a atividade A. Em seguida, a aplicação realiza a atividade B.

Conversão dos papéis para as swimlanes




Swimlanes modeladas

Cliente	Aplicação

Carta Perguntas-Guia

Quem serão as Swimlanes?



Olá! Se você ainda está com dúvidas sobre quem serão as swimlanes do diagrama de atividades, veja o exemplo abaixo.

E

PERGUNTAS-GUIA

- Quem são os atores que podem realizar as ações no sistema/aplicação?
- Quais responsabilidades estes atores assumem na aplicação/sistema?
- A aplicação realiza alguma interação com os atores?

F

Exemplo de uma descrição

“José necessita realizar uma viagem para São Paulo e precisa fazer as reservas de um voo. Para isso, ele precisa executar a atividade A. Em seguida, a aplicação realiza a atividade B.”

Cliente	Aplicação

Figura 4.11. Exemplo das cartas Passo da Modelagem e Perguntas-Guia.

Para motivar os estudantes durante a construção do DA, o jogo emprega elementos de gamificação (Deterding *et al.*, 2011). Cada tipo de carta possui uma pontuação diferente e ao utilizá-las os jogadores ganham um emblema com uma pontuação (Figura 4.13). Por exemplo, quando os jogadores utilizam apenas um tipo de carta em uma etapa do jogo, eles ganham um emblema com a pontuação da carta utilizada. Porém se utilizarem as cartas combinadas, por exemplo, a carta Passo da Modelagem e a carta Exemplo, estes ganham os pontos referentes às duas cartas. O objetivo é fazer com que os jogadores utilizem mais cartas possíveis para que aprendam a modelar. Além disso, há mais dois emblemas com uma pontuação extra para a equipe (a) que terminar primeiro a modelagem e (b) cujo diagrama estiver mais correto e completo. Ganha o jogo a equipe que tiver mais pontos no final. Vale frisar que o professor atua como moderador do jogo. O moderador deve explicar as regras do jogo às equipes, entregar a

primeira carta (Passo da Modelagem) e informar quais cartas ainda estão disponíveis, caso a equipe solicite também outra(s) carta(s). Além disso, o moderador deve anotar as cartas solicitadas por cada equipe em cada etapa do jogo.



Carta Dicas	Carta Exemplos
<p style="text-align: center;">Como identificar os atores?</p> <div style="display: flex; align-items: center;">  <div style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; background-color: #ffe0b2;"> <p>Olá! Se você ainda está com dúvidas sobre quem são os atores da descrição textual as dicas abaixo podem te ajudar.</p> </div> </div>	<p style="text-align: center;">Exemplos</p> <div style="display: flex; align-items: center;">  <div style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; background-color: #ffe0b2;"> <p>Olá! Se você ainda está com dúvidas sobre quem são os atores da descrição textual veja o exemplo abaixo.</p> </div> </div>
<p style="text-align: center;">DICAS A</p> <ul style="list-style-type: none"> O ator representa alguém ou alguma coisa que interage com a aplicação. O nome que identifica o ator deve ser o papel que ele representa para a aplicação. Ator é o papel e não a pessoa. 	<p style="text-align: center;">C Exemplo para identificar um ator</p> <p>João gostaria de solicitar um produto do departamento de compras de uma empresa. Para realizar a solicitação do produto ele deve, primeiramente, realizar login no sistema. Após o login, o sistema apresenta uma tela pedindo que João preencha os campos de código do produto, quantidade e qual é o departamento que está solicitando(...).</p>
<p style="text-align: center;">Exemplo para identificar um ator B</p> <p><u>João</u> deseja comprar um produto em uma loja de eletrônicos. Após escolher o seu produto, <u>João</u> chega ao ponto de vendas com os produtos da compra. Para cada produto trazido pelo cliente, <u>Maria</u> registra o código e a quantidade do produto. O <u>sistema</u>, por sua vez, determina o preço do produto e o adiciona à compra</p> <p style="text-align: center;">Identificando atores</p> <ul style="list-style-type: none"> João é o cliente Maria é a atendente da loja Sistema é a aplicação com quem Maria está interagindo 	<p style="text-align: center;">Identificando atores</p> <ul style="list-style-type: none"> João é o usuário, logo o papel que ele assume na aplicação é o de funcionário. Sistema é com quem o João está interagindo, logo é também assume um papel. Departamento de compras é um setor na qual interage com o sistema para solicitar a compra, logo, ele também assume um papel.

Figura 4.12. Exemplo das cartas Dicas e Exemplo.

	<p>Valor: 100 pontos Regra: Caso solicitada a Carta Passo da Modelagem, a equipe ganha este emblema e a pontuação.</p>		<p>Valor: 10 pontos Regra: Caso solicitada a Carta Exemplos, a equipe ganha este emblema e a pontuação. Além disso, acumula as pontuações e o emblema desta carta e das cartas utilizadas anteriormente.</p>
	<p>Valor: 50 pontos Regra: Caso solicitada a Carta Perguntas-Guia, a equipe ganha este emblema e a pontuação. Além disso, a equipe acumula a pontuação e o emblema desta carta e das cartas utilizadas anteriores.</p>		<p>Valor: 50 pontos Regras: Esta medalha é dada a equipe que terminar primeiro a modelagem do diagrama de atividades, a partir do cenário dado pelo moderador.</p>
	<p>Valor: 20 pontos Regra: Caso solicitada a Carta Dicas, a equipe ganha mais este emblema e a pontuação. Além disso, a equipe acumula as pontuações e o emblema desta carta e das cartas utilizadas anteriormente.</p>		<p>Valor: 150 pontos Regras: Este troféu é dado a equipe que modelar o diagrama de forma correta e completa.</p>

Figura 4.13. Pontuação e Regras utilizadas no jogo.

No início do jogo, todos da equipe recebem uma descrição textual com o problema deverá ser modelado (cenário, especificação de casos de uso, dentre outros). Recomenda-se que

a equipe leia e discuta a descrição textual em conjunto para que todos obtenham o mesmo nível de entendimento sobre o problema. Após isso, os jogadores recebem as cartas e são informados que devem seguir as instruções de construção/ identificação que há em cada carta. Por exemplo, na carta Passo da Modelagem 1 (Identificação da *Swimlane*) há uma heurística que ajuda os jogadores a identificar quem são os atores envolvidos no cenário, então os jogadores começam a identificar na descrição textual os atores que podem ser as *swimlanes* do DA. Portanto, espera-se que seguindo as instruções das cartas em cada etapa do jogo, os jogadores identifiquem os elementos e modelem o diagrama. A versão completa do *ActGame* está disponível em um relatório técnico (Silva *et al.*, 2017b).

Com o objetivo de verificar a efetividade da aprendizagem, motivação e experiência do usuário ao utilizar o jogo *ActGame*, realizou-se uma avaliação do jogo utilizando o modelo MEEGA proposto por Savi *et al.* (2011).

4.4.2 Avaliação do Jogo *ActGame*

Savi *et al.* (2011) afirmam que os jogos propostos com o objetivo de apoiar ao processo de ensino e aprendizado devem ser avaliados para garantir que os mesmos tenham um efeito positivo no aprendizado do jogador. Desta forma, para a avaliação do aprendizado com o jogo *ActGame*, foi utilizado o modelo MEEGA (Savi *et al.*, 2011). A avaliação do jogo foi realizada com 29 estudantes voluntários do 5º período do Curso de Ciência da Computação da Universidade Federal do Amazonas. Antes da avaliação do jogo, todos os estudantes assistiram a um treinamento em que foi realizada uma breve contextualização sobre DA, além de uma contextualização sobre como seria realizada a avaliação. Após isso, os estudantes foram convidados a assinar um termo de consentimento, no qual concordavam participar do estudo. Apesar dos vinte e nove estudantes terem assinado o termo, apenas dezenove concordaram em responder o questionário de avaliação.

A avaliação ocorreu em duas etapas: (a) construção do diagrama utilizando o jogo; e (b) preenchimento do questionário de avaliação. Na primeira etapa, os estudantes foram divididos em equipes definidas por conveniência, em que eles decidiram quais estudantes fariam parte do seu grupo. Ao todo, foram formados seis grupos, com quatro ou cinco participantes por grupo. O tempo médio de utilização do jogo pelos participantes foi de 55 minutos (tempo mínimo 35 minutos; máximo 60 minutos). Isto indica que é viável a utilização do jogo durante uma aula. Na segunda etapa, dezenove estudantes responderam ao questionário do modelo MEEGA.

O MEEGA é um questionário padronizado proposto por Savi *et al.* (2011) que possibilita avaliar jogos educacionais. O questionário possui vinte e sete itens fixos divididos em três categorias (Motivação, Experiência do Usuário e Aprendizagem) e onze dimensões. Cada item do questionário é respondido em uma escala de Likert de cinco pontos com alternativas de resposta variando de discordo fortemente (-2) a concordo fortemente (2). Para avaliar o impacto na aprendizagem, o MEEGA ainda possui perguntas customizadas aos objetivos no jogo em que os estudantes realizam uma autoavaliação sobre o conteúdo antes e depois de utilizarem o jogo. Além disso, Foi acrescentada três questões abertas para permitir um melhor entendimento das respostas dos participantes em relação às categorias: (a) “Do que você gostou no jogo”; (b) “Do que você não gostou no jogo?”; e (c) “O que você mudaria para melhorar o jogo.”

4.4.2.1 Análise dos Resultados da Categoria Motivação do MEEGA

Com relação à categoria Motivação, avaliou-se as seguintes dimensões: Satisfação, Confiança, Relevância e Atenção. A Figura 4.14 mostra as afirmativas relacionadas a cada uma das dimensões e o grau de concordância com relação a cada afirmação.

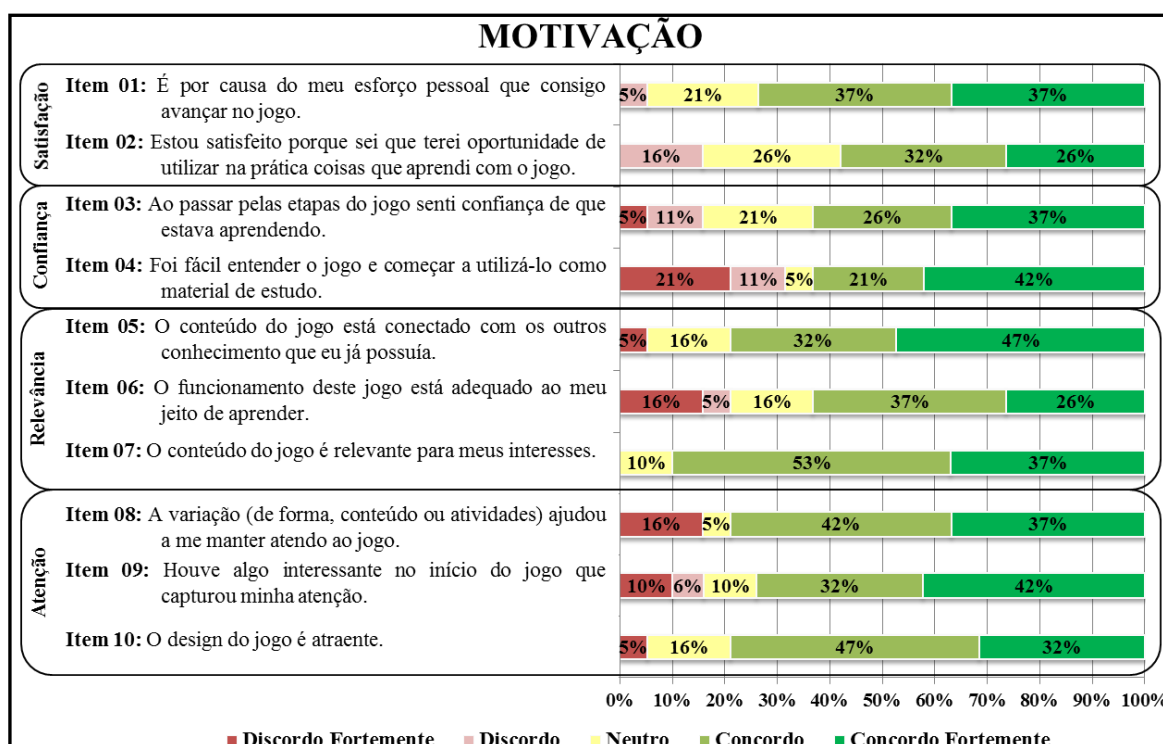


Figura 4.14. Gráfico de Avaliação da categoria Motivação.

No que diz respeito a dimensão Satisfação, 74% dos participantes concordaram que conseguiam avançar no jogo devido ao esforço pessoal. No item que avalia se o participante

teria oportunidade de utilizar na prática o que foi aprendido no jogo (item 02), houve concordância de 58% dos participantes. O participante P08 comentou que gostou de “*como o jogo faz você identificar primeiros as swimlanes e as atividades antes de aplicar na prática*” e o participante P01 frisou que “*percebi que todos os integrantes da equipe estavam motivados para solucionar o desafio.*” No que se refere a dimensão Confiança, obteve-se 63% de concordância em seus dois itens: o aprendizado obtido ao passar pelas etapas do jogo (item 03) e se o jogo foi fácil de entender e começar a jogar (item 04). Apesar de não haver muitos participantes discordando ou permanecendo neutro, percebe-se que esta dimensão possuiu a maior quantidade de discordâncias. Logo, este aspecto do jogo merece ser melhor investigado junto aos estudantes para buscar formas de melhorá-lo.

Sobre a dimensão Relevância, dois itens foram bem avaliados. O item que avalia se o conteúdo estava relacionado com outros conhecimentos que já possuíam (item 05), obteve concordância de 79% dos participantes. Enquanto que 63% dos participantes concordaram que o funcionamento do jogo é adequado. Além disso, 90% dos participantes concordaram que o conteúdo do jogo é importante para os interesses deles.

Por fim, quanto a dimensão Atenção, 79% dos participantes concordaram que a variação do jogo ajudou a mantê-los atentos e 74% dos participantes consideraram que houve algo que capturou a sua atenção ao jogo. Sobre o design do jogo, 79% consideraram que o design do jogo é atraente. O comentário do participante P12 reforça os resultados desta dimensão, pois, segundo ele, o jogo “*é uma atividade didática lúdica*”.

4.4.2.2 Análise dos Resultados da Categoria Experiência do Usuário do MEEGA

Nesta categoria foram avaliadas cinco dimensões: Competência, Diversão, Desafio, Interação Social e Imersão. A Figura 4.15 mostra as afirmações relacionadas a cada uma das dimensões e o grau de concordância com relação a cada afirmação. Com relação à dimensão Competência, 64% dos participantes tiveram sentimentos positivos de eficiência do jogo, enquanto que 84% afirmam que conseguiram atingir os objetivos do jogo por meio de suas habilidades.

Na dimensão Diversão, dois itens foram bem avaliados. Entretanto, em outros dois, nota-se uma grande quantidade de participantes discordando ou permanecendo neutros. Para o item que avalia se os participantes gostariam de utilizar o jogo novamente (item 13), houve concordância por parte de apenas 31% dos participantes. Apesar desta baixa concordância, 57% dos participantes afirmaram que recomendariam o jogo para seus colegas. Já o item que avalia

se os participantes ficaram desapontados com a interrupção do jogo (item 15), 37% dos participantes concordaram. No entanto, 69% dos participantes afirmaram que se divertiram com o jogo, tal fato foi ressaltado pelo participante P05, que gostou “*da maneira criativa que foi elaborada para aprendermos a modelagem*” do diagrama de atividades.

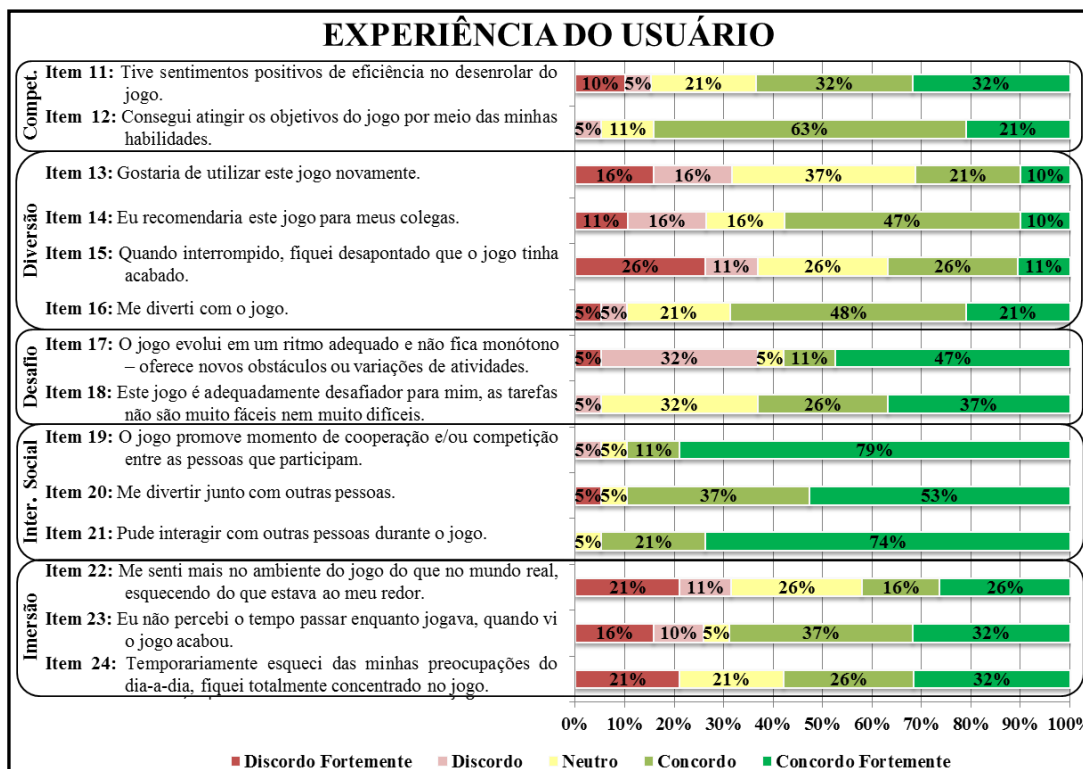


Figura 4.15. Gráfico de Avaliação da categoria Experiência do Usuário.

No que diz respeito à dimensão Desafio, 58% concordam que o jogo evolui em um ritmo adequado e não fica monótono, seja oferecendo obstáculos ou situações, conforme relatado no comentário do participante. Além disso, 63% afirmam que o jogo possui tarefas desafiadoras. Apesar deste item ter a maior porcentagem de discordâncias (32%) na dimensão Experiência do Usuário, o participante P09 avaliou positivamente o jogo: “*gostei da forma que o jogo fazia os integrantes interagirem para solucionar o problema junto com os obstáculos que apareciam no decorrer do jogo*”. Além disso, o participante P11 frisou que gostou do “*dinamismo e a agilidade do processo*” do jogo, avaliando positivamente a variação das atividades.

A dimensão Interação Social foi a melhor avaliada nesta categoria. Percebeu-se que 90% dos participantes concordaram em dois itens: no item que avalia se o jogo promove a cooperação e a competição entre as pessoas que participam (item 19) e no item que avalia se os estudantes se divertiam com os outros participantes enquanto utilizavam o jogo (item 20). Com

relação à interação com outras pessoas do grupo (item 21), houve uma concordância de 95% dos participantes. O participante P16 comentou que gostou da “*interação com o grupo e a discussão de ideias*” e o participante P03 frisou que gostou “*do forte espírito em equipe*” que o jogo proporciona. Além disso, o participante P17 afirmou que o jogo “*é dinâmico e envolve a percepção e a competitividade*” entre as equipes.

Para a dimensão Imersão, o item que avalia se o participante se sentiu mais no ambiente do jogo do que no mundo real, obteve uma concordância de apenas 42% (item 22). No entanto, 69% dos participantes não perceberam o tempo passar enquanto jogavam e 58% afirmam que esqueceram as preocupações enquanto estavam jogando.

4.4.2.3 Análise dos Resultados da Categoria Aprendizado do MEEGA

A Figura 4.16 apresenta o gráfico dos resultados da categoria Aprendizagem. Para essa categoria, houve uma concordância de 79% sobre o item que avalia se o jogo contribui para o desempenho profissional do participante (item 25). No item que avalia se o jogo foi eficiente para a aprendizagem dos participantes em comparação com outras atividades (item 26), houve uma concordância de 72%. Além disso, 79% dos participantes concordam que o jogo contribuiu para a aprendizagem na disciplina.

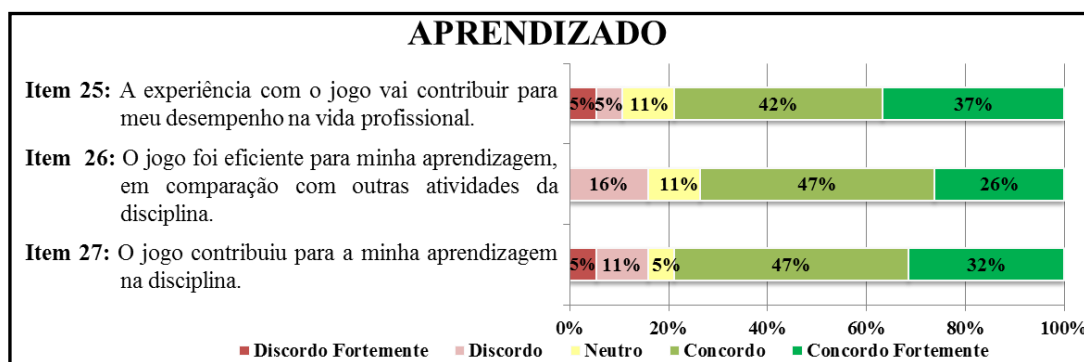


Figura 4.16. Gráfico de Avaliação da categoria Aprendizagem.

Esses resultados são reforçados pelos comentários dos participantes. O participante P02 afirmou que “*(o jogo) proporcionou ótimo aprendizado*”. Já o participante P18 disse que “*as etapas do jogo facilitam no aprendizado*” e o participante P10 frisou que o jogo é “*uma alternativa ao padrão aula com slides. (o jogo é) bem criativo, conta com vários conceitos úteis para o desenvolvimento do conhecimento*”.

Savi *et al.* (2011) sugerem avaliar também a aprendizagem dos estudantes através da autoavaliação em relação ao seu aprendizado sobre o conteúdo que foi ensinado. Os conceitos avaliados estavam relacionados aos elementos do DA e foram avaliados do ponto de vista de

três objetivos, antes e depois de utilizar o jogo: (a) lembrar o que é, (b) compreender como funciona e (c) aplicar na prática. Os estudantes deveriam atribuir uma nota de 1 a 5, onde 1 estava relacionado a “pouco conhecimento” e 5 a “muito conhecimento”.

A Figura 4.17 apresenta o gráfico com as médias de autoavaliação da aprendizagem dos estudantes em relação aos elementos do DA, antes e depois do jogo. É possível perceber que houve um aumento no nível de conhecimento dos estudantes para todos os elementos do DA. Este resultado é positivo, pois indica que o jogo alcançou seus objetivos em relação ao auxílio na aprendizagem de DA.

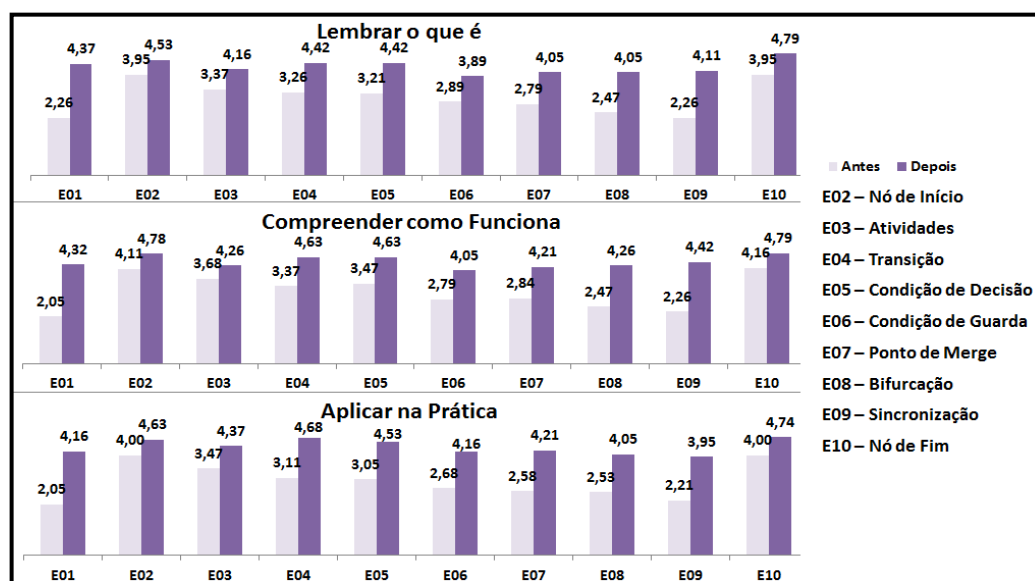


Figura 4.17. Gráfico das medidas de autoavaliação.

4.4.3 Considerações sobre o estudo

Esta subseção apresentou o jogo *ActGame (Activity Diagram Game)*, baseado em cartas, que tem como objetivo auxiliar no processo de ensino e aprendizagem de Diagrama de Atividades. Foi realizada também uma avaliação experimental inicial utilizando o modelo MEEGA, proposto por Savi *et al.* (2011). Através dos resultados da avaliação do jogo, no que diz respeito à categoria Motivação, percebeu-se que o jogo teve efeito positivo. O design do jogo foi considerado atraente, o conteúdo relevante e conectado a outros conhecimentos, além de ter ajudado a manter os estudantes atentos ao jogo. Em relação à categoria Experiência do Usuário, os resultados também mostraram que o jogo proporcionou uma experiência de uso relativamente positiva aos estudantes, com destaque para a dimensão interação social. Além disso, os resultados mostraram que há indícios sobre a utilidade do jogo para a aprendizagem de DA. Este resultado também foi percebido na média das concordâncias dos participantes com relação à categoria Aprendizagem e na autoavaliação realizada pelos estudantes. Apesar disso,

observou-se também pontos de melhorias no jogo, tais como, melhorar o jogo para que os estudantes se divirtam e se sintam imersos enquanto estiverem jogando e fazer com que os estudantes se sintam seguros que estão modelando um DA de forma correto.

Savi *et al.* (2011) afirmam que os jogos propostos com o objetivo de apoiar ao processo de ensino e aprendizado devem ser avaliados para garantir que os mesmos tenham um efeito positivo no aprendizado do jogador. Desta forma, para a avaliação do aprendizado com o jogo *ActGame*, foi utilizado o modelo MEEGA (Savi *et al.*, 2011). A avaliação do jogo foi realizada com 29 estudantes voluntários do 5º período do Curso de Ciência da Computação da Universidade Federal do Amazonas. Antes da avaliação do jogo, todos os estudantes assistiram a um treinamento em que foi realizada uma breve contextualização sobre DA, além de uma contextualização sobre como seria realizada a avaliação. Após isso, os estudantes foram convidados a assinar um termo de consentimento, no qual concordavam participar do estudo. Apesar dos vinte e nove estudantes terem assinado o termo, apenas dezenove concordaram em responder o questionário de avaliação.

4.5 SÍNTESE DO CAPÍTULO

Neste capítulo foi apresentado estudos experimentais exploratórios realizados com o objetivo de avaliar os efeitos de uso de diferentes estratégias de ensino durante o processo de aprendizagem de modelagem de software utilizando os diagramas da UML. A partir dos resultados dos estudos, percebeu-se que os professores podem ir além das abordagens de ensino tradicionais que são comumente utilizadas em sala de aula. Os professores podem empregar outros métodos de ensino / recursos educacionais como forma de melhorar o processo de ensino e aprendizagem de modelagem de software. Ao empregar estes novos métodos de ensino, o professor minimiza a lacuna entre os conceitos ensinados em sala de aulas e as habilidades necessárias pela indústria de software, pois incentiva os estudantes a trabalhar em grupo ou sozinho— em momentos que são necessários e adequados, conforme visto na Subseção 4.2.4—, a discutir os diversos pontos de vistas sobre o problema, por exemplo. Desta forma, este capítulo apresentou resultados que visam contribuir para a comunidade de Educação em Engenharia de Software, mostrando que é de suma relevância investigar o uso de novos métodos de ensino como forma de apoio no ensino de modelagem de software. Logo, há a necessidade de uma investigação mais profunda sobre a viabilidade de uso destes novos métodos/ estratégias de ensino, levando em consideração as percepções dos estudantes e professores de diferentes contextos de ensino.

CAPÍTULO 5 – CICLO DE RELEVÂNCIA (PARTE III) - ESTUDO DE VIABILIDADE SOBRE O USO DAS ESTRATÉGIAS DE APRENDIZAGEM ATIVA A PARTIR DAS PERSPECTIVAS DOS ESTUDANTES E PROFESSORES

Este capítulo apresenta os resultados do terceiro ciclo de relevância. Neste ciclo, foi conduzido um estudo de viabilidade realizado com o objetivo de verificar se a adoção de estratégias de aprendizagem ativa no contexto de diversas instituições de ensino é viável de ser aplicada. Para isso, analisou-se as percepções dos estudantes e professores em relação às estratégias de aprendizagem ativa utilizadas para apoiar o ensino e aprendizagem de modelagem de software, empregando os diagramas da UML.

5.1 INTRODUÇÃO

Os resultados obtidos a partir dos ciclos de relevância anteriores (Capítulo 3 e Capítulo 4) geraram discussões sobre o uso das estratégias empregadas pelos professores. Essas estratégias estão fortemente vinculadas à modelagem do ensino, pois permitem que os estudantes (a) elaborem várias soluções para um determinado problema (Schilling e Sebern, 2012; Knobloch *et al.*, 2018); (b) melhorem as habilidades de trabalho em equipe (Knobloch *et al.*, 2018; Nunes *et al.*, 2015); (c) melhorem as habilidades de comunicação oral e escrita usadas para modelar os diagramas (Knobloch *et al.*, 2018; Nunes *et al.*, 2015). No entanto, poucos trabalhos descrevem o uso dessas estratégias no ensino de modelagem de software (Silva *et al.*, 2019). O ensino da modelagem de software é importante porque ajuda os alunos a desenvolver uma mentalidade de pensamento abstrato e a compreender o conceito de design dos componentes (objetos) de um sistema (Hadar e Hadar, 2007). Além disso, após o uso das estratégias utilizadas nos estudos apresentados no capítulo anterior (Capítulo 4), os professores comentaram em uma discussão informal que tiveram diversas dificuldades em empregá-las devido aos diversos desafios enfrentados para implementá-las em sala de aula.

Tendo em vista os resultados formais e informais obtidos nos estudos investigativos do Ciclo de Relevância I, realizou-se uma categorização das tecnologias de ensino identificadas no Mapeamento Sistemático da Literatura. A partir do Mapeamento Sistemático, foram selecionadas estratégias de ensino que possuem qualquer princípio de aprendizagem ativa em

seu referencial teórico, são elas: *Inspection-Based strategy*, *Think-Pair-Square*, *Positive Examples*, *Negative Examples*, *Learning based on similar systems*.

Desta forma, um estudo experimental foi realizado de verificar a viabilidade de uso das estratégias ativas de aprendizagem identificadas no Mapeamento Sistemático da Literatura. Para isso, analisou-se as percepções dos estudantes e professores em relação às estratégias de aprendizagem ativa para apoiar o processo de ensino e aprendizagem dos diagramas da UML. Para guiar nesse objetivo, foram definidas as seguintes perguntas de pesquisa:

- **QP1:** Como os estudantes de graduação percebem as estratégias de aprendizagem ativa empregadas durante o ensino de diagramas UML?
- **QP2:** Quais desafios os professores percebem ao usar estratégias de aprendizagem ativas enquanto ensinam diagramas da UML?

Para responder à questão de pesquisa QP1, foram analisados qualitativamente os dados coletados de questionários e grupo focal (Kontio *et al.*, 2004) realizados em quatro turmas de duas universidades diferentes do Brasil: Universidade Federal do Amazonas (UFAM) e Universidade Estadual de Maringá (UEM). Isso permitiu entender melhor as dificuldades e os benefícios percebidos pelos estudantes ao usar essas estratégias. Para responder à RQ2, foram conduzidas entrevistas semiestruturadas com professores para investigar os desafios encontrados ao incorporar estratégias de aprendizagem ativa em sala de aula. Todos os dados foram analisados usando procedimentos de codificação (Strauss e Corbin, 1998).

5.2 MÉTODO DE PESQUISA

Foi conduzido um estudo experimental qualitativo com o objetivo de analisar as percepções de professores e estudantes sobre o uso dessas estratégias. A compreensão de quais fatores impedem e/ou ajudam os estudantes a alcançar metas de aprendizado, podem ajudar os professores a entender como essas estratégias podem ser usadas no contexto do ensino de modelagem de software. Para atingir as questões de pesquisa acima citadas, seguiu-se o método apresentado na Figura 5.1 e explicado nesta seção.

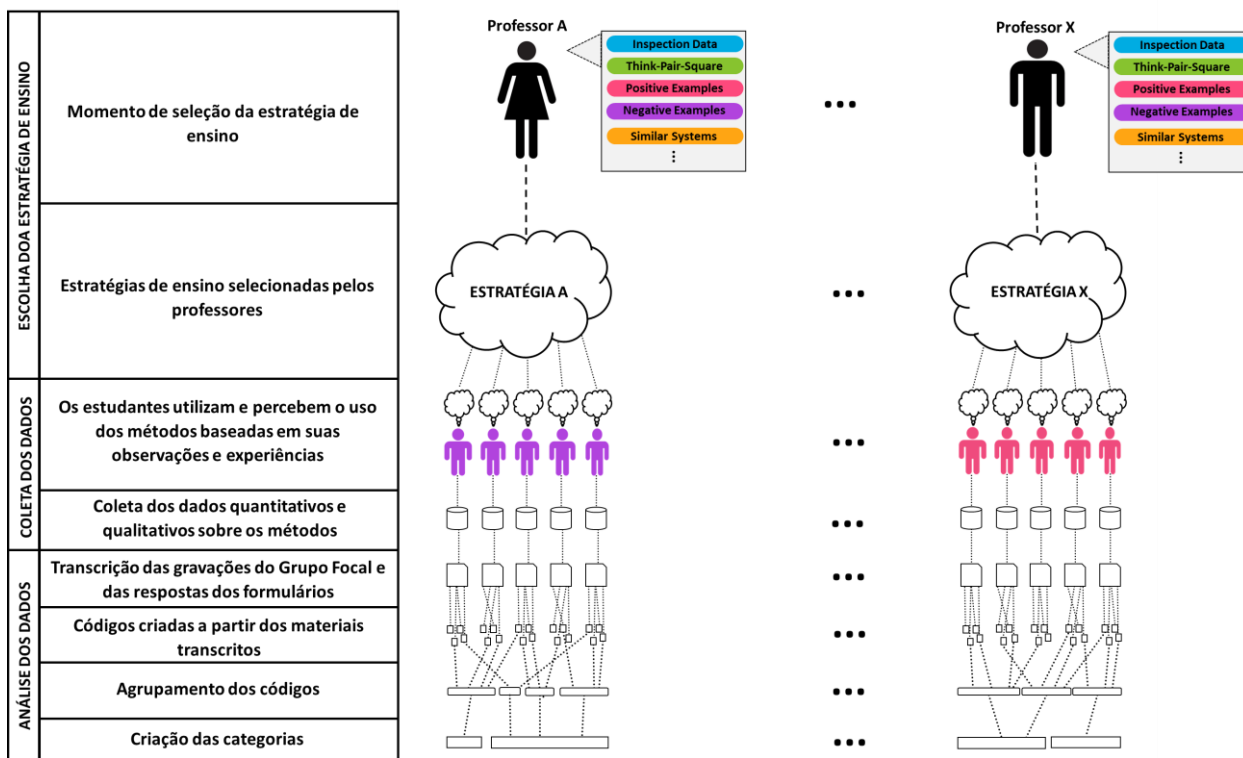


Figura 5.1. Visão geral do método de pesquisa.

5.2.1 Participantes

O estudo foi realizado em duas universidades brasileiras: Universidade Estadual de Maringá (UEM) e Universidade Federal do Amazonas (UFAM). O estudo envolveu quatro professores e 100 estudantes. A Tabela 5.1 apresenta uma visão geral dos participantes deste estudo.

Tabela 5.1. Informações sobre as turmas.

Universidade	ID do professor	Curso	Nome da disciplina	Estratégia escolhida	Diagrama UML	# estudantes
UFAM (Turma A)	I01	Sistemas de Informação	Análise e Projeto de Sistemas	Inspection-Based strategy	Diagrama de classes	16
				Think-Pair-Square	Diagrama de Sequência	
UFAM (Turma B)	I02	Ciência da Computação	Introdução à Engenharia de Software	Positive Examples	Diagrama de UC	28
				Negative Examples	Descrição de UC	
				Learning based on similar systems	Diagrama e Descrição textual de UC	
UEM (Turma C)	I03	Ciência da Computação	Engenharia de Software	Negative Examples	Class diagram	35
UFAM (Turma D)	I04	Ciência da Computação	Análise e Projeto de Sistemas	Inspection-Based strategy	Sequence Diagram	21

Em relação às Turmas A e B, as aulas ocorreram no primeiro semestre de 2018 na UFAM, em Manaus-AM, Brasil. Ambos as disciplinas fazem parte do 5º semestre do curso e é o primeiro contato dos estudantes com a modelagem de software. Nestas aulas, os professores

apresentaram os conceitos sobre modelagem de Casos de Uso (UCs): o diagrama e a descrição textual. Depois disso, uma tarefa foi aplicada aos estudantes para exercitar um pouco mais sobre os conceitos ensinados.

As aulas das turmas C e D ocorreram no segundo semestre de 2018. Na Turma C (UEM), os estudantes também tiveram seu primeiro contato com modelagem de software na disciplina de Engenharia de Software. Na Turma D, os estudantes já haviam aprendido modelagem de software e aplicado os conceitos aprendidos em um trabalho prático em um curso anterior. Essas aulas se concentraram em: (a) apresentar os fundamentos da modelagem de software, bem como os conceitos de análise e *design* orientados a objetos; (b) praticar os conceitos usando UML. Para identificar os estudantes neste estudo, criou-se um identificador, PXXC, no qual XX é um número exclusivo para um estudante e C representa a turma à qual eles pertencem. Por exemplo, P01A refere-se ao estudante 01 da Turma A.

Todos os professores têm mais de doze anos de experiência na indústria de software e mais de sete anos de experiência no ensino de modelagem de software. Com relação às estratégias de aprendizagem ativa, todos os professores já tinham empregado estratégias de aprendizagem ativa, por exemplo, Aprendizagem Baseada em Problemas, Aprendizagem Baseada em Projetos, Sala de Aula Invertida e outros. No entanto, os professores não conheciam as estratégias escolhidas para a condução do estudo.

É importante enfatizar que os estudantes forneceram voluntariamente os dados para a pesquisa. Quando os professores forneceram o termo de consentimento, os estudantes foram informados de que podiam optar por não participar a qualquer momento, mas deveriam realizar a tarefa, uma vez que já fazia parte do cronograma da disciplina.

5.2.2 Coleção dos dados

Para comparar as estratégias, aplicou-se questionários pós-modelagem com o objetivo de avaliar o aprendizado percebido pelos estudantes. Os estudantes forneceram suas respostas em uma escala Likert de 5 pontos, com opções que variam de “Eu discordo totalmente” (-2) a “Eu concordo totalmente” (2), com uma opção neutra.

Os itens do questionário, apresentados na Tabela 5.2, foram baseados nas dimensões da aprendizagem (Wangenheim *et al.*, 2013; Sindre e Moody, 2003). Os itens 01 e 02 visam avaliar se as estratégias contribuíram para os resultados da aprendizagem do curso; os itens 03 a 08 foram projetados de acordo com os níveis de aprendizado de Taxonomia de Bloom (Anderson

et al., 2001); e os itens 09 a 18 tiveram como objetivo avaliar a atitude positiva geral em relação às estratégias (Ekahitanond, 2013; Orawiwatnakul e Wichadee, 2016).

Tabela 5.2. Itens avaliados em questionários pós-modelo.

	Item	Descrição do Item
Dimensão Factual da Taxonomia de Bloom	Item 01	A estratégia contribuiu para a minha aprendizagem na disciplina.
	Item 02	A estratégia foi eficiente para minha aprendizagem, em comparação com outras atividades da disciplina.
	Item 03	A estratégia auxiliou a lembrar os conceitos aprendidos sobre diagrama.
	Item 04	A estratégia contribuiu para compreender como os conceitos aprendidos podem ser utilizados na modelagem do diagrama.
	Item 05	A estratégia contribuiu para aplicar dos conceitos do diagrama durante a resolução dos problemas.
	Item 06	A estratégia contribuiu para organizar o diagrama durante a modelagem.
	Item 07	A estratégia contribuiu para verificar se o diagrama foi modelado corretamente.
	Item 08	A estratégia contribuiu para criar o diagrama durante a modelagem.
Atitudes dos estudantes sobre as estratégias	Item 09	Com o uso da estratégia, me senti mais motivado para aprender que habitualmente.
	Item 10	A estratégia me permitiu melhorar minhas opiniões críticas.
	Item 11	Eu me senti mais conectado com os outros durante a aplicação da estratégia.
	Item 12	O método me fez sentir parte da aula.
	Item 13	Com esta estratégia, pude expressar minhas opiniões livremente.
	Item 14	Com a estratégia, descobri as falhas no que anteriormente acreditei estar certo.
	Item 15	Achei a estratégia muito útil para a minha aprendizagem.
	Item 16	A estratégia facilitou que eu participasse mais ativamente da minha aprendizagem.
	Item 17	A estratégia ajudou a aumentar minhas habilidades de pensamento crítico.
	Item 18	A estratégia me forneceu uma boa experiência ao aprender a modelar o diagrama.

Após responderem o questionário, todos os estudantes de cada turma foram convidados a participarem de uma sessão de Grupo Focal. Todos os estudantes concordaram em participar. A configuração do grupo focal permite investigar um tópico específico, mediado por um moderador, no qual os participantes fornecem suas respostas e discutem as idéias de outros, enriquecendo as informações obtidas (Kontio *et al.*, 2004).

A sessão do Grupo Focal foi dividida em três etapas. No primeiro passo, para incentivar os estudantes a apresentarem suas opiniões negativas, usou-se uma dinâmica de *Lovers e Haters* (Colucci, 2008), ver Figura 5.2. Nessa dinâmica, os estudantes foram divididos em equipes, e cada equipe tem um papel predefinido durante a discussão (veja a), na qual os *Lovers* devem argumentar a favor das estratégias e os *Haters* contra eles. No início da sessão, os papéis de cada equipe (*Lovers* ou *Haters*) foram definidos de forma aleatória.

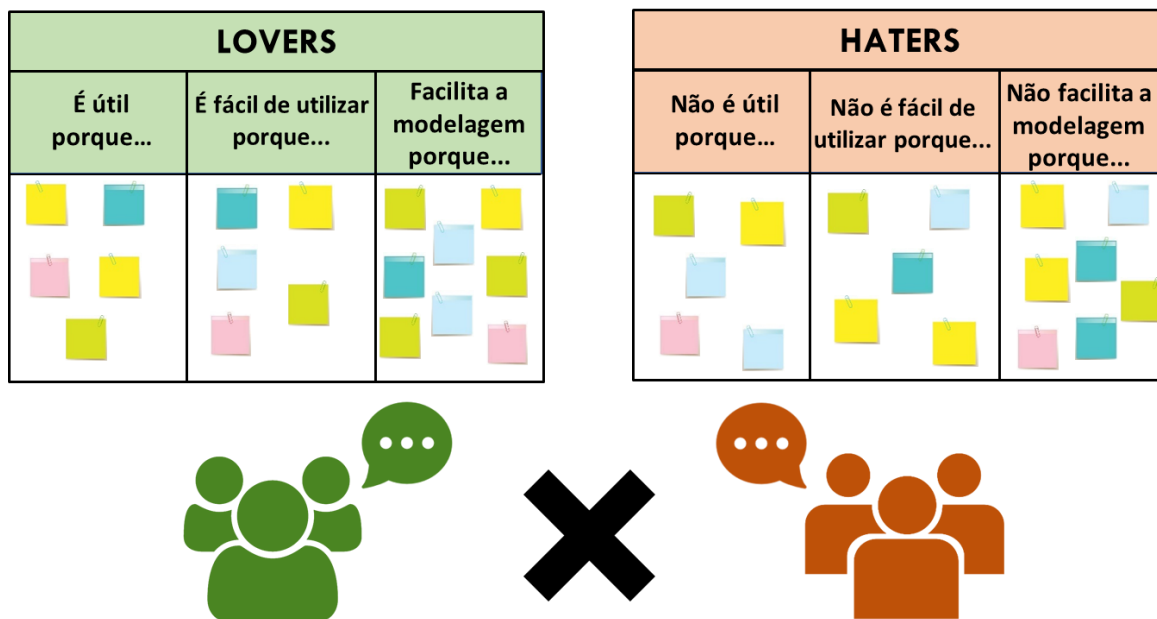


Figura 5.2. Quadro criado para orientar a discussão do Grupo Focal do tipo *Lovers X Haters*.

Depois disso, (segundo passo) cada estudante usou notas adesivas coloridas para escrever as dificuldades, facilidades, pontos positivos, negativos ou dúvidas que eles tinham ao criar o diagrama com a estratégia. Em seguida (terceira etapa), cada estudante de cada equipe colocou suas anotações no branco (veja a Figura 5.3) e as explicou. O moderador pediu aos outros estudantes que comentassem se concordavam ou não com a opinião do colega. Cada sessão durou 1h30 e todas as informações desta sessão foram gravadas em vídeo pelo moderador, com o consentimento dos participantes presentes.



Figura 5.3. Sessão do Grupo Focal.

Depois que os professores empregaram as estratégias, foram realizadas entrevistas semiestruturadas, que consistiam em uma mistura de perguntas abertas e específicas, projetadas para extrair tipos de informações antecipadas e inesperadas (Seaman, 1999). Nesse tipo de

entrevista, as perguntas são planejadas e procura-se respondê-las, mas não são necessariamente feitas da mesma maneira ou na mesma ordem em que são listadas (Runeson e Host, 2009). Criou-se também o roteiro de entrevistas de acordo com as recomendações da literatura (Seaman, 1999; Runeson e Host, 2009). Para atender aos requisitos éticos da pesquisa, o objetivo da pesquisa foi explicado a cada professor e foi aplicado um termo de consentimento, garantindo a confidencialidade dos dados fornecidos e o anonimato do professor. As entrevistas foram realizadas individualmente, dentro do tempo esperado (entre 15 e 30 minutos).

Os pesquisadores não se apressaram na entrevista e o entrevistado conseguiu se expressar de maneira calma e ininterrupta. Durante as entrevistas, o professor pôde expressar suas percepções sobre o uso das estratégias durante o ensino de modelagem, bem como os desafios enfrentados na aplicação dessas estratégias.

5.2.3 Análise de dados

Analisou-se qualitativamente a transcrição das respostas seguindo os procedimentos de codificação (Strauss e Corbin, 1998). O objetivo da análise qualitativa foi codificar, categorizar e sintetizar dados, a fim de identificar as dificuldades e os benefícios percebidos pelos estudantes e professores após o uso das estratégias. Inicialmente, todo o áudio dos grupos focais foi transcrito literalmente. Utilizou-se a ferramenta Atlas.ti² para apoiar a análise e síntese de dados. A análise dos dados começou com a codificação aberta das transcrições. O processo de codificação utilizado nesse estudo foi representado na Figura 5.4. A análise foi conduzida por um pesquisador e discutida com os outros pesquisadores em várias reuniões. Isso foi realizado para mitigar qualquer viés potencial no processo de codificação. As subseções a seguir apresentam mais detalhes sobre os resultados qualitativos.

² Atlas.ti – The Knowledge Workbench, Scientific Software Development – <http://www.atlasti.com>

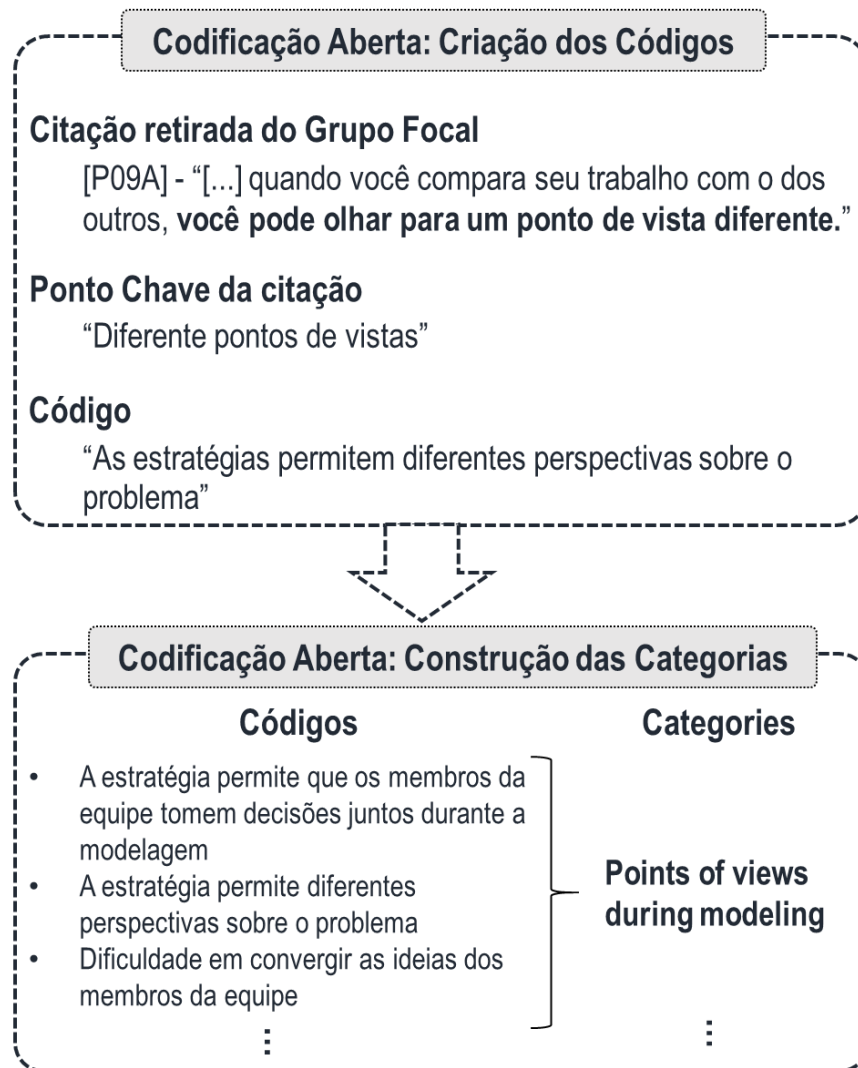


Figura 5.4. Exemplo de codificação.

5.3 RESULTADOS

Nesta subseção serão apresentados os resultados de acordo com as perguntas da pesquisa.

5.3.1 QP1: Como os estudantes de graduação percebem as estratégias de aprendizado ativas empregadas durante o ensino de diagramas UML?

Para responder a QP1, primeiro foram analisadas as respostas dos estudantes no questionário sobre as percepções das estratégias empregadas. Depois disso, foram analisados os comentários dos estudantes durante a sessão do Grupo Focal e foram identificados fatores que influenciaram positiva ou negativamente o uso das estratégias.

5.3.1.1 Percepção dos estudantes sobre as estratégias

A Figura 5.5 e a Figura 5.6 apresentam as percepções dos estudantes sobre sua experiência de aprendizado ao usar estratégias ativas. As percepções dos estudantes foram agrupadas sobre cada uma das estratégias.

Em relação à “dimensão factual” da taxonomia de Bloom (itens 3 a 8), a maioria dos estudantes concordou que as estratégias de aprendizado contribuíram positivamente para o aprendizado. Para o item que avaliou se as estratégias de aprendizagem contribuíram para a aprendizagem (item 01), foi obtido um bom nível de concordância para 4 das 5 estratégias (91,7% de concordância para a *Inspection-Based strategy*, 81,1% de concordância para *Think-Pair-Square*, 76,2 % de concordância para *Negative Examples*, 60,70% de concordância para *Positive Examples*). O único resultado contrastante foi para a estratégia *Similar Systems*, para o qual apenas 46,4% dos estudantes concordaram. A estratégia *Similar Systems* obteve o pior desempenho, dentre todas as estratégias, para o item que avalia se as estratégias de aprendizagem ajudam na modelagem correta de um diagrama (item 07), com 21,4% de discordância. Essa discordância pode estar relacionada ao fato de que essa estratégia deve ter sido usada apenas para criar diagramas. No geral, os resultados mostram que o uso dessas estratégias de aprendizagem promoveu uma boa percepção para os estudantes, principalmente em termos de lembrança, interpretação e aplicação.

Com relação às atitudes, a *Inspection-Based strategy* evidenciou os mais altos níveis de concordância. Percebeu-se que mais de 90% dos participantes concordaram que a estratégia era útil para a aprendizagem (item 15) e que eles (estudantes) participaram mais ativamente (item 16). No entanto, notou-se que no item 09 (motivação para aprender), um grande número de estudantes permaneceu neutro (44,4%). Apesar disso, de acordo com a percepção dos estudantes, essas estratégias contribuíram para o aprendizado dos diagramas. Em relação à estratégia de *Similar Systems*, notou-se um grande número de participantes discordando ou mantendo-se neutros quando perguntados se a estratégia os motivava a aprender (item 9); e se eles (os estudantes) se sentiam mais conectados aos colegas (item 11). Além disso, mais de 50% dos estudantes permaneceram neutros porque não tiveram uma boa experiência em aprender a modelar usando essa estratégia (item 18). Isso mostra que a estratégia de *Similar Systems* não se destacou como mais eficiente na entender os conceitos de diagramas.

Dimensão Factual da Taxonomia de Bloom

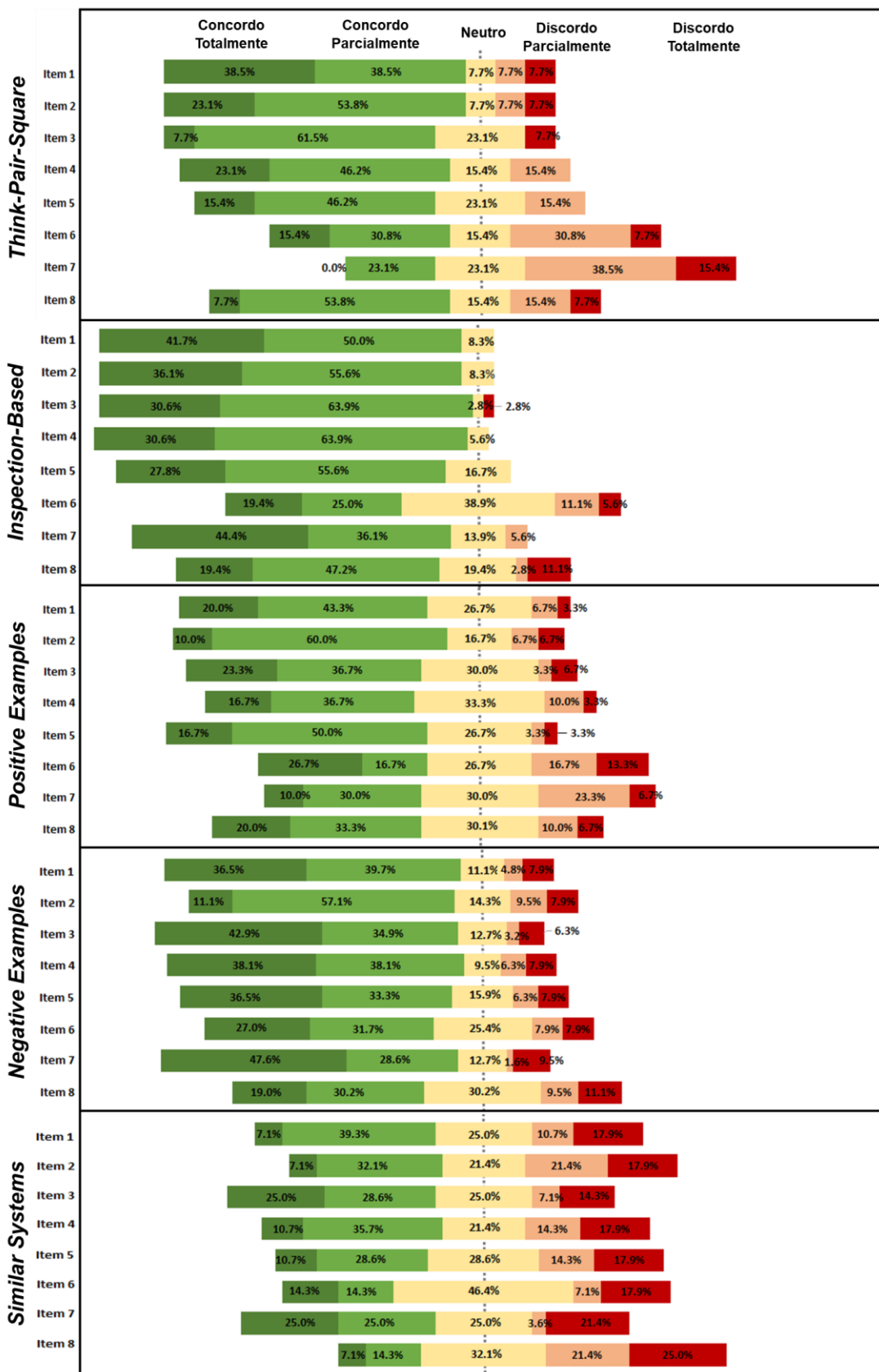


Figura 5.5. Percepções dos estudantes sobre as estratégias (1/2).

Atitudes dos estudantes sobre as estratégias

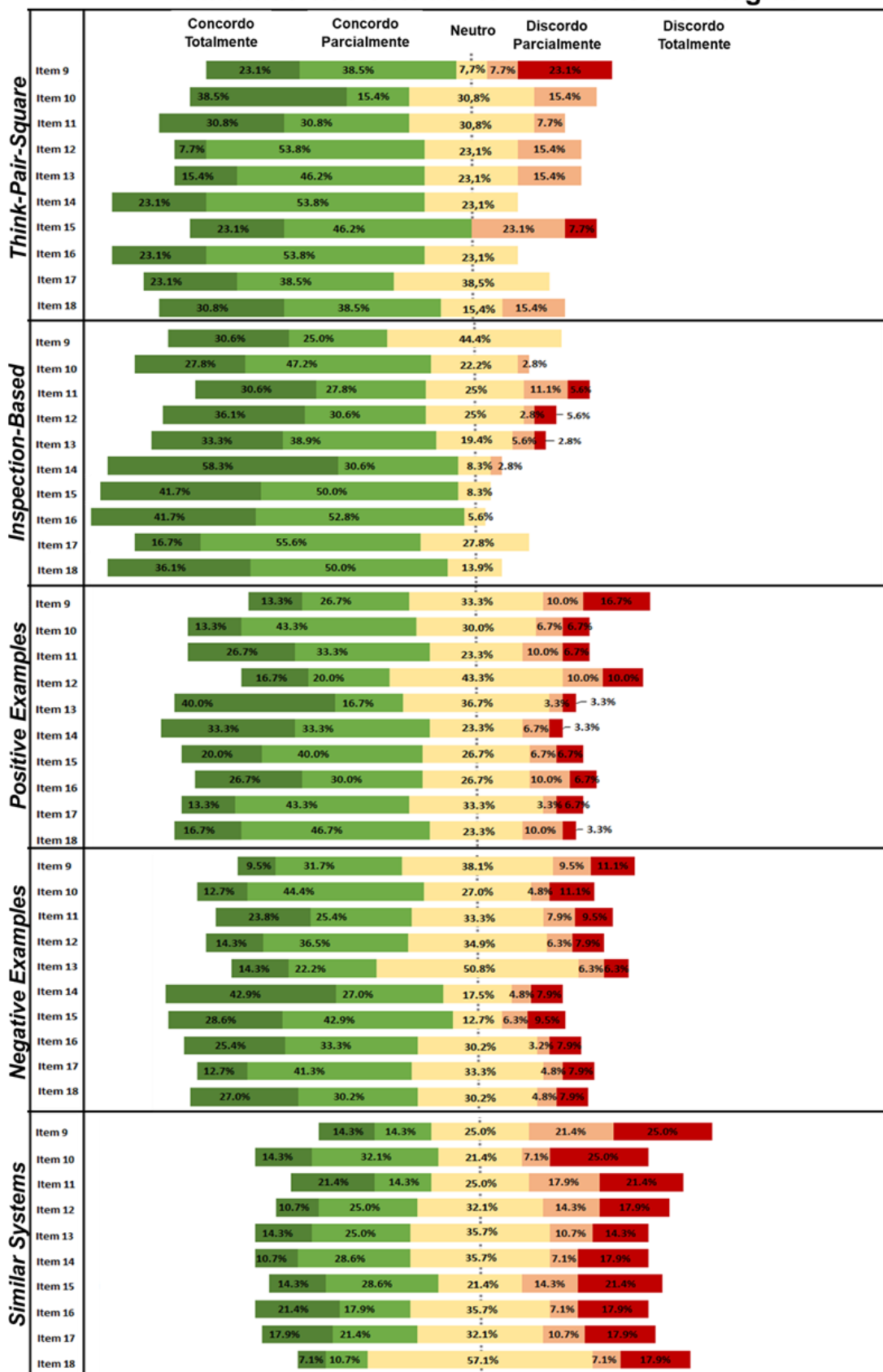


Figura 5.6. Percepções dos estudantes sobre as estratégias (2/2).

5.3.1.2 Resultados qualitativos sobre a percepção dos estudantes sobre as estratégias

Ainda sobre as percepções dos estudantes, analisou-se qualitativamente seus comentários durante a sessão do grupo focal. Encontrou-se cinco fatores principais identificados no estudo qualitativo sobre estratégias de aprendizado ativo (ver Tabela 5.3).

Tabela 5.3. Fatores e subfatores identificados no estudo qualitativo.

Fatores	Subfatores	Estratégias de Aprendizagem Ativa				
		<i>Think-Pair-Square</i>	<i>Inspection Based strategy</i>	<i>Positive Examples</i>	<i>Negative Examples</i>	<i>Similar Systems</i>
Ponto de vista dos estudantes durante a modelagem	(+) A estratégia permite que os membros da equipe tomem decisões juntos durante a modelagem	X				
	(+) A estratégia permite diferentes perspectivas sobre o problema	X	X			
	(-) Dificuldade em convergir as ideias dos membros da equipe	X	X			
Aprendizagem colaborativa	(+) A colaboração tornou possível criar um diagrama com mais detalhes	X				
	(+) A interação com a equipe melhorou o aprendizado	X	X			
	(+) As estratégias proporcionaram a oportunidade de aprender a trabalhar em equipe		X	X		
Apoio durante a modelagem	(+) A estratégia guia os estudantes durante a modelagem			X	X	
	(+) A estratégia ajuda os estudantes a identificar defeitos durante a modelagem			X	X	
	(-) A estratégia não apoia estudantes durante a modelagem			X	X	
	(-) Falta de moderador nos grupos para a tomada de decisão	X	X			
A estratégia permite o estudante receber o <i>feedback</i>	(+) A estratégia possibilita receber <i>feedback</i> dos problemas identificados pela outra equipe		X			
	(-) Falta de comunicação/interação entre equipes (modeladores e inspetores)		X			
Abstração e resolução de problemas	(+) A estratégia permite abstrair melhor o problema					X
	(+) A estratégia integra teoria com prática					X
	(-) A estratégia não permite abstrair novos cenários					X
Legenda:						
(+)						
Trata-se dos benefícios percebidos pelos estudantes após o uso das estratégias ativas de aprendizagem.						
(-)						
Trata-se das dificuldades percebidas pelos estudantes após o uso das estratégias ativas de aprendizagem.						

Pontos de vista durante a modelagem (primeiro fator): ao analisar os dados, percebeu-se que as estratégias influenciavam diretamente as perspectivas dos estudantes durante a modelagem da solução. Esse fator está relacionado às capacidades dos estudantes de trabalhar em equipe, nas quais existem diferentes perspectivas e em que os membros da equipe expressam seu ponto de vista sobre o problema durante a modelagem. Esse fator foi composto por benefícios (a e b) e dificuldades (c) percebidas pelos estudantes (ver na Tabela 5.4): (a) a estratégia permite que os membros da equipe tomem decisões juntos durante a modelagem; (b) a estratégia permite diferentes perspectivas sobre o problema, e (c) dificuldade em convergir as ideias dos membros da equipe. Nas tabelas serão destacados e apresentados esses subfatores.

Tabela 5.4. Primeiro fator: Pontos de vista durante a modelagem.

<p>A estratégia permite que os membros da equipe tomem decisões juntos durante a modelagem (evidenciado por 2 estudantes)</p> <p><i>Think-Pair-Square:</i> “[...] primeiro, pegamos todos os pontos de todos os diagramas e conversamos, depois tivemos as melhores ideias [...] e (durante a modelagem) as pessoas trabalharam para colocar as ideias de todos na versão final do diagrama.” – Estudante P17A</p>
<p>A estratégia permite diferentes perspectivas sobre o problema (evidenciado por 12 estudantes)</p> <p><i>Think-Pair-Square (3 de 12):</i> “Eu pensei que a estratégia era muito útil porque em cada estágio parece que estamos fazendo um diagrama mais completo a cada passo que melhoramos, no final, teremos um diagrama mais completo.” Estudante P10A</p> <p><i>Inspection-Based strategy (8 de 12):</i> “[...]à medida que criamos nossos modelos e a outra equipe dá feedback, nós não criamos um modelo com base em nossa própria opinião.” – Estudante P03D</p>
<p>Dificuldade em convergir as ideias dos membros da equipe (evidenciado por 8 estudantes)</p> <p><i>Think-Pair-Square (7 de 8):</i> “Eu tinha muitos diagramas para comparar, isso me confundiu e trouxe mais dúvidas do que me ajudou.” – Estudante P01A</p> <p><i>Inspection-Based strategy (1 de 8):</i> “passamos muito tempo tentando conciliar opiniões diferentes dentro da própria equipe.” – Estudante P08D</p>

Aprendizagem colaborativa (segundo fator): as estratégias ajudaram os estudantes que tiveram dificuldades em modelar a aprender com outros estudantes mais experientes, melhorando a qualidade da colaboração e incentivando a comunicação e a contribuição ativa dos estudantes. Sobre essa categoria, foram identificados três benefícios do uso de estratégias (Tabela 5.5): (a) a colaboração tornou possível criar um diagrama com mais detalhes; (b) a interação com a equipe melhorou o aprendizado; e (c) as estratégias proporcionaram a oportunidade de aprender a trabalhar em equipe.

Tabela 5.5. Segundo fator: Aprendizagem colaborativa.

<p>A colaboração tornou possível criar um diagrama com mais detalhes (evidenciado por 4 estudantes)</p> <p><i>Think-Pair-Square:</i> “a colaboração foi ótima, porque eu pude fazer um diagrama mais detalhado.” – Estudante P04A</p>
<p>A interação com a equipe melhorou o aprendizado (evidenciado por 8 estudantes)</p> <p><i>Think-Pair-Square (4 de 8):</i> “apesar de ter perdido as aulas anteriores, consegui assimilar o conteúdo e a ideia, e participar da modelagem.” Estudante P05A</p> <p><i>Inspection-Based strategy (5 de 8):</i> “como eu não havia criado o diagrama na última aula, estava meio perdido. Quando comecei a modelar com a equipe, os colegas estavam me explicando e aprendi rapidamente e até consegui ajudar.” – Estudante P16D</p>
<p>As estratégias proporcionaram a oportunidade de aprender a trabalhar em equipe (evidenciado por 7 estudantes)</p> <p><i>Positive Examples (5 de 7):</i> “[A estratégia ajudou] a continuar aprendendo por causa da oportunidade de trabalhar em equipe e ouvir opiniões diferentes.” – Estudante P08B</p> <p><i>Inspection-Based strategy (3 de 7):</i> “[...] existem diferentes maneiras de trabalhar, mas trabalhar em equipe é apenas aprender a ceder e ouvir a opinião dos outros. Eu estava pensando em fazer o trabalho de maneira diferente e meu colega já havia decidido outra maneira de trabalhar, mas depois acertamos.” – Estudante P11A</p>

Apoio durante a modelagem (terceiro fator): Os estudantes também relataram pontos positivos e negativos relacionados ao apoio fornecido por algumas das estratégias durante a modelagem (ver Tabela 5.6). Esta categoria incluiu quatro fatores: (a) a estratégia guia os estudantes durante a modelagem; (b) a estratégia ajuda os estudantes a identificar defeitos durante a modelagem; (c) a estratégia não apoia os estudantes durante a modelagem; e (d) na estratégia, há falta de moderador nos grupos para a tomada de decisão. Entre os fatores identificados, os fatores (a) e (b) são benéficos, enquanto (c) e (d) são dificuldades percebidas pelos estudantes.

Tabela 5.6. Terceiro fator: Apoio durante a modelagem.

<p>A estratégia guia os estudantes durante a modelagem (evidenciado por 18 estudantes)</p> <p><i>Positive Examples (15 de 18):</i> “quando não sabíamos o que fazer na discussão, analisamos a planilha de exemplo positivo (...) Contamos com isso para ver se estava certo.” – Estudante P17B</p> <p><i>Negative Examples (3 de 18):</i> “ajudou a entender melhor como a especificação de casos de uso funciona.” – Estudante P09C</p>
<p>A estratégia ajuda os estudantes a identificar defeitos durante a modelagem (evidenciado por 8 estudantes)</p> <p><i>Inspection-Based strategy (3 de 8):</i> “[a estratégia ajudou] a resolver o problema a tempo, e evitar erros tolos e ter que refazer o diagrama.” – Estudante P10D</p> <p><i>Positive Examples (1 de 8):</i> “Consegui identificar os erros que cometi, para corrigi-los.” – Estudante P21B</p> <p><i>Negative Examples (6 de 8):</i> “[Foi] muito positivo porque me ajudou a encontrar meus próprios erros que eu estava escrevendo.” – Estudante P35B</p>
<p>A estratégia não apoia os estudantes durante a modelagem (evidenciado por 15 estudantes)</p> <p><i>Positive Examples (6 de 15):</i> “[essa estratégia] contribuiu pouco, pois a maioria da equipe nem olhou (os exemplos positivos) e se concentrou mais na própria atividade.” – Estudante P10B</p> <p><i>Negative Examples (9 de 15):</i> “Eu já tinha dúvidas e os exemplos negativos não me ajudaram, porque não sabia distinguir o que estava certo do que estava errado (durante a modelagem).” – Estudante P22B</p>
<p>Na estratégia, há a falta de moderador nos grupos para a tomada de decisão (evidenciado por 6 estudantes)</p> <p><i>Think-Pair Square (3 de 6) e Inspection-Based strategy (4 de 6):</i> “se eu perguntar a uma pessoa que tem o mesmo nível de conhecimento que eu, estaremos em dúvida; mas se eu perguntar a alguém que realmente sabe modelar, ele me ajudará.” - Estudante P03A</p>

A estratégia permite que o estudante receba feedback (quarto fator): a partir dos resultados, percebeu-se que o *feedback* fornecido foi considerado essencial pelos estudantes para melhorar a modelagem e pode ser um fator significativo na motivação dos estudantes. Foram identificados aspectos positivos e negativos sobre o *feedback* (Tabela 5.7), que se relacionam à estratégia baseada em inspeção, uma vez que, dentre as estratégias empregadas pelos professores, apenas esta possui uma etapa explícita de *feedback*. Constatou-se que (a) a

estratégia possibilita receber *feedback* dos problemas identificados pelas outras equipes; e (b) houve falta de comunicação/interação entre as equipes (modeladores e inspetores).

Tabela 5.7. Quarto fator: A estratégia permite que o estudante receba *feedback*.

<p>A estratégia possibilita receber feedback dos problemas identificados pelas outras equipes (evidenciado por 5 estudantes)</p> <p><i>Inspection-Based strategy:</i> “quando outras pessoas corrigem nossa modelagem, identificam erros que não conseguimos ver imediatamente [...] e provavelmente só veríamos mais de frente quando percebemos que algo está faltando em alguma parte do projeto.” – Estudante P10D</p>
<p>Falta de comunicação/interação entre as equipes (modeladores e inspetores) (evidenciado por 10 estudantes)</p> <p><i>Inspection-Based strategy:</i> “tem gente que não entendeu o nosso diagrama, então a pessoa relatou o problema e a gente perdeu tempo tentando entender por que ela relatou aquele problema.” – Estudante P08A</p>

Abstração e resolução de problemas (quinto fator): Ao realizar a análise dos dados, percebeu-se que a estratégia de *Similar Systems* influenciou os estudantes no processo de abstração e na solução de problemas (ver Tabela 5.8). Esta categoria incluiu três fatores: (a) a estratégia permite abstrair melhor o problema; (b) a estratégia integra teoria com prática; e (c) a estratégia não permite abstrair novos cenários.

Tabela 5.8. Quinto fator: abstração e resolução de problemas.

<p>A estratégia permite abstrair melhor o problema (evidenciado por 11 estudantes)</p> <p><i>Similar Systems:</i> “essa estratégia facilitou a modelagem porque quando você olha para um problema que realmente existe, surge um pouco dessa abstração e simulação que temos na sala de aula.” – Estudante P09C</p>
<p>A estratégia integra teoria com prática (evidenciado por 7 estudantes)</p> <p><i>Similar Systems:</i> “essa estratégia foi uma das mais úteis, porque, na minha opinião, era a melhor maneira de aplicar o conhecimento que vimos na sala de aula de maneira prática e em um sistema real.” – Estudante P22C</p>
<p>A estratégia não permite abstrair novos cenários (evidenciado por 3 estudantes)</p> <p><i>Similar Systems:</i> “era muito difícil usar um sistema pronto, porque eu tinha que seguir o que havia no sistema e era muito difícil pensar em algo novo [...] teríamos que nos esforçar mais para obter esse tipo de abstração.” – Estudante P37C</p>

5.3.2 QP2: Quais desafios os professores percebem ao usar estratégias de aprendizagem ativas enquanto ensinam diagramas da UML?

Entrevistas de acompanhamento foram realizadas com os professores, a fim de trazer informações mais profundas e responder ao nossa QP2. Os pontos levantados durante as entrevistas ajudaram a entender alguns desafios e incentivaram sugestões sobre como lidar com eles. Durante a análise dos dados, descobriu-se que alguns dos desafios relatados pelos

professores também foram mencionados pelos estudantes na sessão do Grupo Focal. Os desafios identificados foram: (a) os estudantes precisam ter conhecimento prévio sobre o conteúdo a ser ensinado; (b) os professores precisam ter cuidado ao definir o nível de dificuldade dos cenários utilizados; (c) os professores precisam definir adequadamente o tempo para aplicar as estratégias; e (d) os professores precisam tomar cuidado no desenvolvimento dos materiais necessários para implementar as estratégias.

O primeiro desafio (**os estudantes precisam ter conhecimento prévio sobre o conteúdo a ser ensinado**) foi identificado durante a aplicação de três das cinco estratégias: *Inspection-Based strategy*, *Think–Pair–Square* e *Similar Systems*. Os comentários dos estudantes e professores sobre esse desafio podem ser vistos na Tabela 5.9.

Tabela 5.9. Primeiro Desafio: Os estudantes precisam ter conhecimento prévio sobre o conteúdo a ser ensinado.

Inspection-Based strategy: “tivemos pouco conhecimento da notação e fomos expostos a um problema que exige que você já tenha um conhecimento mais razoável da notação.” – Estudante P08A

Think-Pair-Square: “nossa falta de conhecimento na notação pode ter prejudicado completamente o uso da estratégia.” – Estudante P08A

Similar Systems: “Eu pensei que era fácil (usar a estratégia), porque já tínhamos feito um exercício sobre casos de uso e já tínhamos uma noção do que era fazer [...] mas se não tivéssemos feito nada e usássemos a estratégia para modelar, o resultado seria muito ruim.” – Estudante P08A

Think-Pair-Square e Inspection-Based strategies: “Eu acho que deveria ter feito um ou dois diagramas para eles verem como é antes de aplicar as estratégias (*Inspection-Based and Think-Pair-Square strategies*).” – Professor I01

Similar Systems: “os estudantes tiveram muitas perguntas. Eles me disseram que precisavam de um passo a passo mais detalhado.” – Professor I02

Quando questionado sobre esse primeiro desafio, o professor I01 comentou que, após aplicar a estratégia, percebeu que o momento escolhido para uso com os estudantes era inconveniente. O segundo professor (I02) disse que, antes de aplicar a *Similar Systems strategy*, os estudantes deram *feedback* de que tinham muitas dúvidas sobre como realizar a modelagem do caso de uso. Segundo o professor (I02), os estudantes conseguiram entender melhor depois de uma aula que esclareceu as questões sobre modelagem.

Também foi observado que **os professores precisam ter cuidado ao definir o nível de dificuldade dos cenários utilizados**. Esse problema foi relatado pelos estudantes durante o uso da estratégia *Inspection-Based*, e o professor observou esse problema ao empregar a estratégia *Think-Pair-Square* (consulte a Tabela 5.10).

Tabela 5.10. Segundo Desafio: Os professores precisam ter cuidado ao definir o nível de dificuldade dos cenários utilizados.

Inspection-Based strategy: “a descrição textual da *Inspection-Based strategy* era um pouco confusa, era difícil de entender e faltava as informações que queríamos.” – Estudante P11A

Inspection-Based strategy: “essa estratégia depende completamente do cenário em que as pessoas estão [...] usando o cenário que usamos foi um cenário que veio do setor, era um cenário real, por isso era um pouco complexo. O cenário tinha informações ausentes, não era tão completo como se fosse um exercício, um ‘toy problem’, que costumamos usar em aulas.” – Professor I01

Em relação ao comentário do P11A, isso ocorreu porque o professor selecionou um cenário de desenvolvimento da vida real como ponto de partida. Isso afetou os estudantes durante a modelagem. Quando questionado sobre os cenários utilizados, I01 comentou que, como o cenário é real, isso afetou diretamente o uso da estratégia na sala de aula.

Ainda com relação à *Inspection-Based strategy*, o professor I04 comentou que, como consequência do uso de um cenário real, teve que planejar outra aula para finalizar a aplicação da estratégia, demonstrando o terceiro desafio (**os professores precisam definir adequadamente o tempo para aplicar as estratégias** - Tabela 5.11). Esse desafio também foi percebido por I03 ao empregar *Negative Examples* e por estudantes usando *Positive Examples* e *Negative Examples*.

Tabela 5.11. Terceiro Desafio: Os professores precisam definir adequadamente o tempo para aplicar as estratégias.

Positive e Negative Examples: “Eu pensei que os exercícios eram muito longos e levei duas aulas para terminar. Talvez reduzir a atividade pudesse ter sido mais fácil.” – Estudante P05B

Inspection-Based strategy: “o cenário era muito grande; levei muito tempo para aplicar a estratégia. Então, acabei passando duas aulas quando estávamos esperando apenas uma aula de duas horas.” – Professor I04

Negative Examples: “os estudantes não puderam realizar a atividade dentro do prazo da aula. Eu tive que reservar mais uma aula para que os estudantes pudessem terminar a atividade” - Professor I03

Quando questionados sobre o uso futuro da *Inspection-Based strategy*, o professor I01 comentou que eles poderiam adaptar os cenários de acordo com as estratégias: “Eu poderia usar problemas simples, problemas menores, para que os estudantes os resolvessem mais rapidamente. Isso é para as duas estratégias, especialmente o *Think-Pair-Square*, porque requer um tempo de aplicação muito mais longo que o primeiro.”

Outro desafio que foi observado é **que os professores precisam tomar cuidado no desenvolvimento dos materiais necessários para implementar as estratégias** (consulte a Tabela 5.12).

Tabela 5.12. Quarto Desafio: Os professores precisam tomar cuidado no desenvolvimento dos materiais necessários para implementar as estratégias.

Positive e Negative Examples: “a tabela com os exemplos contribuiu pouco, já que a maioria dos membros da equipe nem sequer a olhou [...], em poucos casos os exemplos positivos foram consultados.” – Estudante P10B

Inspection-Based strategy: “Eu realmente não gostei de aplicar esta estratégia devido à sua preparação [...] Para usá-la na aula de modelagem, é necessário: (1) ter um cenário/descrição do sistema a ser modelado; (2) ter um checklist para ajudar os estudantes a inspecionar os diagramas; e (3) disponibilizar o material sobre os diagramas para os estudantes. Não é trivial criar tudo isso, especialmente o segundo ponto.” – Professor I01

Positive e Negative Examples: “minha percepção é que eles não a seguiram. Eu dei a tarefa, alguns nem sequer viram as tabelas com os exemplos [...] então, eu percebi que eles não ligavam muito e pararam de ler.” - Professor I02

Positive e Negative Examples: “Eu simplesmente não gostei de ter que preparar os diagramas para os problemas.” - Professor I03

O professor I02 desenvolveu exemplos em formato de tabela quando aplicou as estratégias de *Negative Examples* e *Positive Examples*. No entanto, os estudantes discordaram se essa era a maneira mais apropriada de apresentar essas estratégias. Assim, o P10B relatou que não considerou essas estratégias úteis e não utilizaram as tabelas desenvolvidas pelo professor. I03 empregou *Negative Examples* em sua aula e, em vez de usar uma tabela, eles mostraram exemplos negativos em diagramas já modelados. Embora o professor não tenha gostado de preparar os diagramas para os problemas, acrescentou, “*não é muito esforço se pensarmos que normalmente passamos quase um dia inteiro para preparar uma aula de duas horas, por isso não foi tanto esforço.*”

Além disso, os professores relataram alguns benefícios do uso de estratégias de aprendizagem ativas durante o ensino da modelagem de software. Os benefícios identificados foram: (a) os estudantes estavam se sentindo mais engajados/desafiados ao usar as estratégias; e (b) as estratégias permitem que os estudantes desenvolvam diagramas mais completos e corretos, conforme pode ser visto na Tabela 5.13.

Tabela 5.13. Benefícios das estratégias percebidos pelos professores ao empregar estratégias ativas de aprendizagem.

Os estudantes estavam se sentindo mais engajados/desafiados ao usar as estratégias

Inspection-Based strategy: “Acho que eles adoraram colocar a mão na massa com essa estratégia (...) tenho a sensação de que eles sentiram um prazer em realizar a modelagem usando essa estratégia.” – Professor I03

Inspection-Based strategy: “os estudantes se sentiram desafiados e estavam realmente motivados a fazer a modelagem.” - Professor I03

As estratégias permitem que os estudantes desenvolvam diagramas mais completos e corretos

Think-Pair-Square: “the students had difficulty in looking at the sequence diagram and finding out in which class the method would be placed. They got mixed up in this, and the strategy helped them understand very well how to identify this.” – Professor I01

Inspection-Based strategy: “the strategy helped students improve the diagram shortly after finding the mistakes in their colleagues' diagrams. So, the strategy makes the student reflect, the checklist used already makes the student reflect on the diagram” – Professor I01

5.4 DISCUSSÃO

Os resultados deste estudo incentivam discussões em diferentes direções e têm implicações práticas, principalmente no que diz respeito aos professores que buscam usar estratégias de ensino baseadas na aprendizagem ativa. A implicação mais importante pode ser enfatizar as inúmeras maneiras pelas quais os professores podem empregar estratégias no processo de ensino da modelagem de software.

Em relação aos resultados do estudo com os estudantes, notou-se resultados positivos em relação à “dimensão factual” da taxonomia de Bloom (Anderson *et al.*, 2001) e uma atitude positiva, em relação à aprendizagem, quando eles usavam essas estratégias. Também se percebeu que, dentre as estratégias utilizadas, a estratégia *Inspection-Based* foi a melhor na perspectiva dos estudantes, considerando os resultados da Figura 5.5 e Figura 5.6, além dos resultados qualitativos. Isso pode estar relacionado ao fato de que essa estratégia permite que os estudantes recebam *feedback* durante sua aplicação. De acordo com Cooper *et al.*, (2005), o *feedback* é essencial para que os estudantes entendam onde os diagramas precisam ser aprimorados e, o mais importante, porque os diagramas são considerados de alta qualidade ou não.

Essa estratégia (*Inspection-Based*) é baseada em dois princípios fundamentais de aprendizagem ativa: fazer (*doing*) e refletir (*reflection*). A importância desses elementos é enfatizada por John Dewey (1916), que afirma que: “*métodos que são permanentemente bem-sucedidos na educação formal ... dão aos estudantes algo para fazer, não para aprender; e o fazer é de natureza a exigir pensamento ou a anotação intencional de conexões.*” Esses elementos também estão presentes no ciclo de aprendizado de Kolb: ação (*action*) e reflexão (*reflection*). Segundo Schon (1987), a importância da ação e da reflexão é que eles “[*ajudam*] os estudantes a se tornarem proficientes em um tipo de reflexão em ação; e ..., envolve um diálogo de professor e estudante que assume a forma de reflexão recíproca em ação.” Schon

(1897) acrescenta que quando as estratégias usam esse tipo de abordagem, o professor não precisa ensinar o estudante a ganhar mais conhecimento; é suficiente usar essa estratégia como uma forma de treinamento para melhorar o aprendizado.

Os resultados encontrados nesse estudo estão em consonância com os resultados relatados por Liu *et al.* (2001), que mostraram melhor aprendizado dos estudantes ao usar estratégias de ensino com etapas de *feedback*. Além disso, percebeu-se que as atividades de inspeção ajudavam os estudantes a compreender melhor os conceitos ensinados na aula. Os estudantes relataram que o *feedback* fornecido pelo estágio de inspeção lhes permitiu obter informações sobre modelagem e encontrar erros que foram ignorados pelos membros da equipe. A atividade de inspeção melhorou a compreensão dos estudantes sobre os defeitos que devem ser evitados durante a construção dos diagramas da UML. Identificou-se que se pode empregar o uso de *checklists* com os estudantes para identificar sistematicamente defeitos nos diagramas.

Esses resultados são corroborados também com os trabalhos de Cooper (2005), Hilburn (2011) e Silva *et al.* (2017a). Em todos esses estudos, identificou-se vários resultados positivos do uso da inspeção no ensino de Engenharia de Software, por exemplo, a redução do número de defeitos nos diagramas elaborados pelos estudantes. Percebeu-se também que, após realizar a fase de inspeção, os estudantes puderam entender melhor os conceitos e que a identificação de erros impede que os repitam novamente em futuras modelagens. Quando se combina estratégias focadas nas estratégias de “fazer (*doing*)” e “reflexão (*reflection*)”, os estudantes tendem a alcançar objetivos de aprendizado com mais eficácia durante o ensino da modelagem de software. Shekhar *et al.* (2019) comentaram que, quando a estratégia permite que os estudantes interajam, níveis médios a altos de envolvimento dos estudantes foram observados. Por outro lado, estratégias com baixa participação de instrução receberam consistentemente baixo envolvimento dos estudantes.

No entanto, com relação aos Similar Systems, observou-se uma alta taxa de discordância (e *feedback* neutro). Isso pode estar associado ao momento em que a estratégia foi aplicada. No caso desta estratégia, os estudantes da Turma B já haviam sido expostos a duas estratégias de ensino (*Negative e Positive Examples*). Além disso, os estudantes comentaram que o uso dessa estratégia era muito difícil devido ao alto nível de abstração necessário - os estudantes precisam abstrair o cenário de um sistema já desenvolvido primeiro e depois criar o diagrama. Por isso, os estudantes o uso da estratégia para treinar o conhecimento sobre o conteúdo, não diretamente como uma forma de ensinar. Acredita-se que isso ocorre porque, quando os estudantes não

aprenderam completamente o conteúdo, eles possivelmente não executaram a modelagem correta no início.

Embora a literatura afirme que o uso de estratégias diferentes no processo de ensino melhora os níveis cognitivos dos estudantes, conforme relatado em Silva *et al.* (2017a) e Vujović *et al.* (2014) e nos resultados desse estudo, percebeu-se que esse pode não ser o caso de todas as estratégias e/ou conteúdo. Talvez isso tenha sido influenciado pelo contexto cultural dos estudantes e sua cultura educacional, uma vez que alguns estudantes fizeram alguns cursos que adotam estratégias ativas de aprendizagem e outros estão acostumados a aprender passivamente (Nilson, 2016). Isso pode ter influenciado as percepções dos estudantes no ambiente de aprendizagem. Também foi observado que os professores podem não estar prontos para usar novas estratégias de ensino. Foram identificados alguns desafios que eles tiveram ao introduzir essas estratégias em suas salas de aula. Esses desafios podem dificultar a aplicação das estratégias no ensino de modelagem de software. Portanto, é necessário treinar os professores antecipadamente, pois a falta de capacitação dos mesmos pode impactar negativamente a aplicação da estratégia.

O construtivismo implica que os estudantes devem construir novos conhecimentos sobre seus conhecimentos anteriores, o que, no contexto da modelagem de software, é importante. No entanto, o conhecimento prévio dos estudantes sobre o conteúdo, para o qual eles começaram a formular novo conhecimento, não foi totalmente consolidado (Neguyen, 2006). Portanto, o que se percebe no primeiro desafio (**os estudantes precisam ter conhecimento prévio sobre o conteúdo a ser ensinado**) é que os estudantes primeiro precisavam se envolver mais nos processos de aprendizagem. Foi percebido que não é apropriado usar alguns tipos de estratégias no início do aprendizado, pois isso pode confundir os estudantes. Portanto, recomenda-se que, como pré-requisito para o uso dessas três estratégias, o professor precise ensinar os conceitos e demonstrar exemplos práticos para que os estudantes entendam as notações do diagrama ensinado. Assim, estratégias de aprendizagem apoiariam melhor a construção de conhecimento e ajudariam os estudantes durante a modelagem (von Glasersfeld, 2013). As estratégias de aprendizagem ativa também abordam a questão dos estudantes com diferentes níveis de conhecimento, porque cada estudante encontra a melhor maneira de trabalhar e aprender, a fim de contribuir para o resultado final esperado (Freeman *et al.*, 2014).

Segundo Fioravanti *et al.* (2018), os estudantes geralmente não conseguem realizar as atividades devido à complexidade dos cenários utilizados pelos professores. Portanto, **os professores precisam ter cuidado ao definir o nível de dificuldade dos cenários**, pois os

estudantes podem se sentir confusos com o cenário de alta complexidade e não conseguir concluir a atividade de modelagem. Isso, por sua vez, sobrecarrega os professores com vários pedidos de ajuda e assistência. Isso pode ocorrer mesmo se os estudantes estiverem trabalhando em pequenos grupos, nos quais a colaboração e a interação entre os membros da equipe estão presentes, pois os estudantes não se sentem tão seguros quando o cenário é complexo ou baseado em um sistema real.

Altamente relacionado a isso, descobriu-se que **os professores precisam definir adequadamente o tempo para aplicar as estratégias** (desafio 03). De acordo com Port e Kazman (2007), é difícil prever a quantidade de tempo necessária para executar estratégias de aprendizagem ativa. Os autores comentaram que *“isso parece ser altamente variável e altamente idiossincrático (talvez incontrolável, porque depende dos níveis de habilidade dos indivíduos nas aulas e estes variam amplamente.”* Quase todos os estudantes estão preocupados com o estresse devido ao tempo e à entrega da modelagem (Baker e Hill, 2017) e, de acordo com Aksit *et al.* (2016), os estudantes podem reclamar que estavam trabalhando sob pressão de tempo, dificultando a modelagem final. Com isso, os estudantes tenderão a preferir aulas teóricas do que aulas que utilizam essas estratégias (Tharayil *et al.*, 2018). Como sugestão para esse desafio, se os professores desejarem aplicar uma das estratégias em apenas uma aula (com duração de 40 minutos), é melhor usar um *“toy problem”*, seja ele criado pelos professores ou retirado de um livro ou, por exemplo, para que os estudantes modelem apenas parte de um cenário específico. Além disso, em estratégias cujas etapas são bem definidas, como no *Think-Pair-Square*, os professores devem definir um limite de tempo para que os estudantes concluam uma primeira versão da modelagem.

Finalmente, ao empregar essas estratégias na sala de aula, os professores enfrentam a tarefa inevitável de repensar cada aula para torná-la mais interativa, prática, envolvente e centrada no estudante (Port e Kazman, 2007). Portanto, o professor precisa de mais esforço para implementar essas estratégias (o que leva a mais pressão). Como há mais trabalho, **os professores precisam ter cuidado ao desenvolver o material necessário para implementar as estratégias** (desafio 04). Essa etapa é fundamental, pois segundo Aksit *et al.* (2016), a maioria dos estudantes reclama da falta de materiais necessários para aplicar as estratégias. Isso pode ser uma barreira para a implementação de algumas atividades que envolvem aprendizagem ativa, mas certamente não para todos, o que resulta em uma carga de trabalho ainda maior para os professores. Dessa forma, percebeu-se que algumas estratégias precisam de alguma preparação para implementá-las. Isso pode desencorajar alguns professores de aplicarem essas

estratégias, porque exige mais tempo para preparar as aulas e nem sempre há um *feedback* positivo dos estudantes, causando frustração e impedimento temporário à adoção de novas estratégias de ensino pelo professor.

5.5 CONSIDERAÇÕES SOBRE O CAPÍTULO

Este capítulo apresentou os estudos experimentais conduzidos com o objetivo de entender a influência dessas estratégias no processo de ensino e aprendizagem da modelagem de diagramas da UML. Foram coletados dados qualitativos por meio de questionários e sessões de Grupo Focal realizadas com os estudantes, e entrevistas semiestruturadas com os professores. A partir da análise das percepções dos estudantes, pode-se observar que algumas estratégias foram mais úteis (*Inspection-Based strategy*, por exemplo), enquanto outras foram consideradas mais difíceis e pouco úteis (*Similar Systems*, por exemplo).

A análise qualitativa permitiu identificar quinze fatores (benefícios e dificuldades) que influenciam a aprendizagem ao utilizar estratégias ativas de aprendizagem, os quais foram organizados da seguinte forma: pontos de vista durante a modelagem (02 benefícios e 04 dificuldades); aprendizagem colaborativa (03 benefícios); apoio durante a modelagem (04 benefícios e 02 dificuldades); e *feedback* fornecido aos estudantes (01 benefício e 01 dificuldade). Os benefícios e as dificuldades que foram identificadas podem ajudar os professores a escolher as estratégias a serem aplicadas e os estudantes a alcançar um objetivo específico de aprendizado.

Também identificou-se alguns desafios relatados pelos professores ao usar essas estratégias durante o ensino de modelagem: (i) os estudantes precisam ter conhecimento prévio sobre o conteúdo; (ii) os professores precisam tomar cuidado ao definir o nível de dificuldade dos cenários utilizados; (iii) os professores precisam definir adequadamente o tempo para aplicar as estratégias; (iv) e os professores precisam desenvolver cuidadosamente os materiais necessários para implementar as estratégias. Para cada desafio, foram sugeridas maneiras de contorná-los, para quando o professor precisar aplicar algumas das estratégias no futuro.

5.6 SÍNTESE DO CAPÍTULO

Este capítulo apresentou os resultados dos estudos experimentais conduzidos com as estratégias de ensino coletadas através do MSL. Para avaliar a viabilidade de adoção e de uso das estratégias, foram analisadas duas perspectivas do processo: as perspectivas dos estudantes que utilizaram as estratégias para compreenderem melhor os diagramas UML; e as perspectivas

dos professores que empregaram as estratégias para melhorar o ensino dos diagramas UML. A partir dos resultados encontrados, os estudantes relataram suas percepções sobre o uso das estratégias e foram identificados quinze fatores que influenciam no uso de estratégias de aprendizagem ativa no ensino dos diagramas da UML. Enquanto a literatura mostra que os estudantes geralmente respondem positivamente a estratégias de aprendizado ativas, os resultados mostram que alguns estudantes tiveram respostas negativas às adoções de novas estratégias de ensino. Além disso, foram identificados quatro desafios que os professores podem enfrentar ao adotar as estratégias de aprendizagem ativa em sala de aulas. Contudo, para cada desafio foi relatada sugestões de como contorná-los. Portanto, os resultados do estudo demonstram a relevância desse ponto de pesquisa, que ainda continua sendo investigado na comunidade de Educação em Engenharia de Software.

Com base nos resultados deste estudo, percebeu-se que professores e a maioria dos estudantes gostaram de utilizar essas estratégias. As estratégias são viáveis de serem utilizadas por professores de diferentes perfis, desde os novatos (com menos tempo em sala de aula) até os mais experientes. No entanto, notou-se que algumas estratégias precisam de alguma preparação antecipada para implementá-las. Isso pode desencorajar alguns professores de aplicarem essas estratégias, porque exige mais tempo para preparar as aulas e nem sempre há um *feedback* positivo dos estudantes, causando frustração e impedimento temporário à adoção de novas estratégias de ensino pelo professor. Assim, como sugestão para minimizar essas barreiras, seria interessante criar um repositório on-line para o professor consultar os materiais e as informações necessárias para aplicação das estratégias e também para disponibilizar os materiais já utilizados e melhorados, além de deixar a sua percepção sobre a estratégia como forma de *feedback* para outros professores.

CAPÍTULO 6 – CICLO DE *DESIGN* – *OPEN REPOSITORY FOR TEACHING SOFTWARE MODELING FROM ACTIVE LEARNING STRATEGIES (OPENSMALS)*

*Este capítulo apresenta o processo de criação, avaliação e evolução do repositório aberto que visa apoiar os professores na utilização de estratégias de aprendizagem ativa, denominado **Open Repository for Software Modeling Teaching from Active Learning Strategies (OpenSMALS)**.*

6.1 INTRODUÇÃO

Embora a literatura apresente a aplicação de estratégias ativas de aprendizagem em um contexto de modelagem de software, foi observado a partir do Mapeamento Sistemático (Capítulo 3), dos estudos exploratórios (Capítulo 4) e do Estudo de Viabilidade da Pesquisa (Capítulo 5) que faltam estudos relatando como os professores podem realizar o processo de seleção de estratégias de aprendizagem ativa de forma apropriada ao seu contexto de ensino. Além disso, não foi identificado através dos resultados do mapeamento sistemático da literatura um local ou repositório em que os professores possam consultar sobre as estratégias ou encontrar o material didático necessário para aplicá-las em sala de aula.

Com o objetivo de apoiar os professores a aplicar as estratégias ativas de aprendizagem, desenvolveu-se o OpenSMALS (*Open Repository for Teaching Software Modeling from Active Learning Strategies*, em português, Repositório Aberto para o Ensino de Modelagem de Software empregando Estratégias de Aprendizado Ativo). O OpenSMALS é um repositório aberto que apoia os professores a identificar estratégias de aprendizagem ativa mais adequadas ao seu contexto de ensino. O OpenSMALS visa minimizar as dificuldades práticas enfrentadas pelos professores durante a aplicação dessas estratégias fornecendo diretrizes específicas sobre como podem aplicar essas estratégias, cenários de modelagem providos por outros professores, questionários para avaliação das estratégias, entre outros. O OpenSMALS foi desenvolvido seguindo o método de *Design Science Research (DSR)*. O DSR tem sido amplamente aplicado em Sistemas de Informação e em pesquisas educacionais (Hevner e Chatterjee, 2010).

Nesse capítulo, será relatado o estudo piloto e dois ciclos de *Design* realizados para avaliar o OpenSMALS em quatro universidades de diferentes regiões do Brasil, Universidade Federal do Amazonas (UFAM), Universidade Estadual de Maringá (UEM), Universidade Tecnológica Federal do Paraná (UTFPR) e Centro Universitário do Norte (UNINORTE). Os

resultados dos estudos revelaram aos pesquisadores algumas limitações e oportunidades de melhoria no OpenSMALS.

6.2 PRIMEIRO CICLO DE *DESIGN*: DESENVOLVIMENTO DO OPENSMAALS (v1)

Seguindo a metodologia do DSR (Hevner, 2007), e também a partir das necessidades percebidas nos estudos exploratórios, foi proposto um repositório aberto para apoiar os professores na identificação de estratégias de aprendizado ativas mais adequadas ao seu contexto de ensino. O OpenSMALS reutiliza o conhecimento fornecido por outras propostas, baseando-se principalmente em estratégias de aprendizagem ativa usadas para o ensino de modelagem de software. Para apoiar os professores na escolha das estratégias de ensino apropriadas, considerou-se as estratégias identificadas e selecionadas a partir do Mapeamento Sistemático da Literatura explicado no Capítulo 3. No MSL, identificou-se os métodos que possuem qualquer um dos princípios de estratégia de aprendizagem ativa em seus referenciais teóricos. O processo de seleção das estratégias pode ser visto no Material de Apoio (Silva et al., 2020). Todas as estratégias disponibilizadas no OpenSMALS são apresentadas na Tabela 6.1.

Tabela 6.1. Estratégias selecionadas para compor o OpenSMALS.

Nome das estratégias	Descrição das estratégias	Autores
<i>Learning based on similar systems</i>	O método Aprendizagem Baseada em Sistemas Familiares permite que os estudantes investiguem um sistema semelhante ao que se pretende modelar e, com isso, realizar a modelagem do novo sistema utilizando como base as características identificadas no sistema escolhido.	Stoecklin <i>et al.</i> (1998)
<i>Jigsaw</i>	O <i>Jigsaw</i> é uma técnica que consiste em dividir o material de aprendizagem em tarefas parciais. Cada estudante da equipe é como uma peça de quebra-cabeças que executará uma tarefa parcial, que ao final do processo será compartilhada com todos os outros membros da equipe.	Portilho e Campos (2009)
<i>Learning from Erroneous Examples</i>	O <i>Learning from Erroneous Examples</i> (ErrEx) é um método baseado em Aprendizagem Ativa que engaja os estudantes em um exercício de aprendizagem ativa, oferecendo a oportunidade para uma <i>learning experience</i> construtiva e colaborativa através dos exemplos errados. Este método instrui os estudantes a encontrar, explicar e corrigir os erros.	Silva <i>et al.</i> (2017a)
<i>Inspection-Based strategy</i>	O <i>Inspection-Based strategy</i> tem por objetivo apoiar os estudantes que estão aprendendo como modelar os diagramas corretamente. Para isso, os estudantes utilizam comentários de inspeção de diagramas previamente inspecionados por outros estudantes.	Kinjo <i>et al.</i> (2006)
<i>Multimodal Approach</i>	Esta é uma abordagem multimodal que visa fomentar as habilidades de modelagem dos estudantes que são	Thevathayan e Hamilton (2017)

Nome das estratégias	Descrição das estratégias	Autores
	relevantes para a indústria, combinando uma abordagem orientada por projeto com várias atividades que focam em princípios de aprendizagem ativa.	
<i>Negative Examples</i>	O <i>Negative Examples</i> é um método que fornece diretrizes associadas a descrições e exemplos negativos de modelagem, para que estes sejam incorporados no ambiente de ensino de modelagem, fazendo com que os estudantes aprendam observando como se deve fazer uma modelagem de forma correta a partir de exemplos negativos trabalhados em sala de aula.	Bolloju <i>et al.</i> (2011)
<i>PBL + LBL Double Track Teaching Method</i>	Esse método de ensino baseia-se na <i>Lecture based learning</i> (LBL) e <i>Problem based learning</i> (PBL). Neste método de ensino o professor ensina a base teórica relacionada ao problema real e combina enquanto os estudantes estão trabalhando para a resolução de problemas real.	Wang <i>et al.</i> (2013)
<i>Positive Examples</i>	O <i>Positive Examples</i> é um método que fornece diretrizes associadas a descrições e exemplos positivos de modelagem, para que estes sejam incorporados no ambiente de ensino de modelagem, fazendo com que os estudantes aprendam observando como se deve fazer uma modelagem de forma correta.	Bolloju <i>et al.</i> (2011)
<i>Problem Based Learning Adapted</i>	O <i>PBL Adapted</i> é um método pedagógico que enfatiza o papel de um problema, no qual os estudantes são responsáveis pela sua aprendizagem. Este método ajuda os estudantes a desenvolver estratégias e construir conhecimento. Formação de equipe e distribuição de papéis são essenciais para o uso de PBL.	Silva <i>et al.</i> (2017a)
<i>Think-Pair-Square</i>	O Método <i>Think-Pair-Square</i> foi concebido para promover aprendizagens, discussões ativas e para auxiliar na resolução de problemas de forma cooperativa entre os estudantes. O método Think-Pair-Square é considerado idealmente quando se deseja que os estudantes obtenham experiências de trabalho colaborativo.	Scanniello e Erra (2014)

Após identificar as estratégias, organizou-se o conhecimento sobre cada estratégia, criando um modelo conceitual descrito como um diagrama de classes (veja a Figura 6.1). Para isso, definiu-se primeiramente o domínio e escopo que deveriam ser tratados pelo conhecimento levantado. O domínio é a representação e formalização semântica das estratégias de ensino baseadas em princípios de aprendizagem ativa. O escopo está relacionado ao suporte dado a estas estratégias, através do conhecimento organizado e representando semanticamente, facilitando a sua difusão e uso. Em seguida, realizou-se uma especificação em linguagem natural em que as informações sobre as estratégias foram organizadas, através dos artigos que relataram o uso das estratégias. Em seguida, estruturou-se as informações coletadas em um modelo conceitual descrito como um diagrama de classes (Figura 6.1).

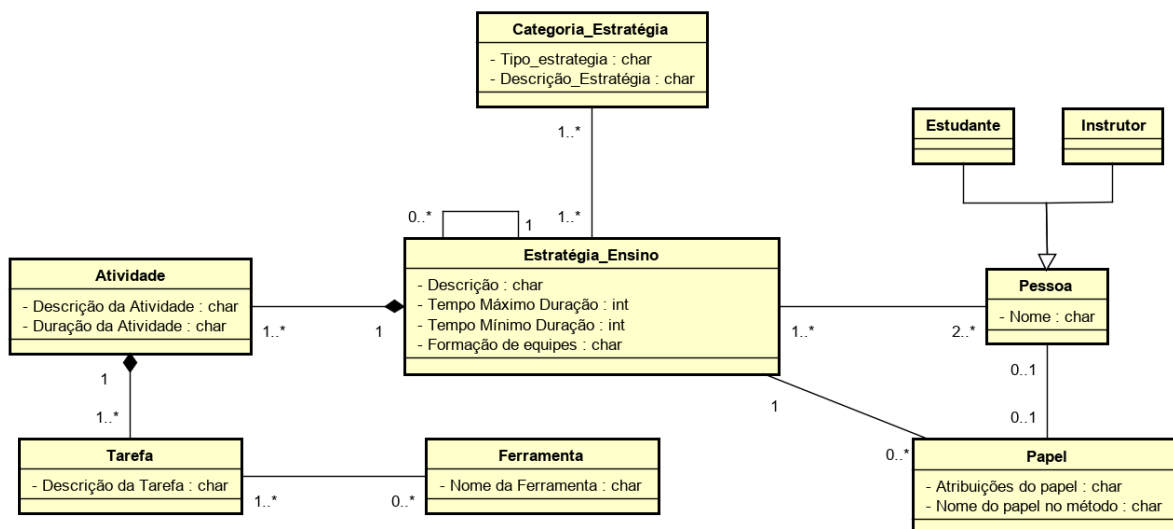


Figura 6.1. Modelo conceitual elaborado com as informações coletadas das estratégias.

No modelo, a classe `Estratégia_Ensino` representa o conjunto de estratégias de aprendizagem ativa identificados. Uma estratégia pode ser usada em conjunto com outra, complementando ou melhorando seus resultados. Isso é apresentado pelo auto relacionamento na classe `Estratégia_Ensino`. A classe `Estratégia_Ensino` está associada à classe `Categoria_Estratégia` que, conforme detalhado por Barkley *et al.* (2014), podem ser do tipo discussão, jogos, escrita, ensino recíproco, resolução de problemas e organizadores gráficos de informação, conforme pode ser visto na Tabela 6.2.

Tabela 6.2. Categorias das estratégias de aprendizagem ativa, baseado em Barkley *et al.* (2014).

Categoria	Descrição da categoria
Estratégias de discussão	Esse conjunto de estratégias tem como principais características o intercâmbio de informações, ideias e opiniões em discussões abertas e provocativas. O objetivo delas, em geral, é: ajudar os estudantes na formulação de ideias e como expressá-las devidamente; encoraja-los a pensar
Estratégias para o ensino recíproco	Nessas estratégias, os estudantes são ao mesmo tempo aprendiz e professor, nelas é dada ênfase à troca mútua entre esses dois papéis. Objetivam-se a incentivar à interdependência, assim como ajudar na retenção de informações pelos estudantes, que recebem reforço imediato de conceitos claros, sintetizando, esclarecendo e relatando ideias.
Estratégias para resolução de problemas	Este grupo de estratégias foi desenvolvido para ajudar estudantes no aprendizado e prática de estratégias para resolução de problemas. Têm como finalidade ensinar estudantes a identificar informações relevantes para a solução de problemas, analisando diferentes aspectos do processo
Estratégias com uso de organizadores gráficos de informação	Esse tipo de estratégia provê aos estudantes estruturas que os ajude a resumir textos, relembrar pontos chave e ideias, além de ajuda-los a avaliar e classificar eficientemente a ordem e integralidade dos processos de pensamentos de um grupo e apontar os pontos fortes e fracos de sua compreensão.
Estratégias com foco na escrita	De uma maneira genérica, a escrita pode treinar o pensamento crítico dos estudantes por ajuda na organização, resumo, integração e síntese de diversos elementos em uma ideia completamente coerente, e os torna cientes de seus próprios processos de aprendizagem

Categoria	Descrição da categoria
Estratégias usando jogos	Esse grupo de estratégias usa a participação em jogos como modo de envolver os estudantes com os conteúdos de uma forma divertida, o que motiva muitos estudantes e promove a aprendizagem ativa através de habilidades de prática em refino e resolução de problemas.

Além disso, uma estratégia consiste em um conjunto de uma ou mais atividades (classe Atividade). Cada atividade possui uma descrição do que deve ser realizado e seus objetivos, além da duração prevista para sua conclusão. Uma atividade é composta por uma ou mais tarefas (classe Tarefa). Uma tarefa é descrita por seus procedimentos. Algumas dessas tarefas podem requerer o uso de uma ferramenta de suporte para sua execução, como por exemplo: uma tarefa em uma estratégia pode ser discutir um determinado assunto, o professor pode definir que a tarefa deve ser executada por meio de uma ferramenta de comunicação síncrona (*chat*). As Estratégias de ensino têm como participantes pessoas, representadas na classe Pessoa, que podem ser de dois tipos: Professores e Estudantes. Essas pessoas podem exercer diversos papéis (classe Papel) na execução da atividade usada. Por exemplo, durante a execução da estratégia *Inspection-Based*, os participantes podem possuir o papel de modelador e de inspetor.

Como forma de explicitar o modelo conceitual elaborado, foi desenvolvido um questionário baseado no conhecimento das estratégias apresentadas no modelo conceitual. Com base nas respostas fornecidas pelos professores no questionário, o OpenSMALS recomenda um conjunto de estratégias de acordo com as necessidades do professor interessado em aplicá-las. Para isso, utilizando como base as informações coletadas no modelo conceitual e nos resultados identificados nos estudos exploratórios (Capítulo 4) e estudo de viabilidade (Capítulo 5), foi construída uma árvore de decisão inicial. Essa versão da árvore de decisão possui três níveis e, cada nível, possui perguntas que guiam o professor na escolha de estratégias mais adequada (Figura 6.2). A árvore de decisão foi criada por um pesquisador e validada por mais três pesquisadores com mais de 7 anos de experiência acadêmica e em ensino de modelagem. Todos os caminhos percorridos até a sugestão da estratégia foram revisados.

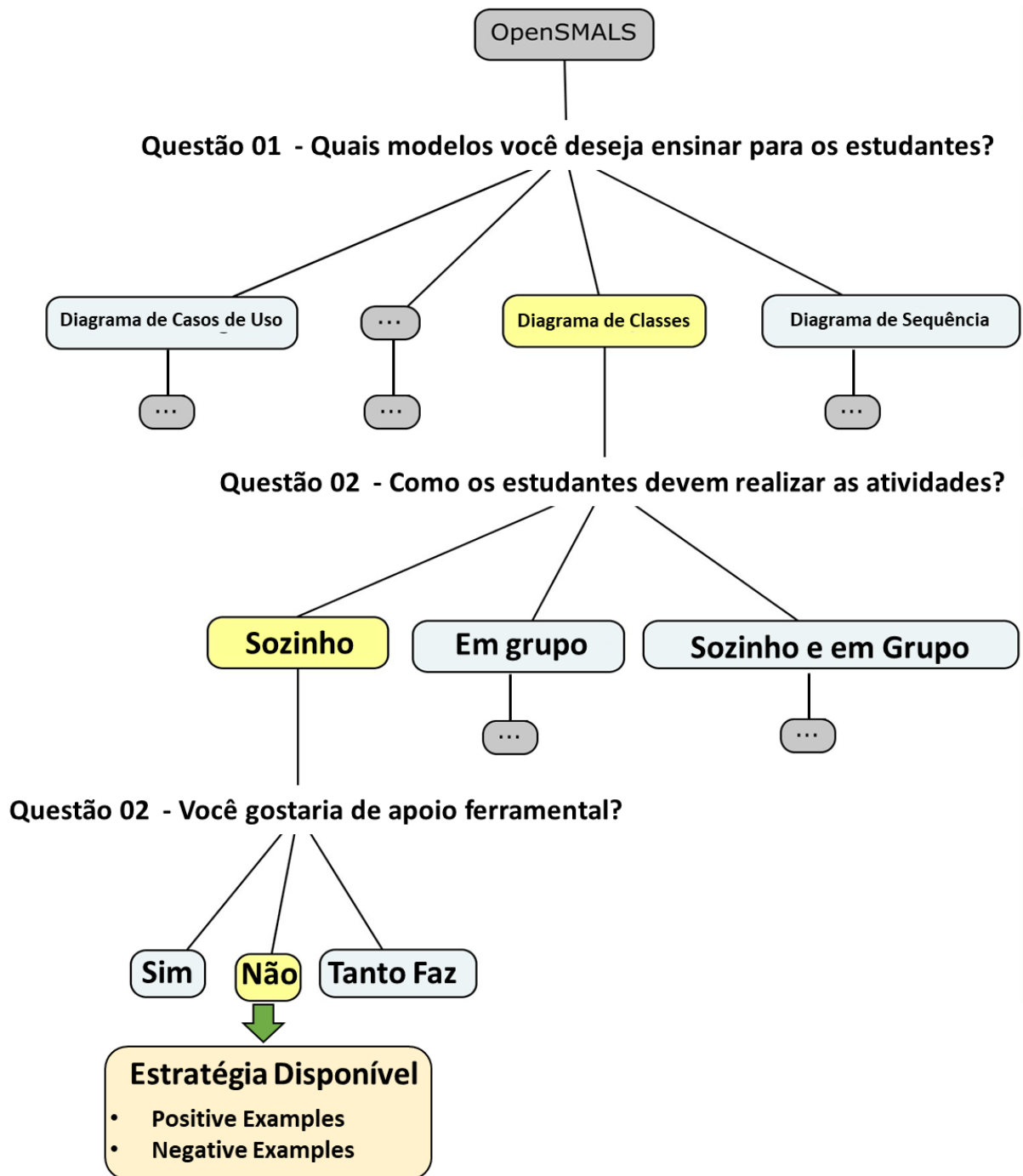


Figura 6.2. Exemplo de grupos de perguntas da primeira versão da árvore de decisão para o questionário.

Para aproximar o modelo de uma aplicação real, a seguir será apresentado um exemplo de seu uso. Um professor ministra uma disciplina de Engenharia de Software que possui 20 estudantes de graduação matriculados. O professor gostaria de realizar uma atividade para que esses estudantes possam melhorar os conhecimentos adquiridos em determinado diagrama e para isto busca uma estratégia e modos de melhor utilizá-la para obter êxito em seu objetivo. Com a ajuda do questionário elaborado, o professor poderá ter a resposta que deseja, bastando

para isso, responder às perguntas que perfilarão e indicarão estratégias que podem ser implementadas por esses professores em sala de aula. Para isso, ele selecionou o diagrama de classes, que é o diagrama que ele irá ensinar em sua turma; em seguida, ele deseja que os estudantes realizem as atividades individualmente, mas não quer que os estudantes dependam de ferramenta para realizar a atividade. A partir dessas características informadas pelo professor, o questionário sugeriu apenas uma estratégia: *Learning From Errouneous Examples*. A recomendação dada pelo questionário está representada na Figura 6.3.

O resultado do questionário mostra ao professor o nome da estratégia, uma breve descrição da estratégia e um link onde o professor pode clicar para obter mais informações. Mais informações sobre a estratégia selecionada podem ser encontradas no portal da web (veja a Figura 6.4 e Figura 6.5).

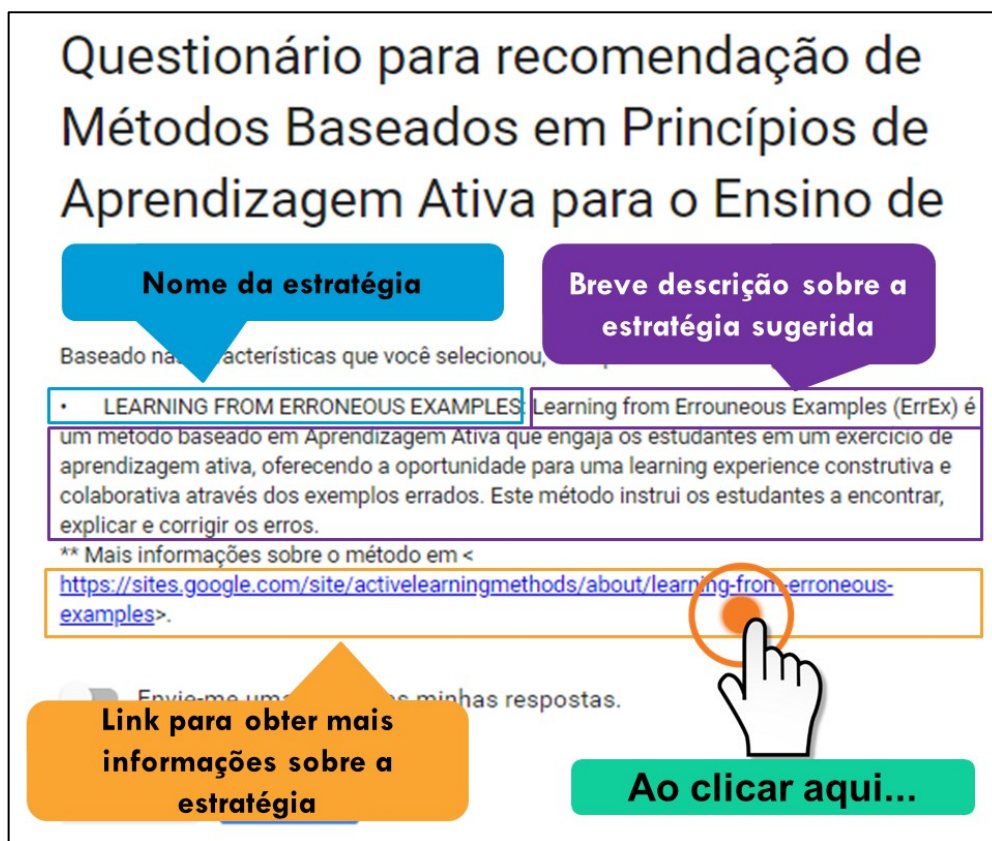


Figura 6.3. Estratégia sugerida pelo questionário.

No portal *web* apresentado nas Figura 6.4 e Figura 6.5, foram agrupadas algumas informações importantes para que o professor possa empregar a estratégia de aprendizado ativa escolhida: o nome da estratégia (Figura 6.4); uma breve descrição da estratégia (Figura 6.4); o diagrama no qual a estratégia pode ser empregada (Figura 6.4); as principais características das estratégias, bem como os principais benefícios que os estudantes obterão após o uso da

estratégia (Figura 6.4); o tipo de categoria à qual a estratégia pertence (Figura 6.4); o tempo mínimo necessário para a aplicação da estratégia (Figura 6.4); o número apropriado de estudantes necessários para a formação dos grupos, se necessário (Figura 6.4); as etapas de uso que podem ser seguidas pelo professor (Figura 6.5); o nome da ferramenta, se houver, e se o professor desejar usá-la com os estudantes (Figura 6.5); documentos como exemplos de uso anterior da estratégia (Figura 6.5); e, finalmente, as referências de onde as informações foram coletadas (Figura 6.5).

The image shows a screenshot of a web portal with a red header. The main content area is titled "Learning from Erroneous Examples". The page includes a navigation menu on the left, a search bar in the top right, and several sections of text. Callouts in various colors point to specific parts of the page:

- Nome do método sugerido:** Points to the title "Learning from Erroneous Examples".
- Breve descrição sobre o Método sugerido:** Points to the "Descrição do Método" section.
- Modelo que o método pode ser empregado para ensinar:** Points to the "Modelo UML que o método pode ser ensinado" section.
- Classificação do método:** Points to the "Características do Método" section.
- Tempo mínimo necessário para a aplicação do método:** Points to the "Tempo (Mínimo) Necessário para aplicação do método" section.
- Quantidade de estudantes por grupo:** Points to the "Quantidade de Estudantes" section.

Figura 6.4. Portal web com informações sobre as estratégias (1/2).

Para investigar o OpenSMALS com mais profundidade, realizou-se um estudo piloto e dois estudos empíricos. O planejamento e execução dos estudos são descritos nas próximas subseções.

<p>Teaching Method for Conceptual Modeling based on Error Analyses Contato dos Pesquisadores</p>	<p>Etapas de Uso do Método:</p> <ul style="list-style-type: none"> Etapa 01. O professor apresenta os conceitos sobre o modelo durante a aula e, em seguida, fornece um cenário para que os estudantes realizem a modelagem e o modelo com defeitos sintáticos e semânticos. Etapa 02. Os estudantes realizam a leitura dos cenários e, em seguida, iniciam o processo de identificação dos defeitos no modelo entregue. Etapa 03. Os estudantes realizam uma reunião de brainstorming para discutir os defeitos identificados. Etapa 04. Os alunos iniciam o processo de modelagem do novo modelo, seguindo o cenário fornecido e os defeitos identificados no diagrama anterior. A modelagem deste novo diagrama pode ser a partir do modelo com defeitos errados. Etapa 05. Os estudantes entregam o novo modelo corrigido.
<p>Site owners Williamson Silva</p>	<p>Apoio Ferramental: Não necessita de apoio ferramental.</p>
	<p>Artigos com Exemplos de Uso do Método:</p> <ul style="list-style-type: none"> Tsovaltzi D, McLaren BM, Melis E, Meyer AK, Dietrich M, Gogvadze G. Learning from erroneous examples. In International Conference on Intelligent Tutoring Systems 2010 Jun 14 (pp. 420-422). Springer, Berlin, Heidelberg. Tsovaltzi D, Melis E, McLaren BM, Meyer AK, Dietrich M, Gogvadze G. Learning from erroneous examples: when and how do students benefit from them?. In European Conference on Technology Enhanced Learning 2010 Sep 28 (pp. 357-373). Springer, Berlin, Heidelberg. Silva WA, Steinmacher IF, Conte TU. Is It Better to Learn from Problems or Erroneous Examples?. In Software Engineering Education and Training (CSEE&T), 2017 IEEE 30th Conference on 2017 Nov 7 (pp. 222-231). IEEE.
	<p>Referências: C. S. Grosse, and A. Renkl. "Finding and Fixing errors in worked examples: can this foster learning outcomes?". In Journal of Learning and Instruction, v.17, n. 6, pp. 612-634, 2007. B. M. McLaren, T. Van Gog, C. Ganoë, M. Karabinos, and D. Yaron. "The efficiency of worked examples compared to erroneous examples, tutored problem solving, and problem solving in computer-based learning environments". In Journal of Computers in Human Behavior, v. 55, pp. 87-99, 2016.</p>

The image shows a screenshot of the web portal content from the previous table, with several callout boxes highlighting specific sections:

- Top Right (Blue Callout):** "Etapas para aplicação do método" (Steps for application of the method).
- Left Side (Green Callout):** "Artigos com exemplos de uso do método" (Articles with examples of method use).
- Bottom Left (Pink Callout):** "Referências para o método" (References for the method).
- Center (Orange Callout):** "Necessidade de ferramenta" (Need for tool).

The highlighted content includes the "Etapas de Uso do Método", "Apoio Ferramental", "Artigos com Exemplos de Uso do Método", and "Referências" sections.

Figura 6.5. Portal web com informações sobre as estratégias (2/2).

6.3 PLANEJAMENTO E EXECUÇÃO DOS ESTUDOS EXPERIMENTAIS CONDUZIDOS PARA AVALIAR AS VERSÕES DO OPENSMALES

O artefato proposto (OpenSMALS) precisa ser avaliado aplicando-o ao problema e ao contexto, verificando se produz os efeitos desejados e se é necessária uma nova iteração sobre o Ciclo de *Design*, enquanto corrobora ou questiona a validade dos pressupostos teóricos (Hevner e Chatterjee; 2010). Assim, para avaliar o uso do OpenSMALS em um contexto prático, realizou-se primeiro um estudo piloto e dois estudos empíricos.

6.3.1 Planejamento

O estudo experimental foi realizado em quatro universidades diferentes. O estudo envolveu cinco professores (I01, I02, I03, I04 e I05) e 163 estudantes. A Tabela 6.3 mostra uma visão geral dos professores.

Tabela 6.3. Informações sobre os professores e a turma.

Avaliação	Universidade	Professor	Cursos	Disciplinas	Estratégias escolhidas	Diagramas UML	#estudantes
Primeiro Ciclo	Turma A – UFAM	I01	Sistema de Informação	Análise e Projeto de Sistemas	Inspection-based Strategy	Diagrama de Classes	14
					Think-Pair-Square	Diagrama de Sequência	
Segundo Ciclo	Turma B – UFAM	I02	Ciência da Computação	Introdução à Engenharia de Software	Positive examples	Diagrama de Casos de Uso	28
					Negative examples	Descrição de Casos de Uso	
	Learning based on similar systems	Diagrama e descrição de UC					
	Turma C - UEM	I03	Ciência da Computação	Engenharia de Software	Negative examples	Diagrama de Classes	35
Terceiro Ciclo	Turma D – UFAM	I04	Ciência da Computação	Análise e Projeto de Sistemas	Inspection-Based strategy	Diagrama de Sequência	21
	Turma E - UEM	I05	Ciência da Computação	Engenharia de Software	Erroneous Examples	Diagrama de Sequência	35
	Class F – UTFPR	I06	Sistema de Informação	Engenharia de Software	Positive examples	Diagrama de Classes	16
	Classe G – Uninorte	I07	Sistema de Informação	Engenharia de Software	Similar systems	Diagrama de Casos de Uso	14

Antes de começar com os ciclos reais de DSR, decidiu-se realizar um estudo piloto. Segundo Nunamaker e Briggs (2011), um pesquisador de *Design Science* pode realizar um estudo piloto primeiro para obter uma base útil para a pesquisa. Para o **estudo piloto**, selecionou-se um professor (I01) que ministrava a disciplina de Análise e Projeto de Sistemas. Esta disciplina é parte do 5º semestre do curso de Ciência da Computação e foi nessa disciplina que os estudantes tiveram o primeiro contato com modelagem de software. O professor tinha mais de 17 anos de experiência na indústria de software, com cinco anos relacionados à experiência prática e ao ensino de modelagem de software.

Depois disso, realizou-se dois estudos experimentais. Estudos empíricos podem ser realizados para melhorar a compreensão dos pesquisadores sobre a aplicação da tecnologia e permitir seu refinamento (Shull *et al.*, 2001). No **primeiro estudo experimental (primeiro ciclo de Design Science)**, selecionou-se outros dois professores (I02 e I03) que empregavam o OpenSMALS em seus cursos. I02 e I03 têm mais de 16 anos de experiência na indústria de software e no ensino de modelagem de software.

No segundo **estudo experimental (segundo ciclo de Design Science)**, I02 e I03 empregaram o OpenSMALS mais uma vez em seus cursos. Os professores usaram o

OpenSMALS para escolher outras estratégias de aprendizado ativas para o ensino de outros diagramas UML. Além disso, outros dois professores (I04 e I05) foram selecionados. O I04 tem mais de 17 anos de experiência no ensino de modelagem de software e o I05 tem três anos de experiência.

6.3.2 Execução

Para a execução do estudo, seguiu-se as etapas descritas na Figura 6.6, e cada etapa é explicada a seguir.

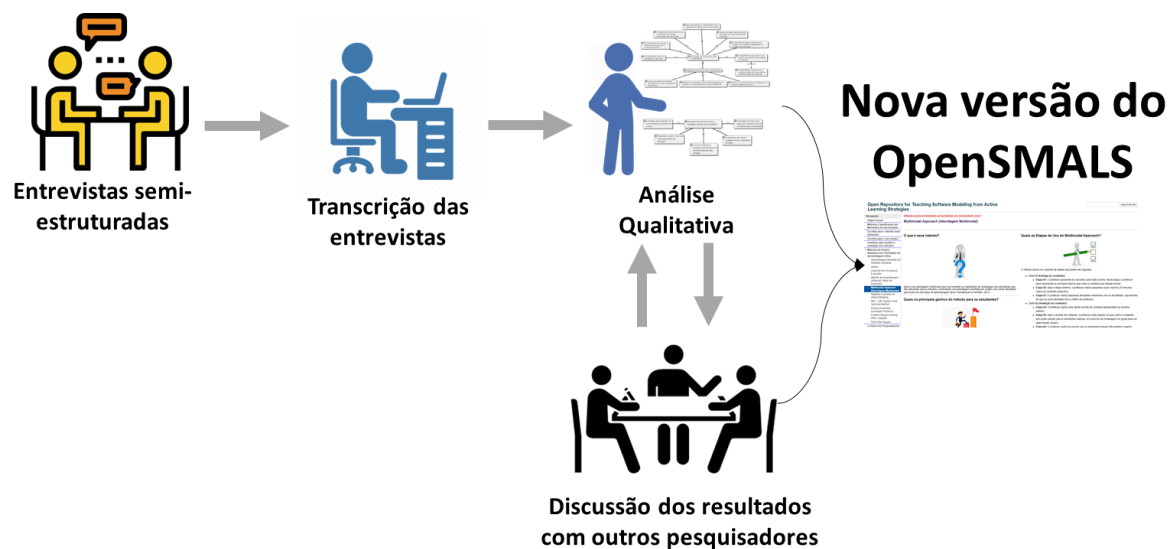


Figura 6.6. Visão geral do método de pesquisa.

Antes de iniciar o uso do OpenSMALS, o moderador entregou e assinou o termo de consentimento para todos os professores. Depois disso, os professores usaram o OpenSMALS para definir a estratégia de ensino que usariam na aula, a partir de uma lista disponível no portal da web. A Tabela 6.3 apresenta as estratégias selecionadas por cada professor.

Após escolher a estratégia, o professor reuniu mais informações sobre a estratégia escolhida para aplicá-la. Depois que os estudantes realizaram a atividade, o professor discutiu as soluções com os estudantes, fazendo-os refletir sobre o problema e a solução que eles modelaram. Dessa forma, os estudantes podem discutir e fazer perguntas sobre os diagramas. Depois que o professor utilizou o OpenSMALS para escolher a estratégia e aplicou a estratégia com a sua turma, foi realizada uma entrevista semiestruturada, que consistiu em uma mistura de perguntas abertas e específicas, projetadas para extrair tipos de informações previstas e inesperadas (Nunamaker e Briggs, 2011). Criou-se um roteiro de entrevista seguindo as recomendações da literatura (Runeson e Höst, 2009). As perguntas foram apresentadas em formato de funil, começando com questões gerais e avançando para questões mais específicas,

conforme sugerido por Runeson e Host (2008). A Parte I da entrevista visou obter o *background* do participante a fim de compreender qual a experiência do professor com estratégias ativas e como que o mesmo ensina modelagem para verificar a influência desses fatores ao utilizar o OpenSMALS. E a Parte II objetivou entender como o OpenSMALS apoiou o participante ao adotar as estratégias de ensino. As entrevistas são comumente usadas na Engenharia de Software Experimental e como parte da pesquisa em *Design Science* (Wohlin *et al.*, 2015). A Tabela 6.4 apresenta o roteiro com as principais perguntas utilizadas durante a entrevista.

Tabela 6.4. *Script* semiestruturado da entrevista.

Parte I – Background do Participante
Q1 – Há quanto tempo você leciona disciplinas na área de Engenharia de Software/ Modelagem de Software?
Q2 – Você possui experiência na indústria em modelagem de Software?
Q3 – Em suas aulas, como você ensina modelagem de software para os estudantes:
Parte II – Sobre o Uso do OpenSMALS
Q4 – Qual a sua percepção sobre o OpenSMALS?
Q5 – Qual a sua percepção sobre as estratégias agrupadas no OpenSMALS?
Q6 – Qual a sua percepção sobre as informações das estratégias que são fornecidas pelo OpenSMALS?
Q7 – Você tem alguma lição aprendida após o uso do OpenSMALS no ensino de diagramas UML?

As entrevistas foram realizadas individualmente e duraram entre 15 e 30 minutos. Durante as entrevistas, o professor pôde expressar suas percepções sobre o uso do OpenSMALS e falar sobre os desafios/dificuldades enfrentadas no uso dessa ferramenta. Esse processo foi realizado para o estudo piloto e estudos experimentais.

6.3.3 Análise dos dados

As análises realizadas com os resultados dos estudos tinham como objetivo de gerar novos *insights* e proporcionar uma melhor compreensão do problema e da solução proposta. Analisou-se qualitativamente a transcrição das respostas seguindo os procedimentos de codificação (Strauss e Corbin, 1998). O objetivo da análise qualitativa foi identificar as dificuldades e os benefícios percebidos pelos professores após o uso do OpenSMALS. Ao analisar os comentários adicionais dos professores, criou-se códigos (conceitos relevantes para a compreensão da percepção sobre as estratégias de aprendizado) relacionados às respostas dos professores (citações dos participantes). Em seguida, os códigos foram analisados e agrupados de acordo com suas propriedades, formando conceitos que representam categorias e subcategorias (codificação axial). Um pesquisador conduziu a análise e discutiu-a com os outros três pesquisadores em várias reuniões. Esta discussão foi realizada para mitigar qualquer

possível viés no processo de codificação. Os resultados da análise qualitativa e as discussões realizadas nos deram *insights* para melhorar o OpenSMALS.

6.3.4 Resultados do estudo piloto

Analisou-se a percepção do professor (I01) sobre o uso do OpenSMALS na escolha de uma estratégia de aprendizado ativa para apoiar o ensino de dois diagramas UML (diagrama de classe e diagrama de sequência). Os principais achados são apresentados na Tabela 6.5.

Tabela 6.5. Percepção do professor I01 sobre o OpenSMALS (v1).

<p>A sugestão do questionário OpenSMALS foi muito genérica</p> <p>“[...] eu achei que foi poucas perguntas e que estas foram gerais demais [...] as respostas que eu tive não foram de acordo com as minhas perguntas, não tive essa percepção. Se você tiver perguntas mais direcionadas para conhecer as características da turma, as características do tempo que o professor tem disponível para aplicar as estratégias. Eu o usaria novamente.” – I01</p>
<p>As informações disponíveis no OpenSMALS foram suficientes para entender como aplicar as estratégias</p> <p>“o ideal seria que o OpenSMALS fornecesse passos detalhados sobre como executar as estratégias com os estudantes. [...], por exemplo, durante a modelagem os estudantes poderiam consultar problemas relacionados ou consultar algum material bibliográfico? Eu acho que não, mas eu falei que eles poderiam consultar material para que eles pudessem obter um conhecimento mais aprofundamento sobre o conteúdo. Mas o OpenSMALS não me forneceu detalhes sobre isso.” – I01</p>
<p>O OpenSMALS pode sugerir estratégias de acordo com o momento em que o professor deseja aplicar a estratégia</p> <p>“O OpenSMALS poderia me perguntar algo como: você deseja usar a estratégia para ensinar os estudantes a modelar ou deseja usá-la para treinar mais ativamente o conteúdo da aula?” – I01</p>

O estudo piloto forneceu algumas causas dos resultados negativos obtidos com este professor. Considerando esses problemas, aprimou-se o OpenSMALS para acomodar as sugestões, resultando em uma segunda versão do portal OpenSMALS.

6.4 SEGUNDO CICLO DE *DESIGN*: DESENVOLVIMENTO DO OPENSMAALS (v2)

Avaliando os resultados encontrados no estudo piloto a partir da perspectiva do Req1 (o OpenSMALS deve permitir que os professores identifiquem as estratégias de aprendizado ativas mais apropriadas), não é possível dizer que foi alcançado no estudo piloto. Assim, revisitou-se o modelo conceitual e analisou-se novamente as informações coletadas para cada estratégia. Como resultado, uma nova versão da árvore de decisão (parte dela é apresentada na Figura 6.7) foi criada.

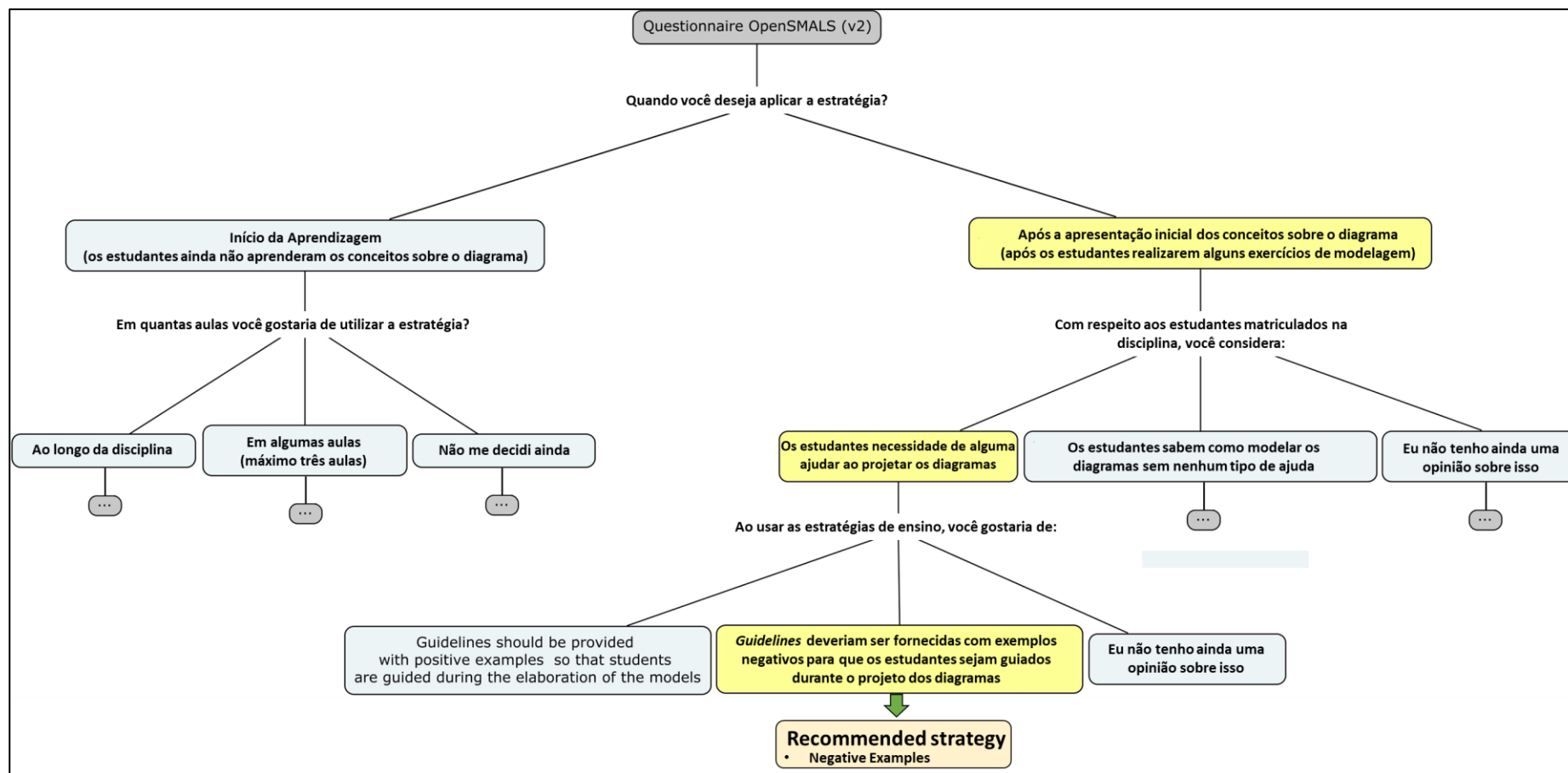


Figura 6.7. Extrato de perguntas da árvore de decisão do questionário - versão final.

As perguntas foram reagrupadas de acordo com o momento que os professores esperam usar a estratégia: no início do aprendizado e depois de ensinar os conceitos sobre os modelos. No primeiro momento, o portal direcionou os professores para estratégias que poderiam ser usadas desde o início da experiência de aprendizado do estudante, para que os mesmos sejam responsáveis por seu próprio aprendizado. Naquele momento, os professores devem atuar apenas como mediadores, ou seja, orientar os estudantes enquanto usam as estratégias. No segundo momento, os professores escolhem as estratégias que desejam aplicar após apresentar os conceitos básicos sobre o diagrama. Esses momentos foram definidos nesta segunda versão do questionário, porque, segundo Erickson (1982), esses momentos são os mais adequados para introduzir novas estratégias para melhorar a aquisição de novas habilidades e conhecimentos. Esses momentos de aprendizado revelam aos professores as novas possibilidades que os estudantes podem alcançar (Erickson, 1982; Weerdt *et al.*, 2002). Vale ressaltar que a nova árvore de decisão foi revisada por mais dois pesquisadores.

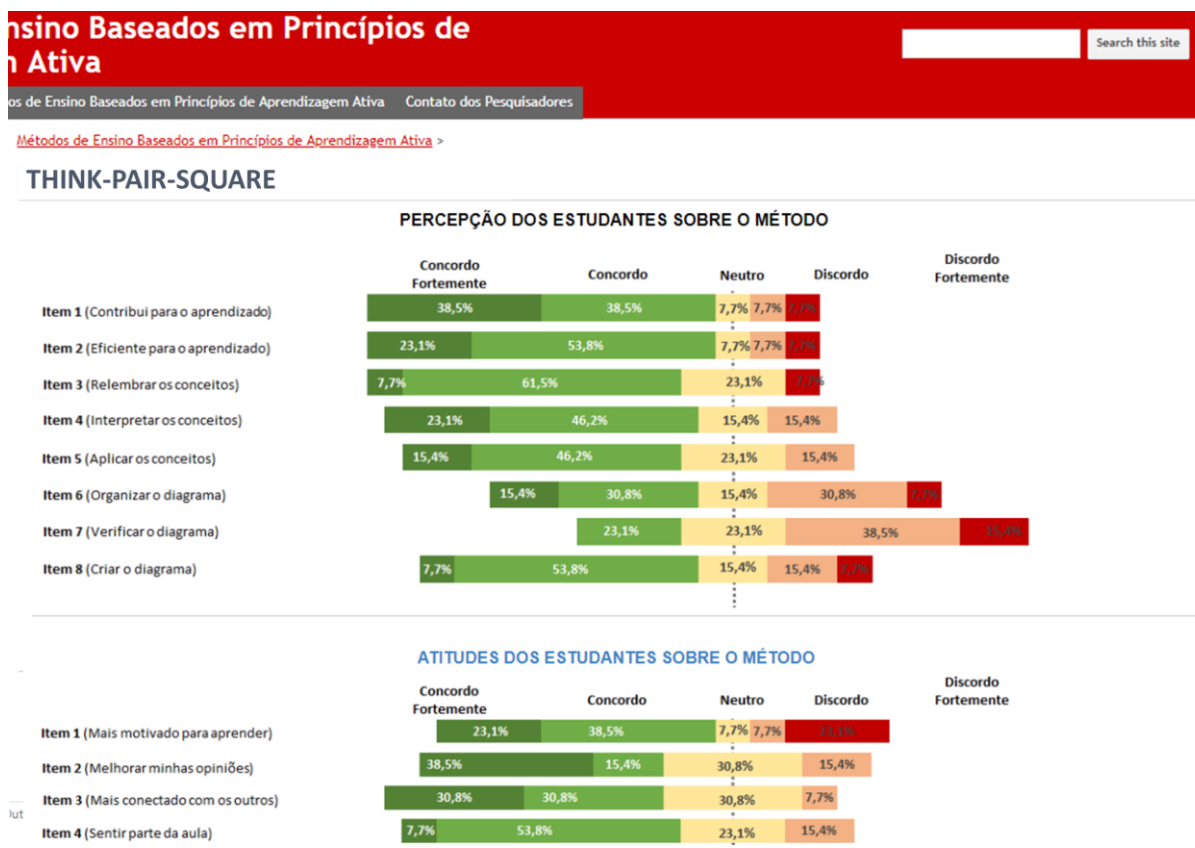


Figura 6.8. Exemplo de percepção dos estudantes sobre a estratégia (*Think-Pair-Square*) na segunda versão do OpenSMALS.

Em relação ao Req2 (o OpenSMALS deve fornecer procedimentos sobre como aplicar estratégias de ensino e prover ferramentas para apoiar o uso dessas estratégias), observou-se que as informações disponíveis no portal eram suficientes. Com base na sugestão do professor

I01, os estudantes responderam os questionários pós-modelagem para avaliar a experiência de aprendizado com as estratégias. Em seguida, as percepções dos estudantes sobre estratégias foram adicionadas no OpenSMALS (v2), conforme pode ser visto na Figura 6.8.

6.4.1 Resultados do primeiro estudo experimental

Realizou-se um estudo experimental para entender como os professores aplicaram o OpenSMALS. Analisando qualitativamente os dados coletados em entrevistas com professores (I02 e I03), identificou-se situações em que ocorreram as dificuldades e a facilidade de uso do OpenSMALS. Encontrou-se três benefícios percebidos pelos professores no uso do OpenSMALS (v2) (Tabela 6.6).

Tabela 6.6. Benefícios percebidos pelos professores usando o OpenSMALS (v2).

<p>OpenSMALS auxilia no planejamento de aulas “O OpenSMALS é muito objetivo, porque pergunta quantas aulas você deseja usar para empregar a estratégia. Isso é bom porque o ajudará no planejamento da aula [...], permitindo que você adapte as estratégias de acordo com o planejamento semestral.” – I02</p>
<p>OpenSMALS recomenda estratégia de acordo com o contexto de ensino “O OpenSMALS recomendou algo de acordo com as opções que eu escolhi no questionário [...] Eu também achei interessante porque não me recomendou apenas uma [estratégia].” – I03</p>
<p>O OpenSMALS sugere estratégias que podem ser usadas independentemente do diagrama que está sendo ensinado “você pode adaptar a estratégia recomendada pelo OpenSMALS a qualquer contexto de ensino, seja para o diagrama de casos de uso ou o diagrama de classes.” – I02</p>

Os códigos relacionados às categorias acima indicaram que os professores tinham uma boa percepção do OpenSMALS (v2). No entanto, os professores (I02 e I03) identificaram algumas dificuldades (Tabela 6.7).

Tabela 6.7. Dificuldades percebidas pelos professores usando o OpenSMALS (v2).

<p>O OpenSMALS não diz como o professor deve conduzir a organização dos estudantes “Fiquei me perguntando se os estudantes poderiam trabalhar em grupo ou não.” – I02</p>
<p>Não está explícito no OpenSMALS como apresentar a estratégia para os estudantes “as descrições dos exemplos são claras, em teoria. Mas, por exemplo, quando escolhi a estratégia e vi que ela tinha apenas texto, peguei as diretrizes e partes relacionadas de diagramas de classes para ilustrar cada um dos problemas negativos. Se eu tivesse uma ilustração para cada um desses problemas, para pessoas como eu que gostam de ilustrar, seria mais legal.” – I02</p>
<p>Houve um viés do facilitador devido à ajuda fornecida durante a implementação da estratégia “se não houvesse a ajuda do facilitador, eu não seria capaz de terminar a tempo e isso me daria um pouco mais de trabalho.” – I03</p>

Também identificados dois códigos que descrevem melhorias no OpenSMALS (v2) (consulte a Tabela 6.8). Os resultados de nossa análise mostraram questões sobre a (a) recomendação da estratégia, que deve levar em consideração o histórico do professor, (b) como

uma maneira de orientar o professor na escolha de uma estratégia, sugeriu-se a criação de um repositório com exemplos trabalhados que pode ser usado como base para outros professores.

Tabela 6.8. Sugestões de melhoria recomendadas pelos professores do OpenSMALS (v2).

<p>A recomendação da estratégia deve levar em consideração o <i>background</i> do professor “se eu fosse um professor novato, com apenas um ou dois anos lecionando, e tivesse três estratégias alternativas, acho que me perderia na escolha. Então, acho que se eu tivesse algo também relacionado à experiência dos professores, talvez isso me guiasse melhor.” – I03</p>
<p>Crie um repositório com exemplos trabalhados que possam ser usados como base para outros professores “uma sugestão seria ter um repositório com exemplos para usar o método a partir de materiais que podem ser personalizados, o que é algo que os professores estão sempre procurando.” – I02 “chega um momento em que você está sem tempo e não tem escolha a não ser dar a aula tradicional. Mas se você puder configurar um banco de cenários e oferecê-lo como um complemento para o OpenSMALS, acho que será muito legal.” – I03</p>

Os resultados desse ciclo de *design* indicaram que o OpenSMALS (v2) forneceu resultados úteis em relação ao seu uso para assistência do professor, quando comparado com os resultados da primeira versão do OpenSMALS. Os resultados mostraram que os professores consideraram essa nova versão do OpenSMALS (v2) útil e viável de usar. Observou-se neste estudo que os requisitos para avaliar o OpenSMALS foram atendidos. Percebeu-se ainda que, depois de melhorar o questionário, os professores relataram que foi sugerido estratégias de aprendizagem ativa de acordo com o contexto do ensino (então, alcançou-se o Req1). com base nas melhorias sugeridas pelos professores foi elaborada uma nova versão do OpenSMALS.

6.5 TERCEIRO CICLO DE *DESIGN*: DESENVOLVIMENTO DO OPENSMALES (v3)

Os resultados do primeiro estudo experimental (Seção 6.4.1) forneceram evidências para fazer melhorias que abordassem as dificuldades identificadas no uso do OpenSMALS. Na Tabela 6.9, apresenta-se as melhorias implementadas no OpenSMALS (v3), veja a Figura 6.9.

Tabela 6.9. Sugestões identificadas no OpenSMALS (v2) e que devem ser aprimoradas em uma nova versão.

Códigos	Professores	Soluções Implementadas
[Sugestão 1] Evitar o viés do facilitador durante a implementação da estratégia.	I02; I03	Criar um repositório onde todas as informações necessárias ao professor estejam disponíveis.
[Sugestão 2] Crie um repositório com exemplos trabalhados que possam ser usados como base para outros professores	I02; I03	
[Sugestão 3] O OpenSMALS pode mostrar como o professor pode conduzir a organização dos estudantes.	I03	Mostrar no OpenSMALS como o professor pode organizar os estudantes.

[Sugestão 4] O OpenSMALS poderia explicar explicitamente como apresentar a estratégia para os estudantes	I02; I03	Adicionar um apêndice ao OpenSMALS mostrando outras maneiras de apresentar estratégias para os estudantes.
[Sugestão 5] A recomendação da estratégia deve levar em consideração o <i>background</i> do professor	I03	Adicionar informações sobre o uso de outros professores no OpenSMALS.

Tinha-se em mente que o objetivo do OpenSMALS é ser usado por qualquer professor que deseje minimizar seus esforços para aplicar estratégias de aprendizagem ativas em seus cursos (Sugestões 1 e 2). Assim, levantou-se a hipótese de que adotar a sugestão de melhoria proposta pelos professores nos ajudaria a minimizar essa dificuldade.

Open Repository for Teaching Software Modeling from Active Learning Strategies


Navigation

- Home
- Strategies classified by moments of learning
- Choose the most appropriate strategy here
- Choose your scenario here
- Artifacts to help evaluate strategies
- Teaching Methods Based on Active Learning Strategies
 - Learning based on similar systems
 - Jigsaw
 - Learning from Erroneous Examples
 - Método de Aprendizagem utilizando Dados de Inspeções
 - Multimodal Approach
 - Negative Examples**
 - PBL + LBL Double Track Teaching Method
 - Positive Examples
 - Problem Based Learning (PBL) Adapted
 - Think-Pair-Square

Teaching Methods Based on Active Learning Strategies >


Negative Examples

What is this strategy?



Negative Examples is a teaching method based on an active learning strategy that provides problems that are associated with negative modeling descriptions and examples so that they are incorporated into the modeling teaching environment.

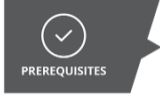
What are the main gains of the strategy for students?



This method allows students to:

- Understand the main difficulties other students had when modeling a particular model.
- Check from the examples worked in class, which are the primary defects that the other students committed to developing a model.
- It allows students to realize the defect and not realize it again.

What are the prerequisites for using this strategy?




- Students need to know the model notation and have previously solved modeling exercises.
- If the instructor wants to use real scenarios, the time for modeling can vary from one to two classes, each class with 1h40min.

Strategy Classification: Strategy for problem-solving.

References:
Bolloju N, Schneider C, Vogel D. Asymmetrical effects of using positive and negative examples on object modeling. In Information Systems Development 2011 (pp. 85-96). Springer, New York, NY.


What are the steps to applying the Negative Examples Strategy?



The strategy has a set of steps that can be followed:


- **Step 01:** The instructor introduces the concepts about the model during class and then provides a scenario for students to perform the modeling.
- **Step 02:** Students build a model based on the scenario delivered by the instructor.
- **Step 03:** The instructor conducts training on the difficulties that are most common during modeling of the requested artifact (below are some defects about some models to assist instructors).
- **Step 04:** The instructor gives the students a set of models with the primary defects marked.
- **Step 05:** The instructor asks students to model a new diagram based on training and a

What is the time for applying the strategy?



To apply the strategy, a class of 1h40min is required.

Come check out some assessments made with this strategy?



[Click here](#) to see the evaluations performed with this strategy.
Have you used this method? How about sharing with us what was the students' perception of the method? Just [Click Here](#)

Figura 6.9. Uma nova versão do portal OpenSMALS (v3).

Desta forma, decidiu-se transformar o OpenSMALS em um repositório aberto, no qual todos os artefatos necessários para implementar as estratégias são disponibilizados aos professores. Os artefatos fornecidos aos professores são:

- **Cenários de modelagem:** se os professores não tiverem cenários disponíveis para aplicar estratégias, eles podem usar alguns dos cenários fornecidos pelo portal (Figura 6.10.a);
- **Questionário de avaliação e uma planilha de coleta de dados:** se os professores desejarem avaliar a experiência de aprendizado e desejarem apresentar esses resultados aos estudantes para refletir sobre seu aprendizado (Figura 6.10.a).
- **Quadro do Grupo Focal:** se os professores desejarem coletar dados qualitativos, é fornecido um quadro com um modelo de grupo focal e instruções sobre como conduzi-lo com os estudantes (Figura 6.10.b).


(a) modeling scenarios	<p>Open Repository for Teaching Software Modeling from Active Learning Strategies</p> <p>Navigation Home Strategies classified by moments of learning Choose the most appropriate strategy here Choose your scenery here Artifacts to help evaluate strategies Teaching Methods Based on Active Learning Strategies Think-Pair Square Learning based on similar systems Jigsaw Learning from Erroneous Example Inspection-Based strategy Multimodal Approach Negative Examples PBL + LBL Double Track Teaching Positive Examples Problem-based Learning Adapted</p> <p>Choose your scenario here</p> <p>Scenarios you can use by employ</p> <p>Share your scenarios with us and help</p>  <p>Choose one of the scenarios below to apply to your class :-)</p> <ul style="list-style-type: none"> ▪ Scenario 01: Document Control System ▪ Scenario 02: Online Visa Request System ▪ Scenario 03: Online Student Accommodation Rental System ▪ Scenario 04: System for Coordination of the Program Committee of a Symposium ▪ Scenario 05: Online System to Support the Knowledge Management of a Research Group ▪ Scenario 06: Factory Restaurant System ▪ Scenario 07: System of a University Restaurant ▪ Scenario 07: Web Support System for Travelers ▪ Scenario 08: Online TCC System <p>W Scenario_for_modeling_01.docx (18k)</p>
(b) artifacts to help evaluate strategies	<p>Open Repository for Teaching Software Modeling from Active Learning Strategies</p> <p>Navigation Home Strategies classified by moments of learning Choose the most appropriate strategy here Choose your scenario here Artifacts to help evaluate strategies Teaching Methods Based on Active Learning Strategies Learning based on similar systems Jigsaw Learning from Erroneous Examples Método de Aprendizagem utilizando Dados de</p> <p>Artifacts to help evaluate strategies</p> <p>Questionnaires that can be used for methods based on active</p> <p>Dear Professor, Below we have added some questionnaires that you can use.</p> <ul style="list-style-type: none"> • Learning Assessment Questionnaire Perception about the strategy <ul style="list-style-type: none"> ▪ If you prefer, a version of the questionnaire is available here. • Frame and instructions to you perform the Focus Group with students. • Worksheet for analyzing the data about students' perceptions about strategies. <p>X Spreadsheet_data_datas_perceptions_students_strategies.xlsx (47k)</p> <p>W Frame_Instructions_Focus_Groups.docx (17k)</p> <p>W Questionnaire_Students_Perception.docx(20k)</p>

Figura 6.10. Páginas da Web com artefatos que os professores podem empregar em seus cursos.

Em relação à Sugestão 3, para cada estratégia, foi adicionado um campo que mostra que os estudantes podem ser agrupados de acordo com os objetivos de aprendizagem que o professor espera que os estudantes alcancem: em grupos (três ou mais estudantes), em pares ou individualmente. Em relação à Sugestão 4, adicionou-se um documento e uma apresentação sobre como apresentar e usar as estratégias em sala de aula. Este documento pode ser encontrado na parte inferior da página de cada estratégia. O professor pode escolher a melhor forma de apresentar esses problemas aos estudantes. Para a Sugestão 5, uma página sobre os professores foi adicionada. Nesse campo, cada professor poderá fornecer informações sobre a turma que implementou a estratégia.

6.5.1 Resultados do segundo estudo experimental

O segundo estudo experimental teve como objetivo entender se era possível reduzir os problemas identificados na v2, após implementar as soluções propostas. Após analisar as entrevistas com os professores que utilizaram a v3 (I02, I03, I04 e I05), identificou-se alguns dos pontos positivos e negativos percebidos ao usar esta versão do OpenSMALS (Tabela 6.10).

Tabela 6.10. Benefícios e dificuldades percebidos pelos professores que usaram (v3).

<p>[Benefício 1] Os professores escolheram as estratégias com base nas sugestões fornecidas pelo questionário “como gostei das recomendações do OpenSMALS da versão anterior, agora respondi ao questionário e ele me recomendou duas estratégias, exemplos negativos e <i>learning from erroneous examples</i>.” – I03 (que já teve experiência anterior com o OpenSMALS) “Eu acho que isso funciona muito bem quando você tem algo muito específico a aplicar; Eu não quero ver as outras estratégias; Quero uma recomendação de uma estratégia que faça o estudante aprender um determinado diagramas” – I05</p>
<p>[Benefício 2] O OpenSMALS recomenda a estratégia de acordo com o contexto de ensino “embora eu não tenha examinado todas as outras estratégias, acho que ele sugeriu uma boa estratégia; era exatamente o que eu queria usar com meus estudantes.” – I04</p>
<p>[Benefício 3] As etapas de uso apoiaram os professores na aplicação das estratégias “Li as descrições e as achei muito fáceis de entender (...) O OpenSMALS descreve as etapas de maneira clara (...) apliquei a estratégia com meus estudantes usando apenas as etapas de uso da estratégia, e achei muito fácil.” – I03</p>
<p>[Benefício 4] O material fornecido pelo OpenSMALS ajudou os professores durante a implementação das estratégias “O OpenSMALS me forneceu vários documentos de suporte para implementar a estratégia, (...) podemos usar os cenários e um formulário de avaliação da estratégia.” - I02</p>
<p>[Dificuldade 1] As percepções dos estudantes e professores não influenciaram a escolha das estratégias “Eu olhei para essas percepções, mas não prestei atenção a essas percepções e usei a estratégia porque quero ter essa experiência.” – I03</p>
<p>[Dificuldade 2] Não está explícito no OpenSMALS o esforço do professor para aplicar as estratégias “Antes de aplicar a estratégia, continuei pensando no esforço de preparar o material, e o OpenSMALS não fornece essas informações.” – I04</p>

A categoria Sugestões de melhoria para o OpenSMALS (v3) contém códigos que descrevem melhorias e novos recursos descritos pelos professores do OpenSMALS (v3), consulte a Tabela 6.11.

Tabela 6.11. Sugestões de melhoria recomendadas pelos professores do OpenSMALS (v3).

<p>O OpenSMALS precisa apresentar outras maneiras de como os professores podem usar as estratégias “Eu acho que valeria a pena também colocar algumas personalizações de como outros professores usaram as estratégias, por exemplo, sugerimos que ela seja aplicada assim, mas temos relatórios de outros professores que aplicaram de outra forma ... isso é muito legal para outros professores vejam que não há apenas uma maneira de aplicar o método.” – I05</p>
<p>O OpenSMALS poderia fornecer os oráculos dos cenários “Acredito que, em vez de deixar o cenário, também deixe uma das soluções possíveis para os professores usarem na sala de aula.” – I05</p>
<p>O OpenSMALS pode fornecer cenários para modelos específicos “Também poderia ter cenários específicos para diagramas específicos, por exemplo, cenários para diagrama de sequência, diagrama de classes, entre outros.” – I05</p>

Os resultados deste segundo ciclo de design mostraram que os professores aceitaram melhor o OpenSMALS (v3). Além disso, pode-se relacionar os motivos por trás dessa melhoria às soluções implementadas com base no *feedback* obtido sobre o uso do OpenSMALS (v2). Também se observou que a barreira relacionada ao apoio para aplicar as estratégias também foi resolvida nesta versão (v3) do OpenSMALS (Req 2). Com o OpenSMALS apoiando o professor no processo de como aplicar as estratégias, o professor poderia participar mais ativamente do processo de ensino com os estudantes. Conforme expresso pelas reflexões do professor, estabelecer uma rotina com estratégias ativas parecia definir o tom e a expectativa de como os estudantes deveriam se envolver durante a aula. Ao entender as causas das dificuldades percebidas pelos professores, conseguiu-se definir estratégias para tentar superar essas dificuldades. Observou-se ainda vários pontos positivos sobre o uso do OpenSMALS, relatados principalmente por professores com muitos anos de experiência na indústria e no ensino de modelagem de software. No entanto, alguns inconvenientes e sugestões de melhoria ainda foram relatados pelos professores. Esses pontos foram utilizados para melhorar o OpenSMALS, gerando sua quarta versão.

6.6 QUARTO CICLO DE *DESIGN*: DESENVOLVIMENTO DO OPENSMALES (v4)

No ciclo de *Design* anterior (usando a v3), os professores relataram que o OpenSMALS não é explícito na estimativa do esforço que o professor terá para aplicar essas estratégias. Nesse sentido, foi perguntado a cada um dos professores que aplicou as estratégias o quanto esforço - em horas - eles gastaram para preparar todo o material para a aula. Em seguida, para cada uma das estratégias utilizadas, foi adicionada uma seção chamada “Quanto tempo leva para implementar a estratégia”, com a média de tempo que os professores anteriores levaram para preparar os materiais (Figura 6.11). Quanto ao OpenSMALS apresentando outras maneiras de

usar as estratégias, não se conseguiu implementar esse ponto, pois todos os professores usaram as estratégias seguindo as etapas descritas no OpenSMALS. Essa melhoria pode ser realizada como um trabalho futuro nesta pesquisa.



Figura 6.11. Alterações feitas para adicionar tempo e esforço para preparar as estratégias.

Em relação a “fornecer cenários para diagramas específicos”, os cenários disponíveis no OpenSMALS. Para cada cenário, foram criadas perguntas relacionadas com propostas de exercícios para os diagramas de UML, como o diagrama de classes, o diagrama de casos de uso, a descrição de casos de uso, um diagrama de sequência e outros. Foram disponibilizadas possíveis soluções para cada um dos problemas disponíveis no OpenSMALS. Além disso, dada a natureza aberta do portal, e para aumentar o número de cenários disponíveis no OpenSMALS, os professores que usaram as estratégias antes forneceram gentilmente novos cenários, para que pudessem ser incorporados ao OpenSMALS e usados por outros professores. Devido à aceitação do OpenSMALS pelos professores e às melhorias feitas no portal, neste terceiro ciclo de *design*, não se realizou um novo estudo experimental. Melhorou-se o OpenSMALS apenas com base nas sugestões propostas.

6.7 CONSIDERAÇÕES SOBRE O ESTUDO

Este capítulo teve como objetivo apresentar o desenvolvimento, avaliação e evolução do o OpenSMALS. O OpenSMALS foi projetado para minimizar os desafios práticos enfrentados pelos professores durante a aplicação dessas estratégias. Para isso, fornece diretrizes específicas sobre como os professores podem aplicar cada estratégia, cenários de modelagem oferecidos por outros professores, questionários de avaliação, entre outros. Foi realizado um estudo piloto e dois estudos experimentais para entender como os professores aplicam o OpenSMALS em seu contexto de ensino. Finalmente, decidiu-se evoluir o OpenSMALS para um repositório aberto, em que os professores podem pesquisar e compartilhar suas experiências usando estratégias de aprendizado ativas. No portal foi

disponibilizado todos os artefatos necessários para aplicar cada estratégia. Assim, os professores podem aplicar a estratégia sem a necessidade dos pesquisadores.

Decidiu-se disponibilizar esses detalhes e explicações porque a literatura educacional nem sempre fornece detalhes sobre como implementar estratégias (Michael; 2007). Como resultado, os professores às vezes são incapazes de identificar os elementos críticos para implementar com sucesso uma estratégia em sala de aula (Borrego *et al.*, 2013). Devido a esse contexto e às barreiras existentes, a maioria dos professores decide seguir o ensino tradicional. O portal OpenSMALS é um esforço para criar uma ferramenta que orienta de forma explícita e acessível os professores sobre como cada uma das estratégias de aprendizado ativas deve ser implementada na sala de aula (com responsabilidade, desenvolvimento lógico e redução de apreensão) durante o ensino da modelagem de software. Destaca-se que a implementação adequada de estratégias ativas de aprendizagem não garante melhorias na aprendizagem dos estudantes. No entanto, aumenta as chances de sucesso em relação à abordagem tradicional (Freeman *et al.*, 2014). Atualmente, estão sendo planejados novos estudos com mais professores e em outros contextos de ensino para identificar novos fatores que podem influenciar o uso do OpenSMALS, bem como novas barreiras enfrentadas pelos professores ao empregar estratégias ativas de aprendizagem.

CAPÍTULO 7 – CICLO DE RIGOR DA PESQUISA E CONSIDERAÇÕES FINAIS

Este capítulo apresenta algumas discussões sobre o rigor da pesquisa, além das conclusões desta pesquisa e as suas principais contribuições. Além disso, são apresentadas as perspectivas futuras, fornecendo possíveis direções de continuidade desta pesquisa.

7.1 CONSIDERAÇÕES

O uso de estratégias de aprendizagem ativa vem ganhando destaque nos cursos de Computação. No entanto, os professores indicam várias barreiras que dificultam o uso dessas estratégias. Para investigar como apoiar os professores a implementar as estratégias de aprendizagem ativa no ensino de modelagem de software, esta tese foi orientada pelo método de pesquisa *Design Science Research*. O DSR permitiu a delimitação do problema de pesquisa, o desenvolvimento, avaliação e evolução de um artefato.

Inicialmente, com o objetivo de entender melhor o problema de pesquisa, foi realizado um mapeamento sistemático da literatura com um duplo objetivo: (a) identificar as principais dificuldades que os estudantes percebem ao realizar a modelagem de software, empregando os diagramas da UML; e (b) categorizar as tecnologias que estão sendo usadas para auxiliar os engenheiros no processo de modelagem. Os resultados do MSL permitem observar o estado da arte das tecnologias que visam melhorar o ensino e o treinamento dos futuros engenheiros de software, no que diz respeito à modelagem de software usando diagramas UML. Entretanto, a partir desses resultados, percebeu-se que existem poucos estudos que relatam/apresentam o uso de estratégias de aprendizagem que norteiam um processo de ensino mais eficaz dos diagramas UML.

Paralelo ao Mapeamento Sistemático, foi realizada uma investigação sobre as dificuldades em aprender a modelagem de Casos de Uso (diagrama e descrição textual). Além disso, foram conduzidos também estudos experimentais investigativos com o objetivo de avaliar os efeitos de uso de estratégias de ensino durante o processo de ensino de modelagem de software empregando os diagramas da UML. Embora a literatura afirme que o uso de estratégias diferentes no processo de ensino melhora os níveis cognitivos dos estudantes, percebeu-se que esse pode não ser o caso de todas as estratégias e/ou conteúdo, percebeu-se que partir desses resultados que os estudantes apresentam alguma resistência ao uso de

estratégias de ensino. Além disso, essas dificuldades também foram percebidas pelos pesquisadores ao empregar esse tipo de estratégia de ensino.

Baseado nos resultados do mapeamento sistemático e dos estudos exploratórios realizado no Ciclo de *Design* do DSR, o artefato que foi desenvolvido, avaliado e evoluído foi o OpenSMALS. O OpenSMALS é um repositório aberto que apoia os professores a identificar estratégias de aprendizagem ativa mais adequadas ao seu contexto de ensino. A avaliação do artefato proposto foi realizada por meio de estudos experimentais conduzidos. Os estudos foram conduzidos em quatro diferentes universidades do Brasil e com o objetivo de avaliar o OpenSMALS em seu real contexto de uso, ou seja, apoiando o professor ao aplicar as estratégias de ensino em sala de aula.

Então, respondendo à questão de pesquisa reportada no Capítulo 1 desta tese de doutorado (Como apoiar os professores na implementação de estratégias de aprendizagem ativa no ensino de modelagem de software?), indica-se como alternativa adotar o OpenSMALS como forma de auxílio quando se desejar adotar alguma estratégia de aprendizagem ativa em sala de aula durante o ensino de modelagem de software. Além disso, algumas das estratégias selecionadas para compor o OpenSMALS não tinham avaliação experimental, por exemplo, *Learning based on Similar Systems*, *Inspection-based Strategy*, *Negative Examples*. Com isso, a partir dos estudos realizados e dos resultados encontrados, foram encontradas evidências positivas sobre o uso destas estratégias no ensino de modelagem de software. Com isso, espera-se que as comunidades e pesquisadores da SEE aproveitem essa pesquisa como base para melhorar a educação em Engenharia de Software.

7.2 RIGOR DA PESQUISA

No que diz respeito ao rigor da pesquisa, este está relacionado à credibilidade, confiança, correção e honestidade da pesquisa, sendo exigido o rigor teórico e o rigor metodológico (Wieringa, 2009). O rigor é garantido quando o pesquisador segue um método de pesquisa já conhecido e validado, presencialmente um que seja amplamente aceito e reconhecido (Hevner e Chatterjee, 2010).

O rigor foi importante tanto no processo de criação do OpenSMALS quanto na sua avaliação e evolução. Esse rigor ajudou o pesquisador para não utilizar um formalismo excessivo prejudicando o processo de construção e avaliação do OpenSMALS, influenciando a relevância da pesquisa. Para isso, utilizou-se a fundamentação uma teórica e uma base de

conhecimento existente técnico existente. Além disso, o DSR não tem por objetivo apenas na aplicação do conhecimento para o desenvolvimento do artefato, é preciso também evoluir o corpo de conhecimento da área de pesquisa (Hevner e Chatterjee, 2010; Wieringa, 2009).

Portanto, visando alcançar o rigor esperado nesta pesquisa, foram utilizados vários métodos de pesquisa para demonstrar as etapas realizadas nesta pesquisa durante a execução dos ciclos de DSR. Entre os quais destacam-se o Mapeamento Sistemático da Literatura, os estudos experimentais que incluem a análise quantitativa da corretude e completude dos diagramas UML elaborados pelos estudantes e a análise qualitativa das sessões de entrevistas de Grupo Focal (*Focus Group*) realizadas com os estudantes, e a análise qualitativa de entrevistas realizadas com os estudantes e professores. Como um dos objetivos de aumentar a base de conhecimento na área de pesquisa, nas subseções a seguir os pesquisadores apresentam as diversas contribuições da pesquisa.

7.2.1 Contribuições da pesquisa

As principais contribuições desta pesquisa até o momento foram:

- Identificação, classificação e análise das evidências existentes sobre as dificuldades enfrentadas pelos engenheiros de software ao realizar modelagem de software empregando os diagramas da UML.
- Identificação, categorização e análise das estratégias de ensino identificadas na literatura científica que estão sendo propostas e que podem ser usadas pelos professores durante o processo de ensino e aprendizagem dos diagramas UML.
- Desenvolvimento de um modelo de dificuldades em modelar casos de uso (diagrama de casos de uso e descrição de casos de uso) que pode servir de base para professores no ensino/aprendizagem de modelagem de casos de casos de uso.
- Desenvolvimento e avaliação experimental do questionário MoDUC, baseado no modelo de dificuldades, para facilitar sua aplicação e entendimento dos professores ao investigar as dificuldades que os estudantes estão enfrentando enquanto aprendem a modelagem de UC.
- Integração das abordagens metodológicas *Design Science Research* e Engenharia de Software Experimental para o desenvolvimento, avaliação e evolução de repositório aberto (OpenSMALS) que orienta de forma explícita e

acessível os professores sobre como implementar cada uma das estratégias de aprendizado ativas na sala de aula.

- Evidências de que o OpenSMALS auxiliou professores de quatro universidades diferentes sobre como implementar estratégias de aprendizado ativo durante o ensino de modelagem de software.
- Evidências sobre as percepções dos estudantes e dos professores sobre o uso de estratégias de aprendizagem ativa durante o processo de aprendizagem de modelagem de software, ao empregar os diagramas UML.
- Evidências sobre a eficácia das estratégias de aprendizagem ativa identificadas na literatura e que professores podem implementá-las em sala de aula, melhorando assim a eficácia do processo de ensino.

7.2.2 Publicações

Como contribuições, também foram resultantes publicações em conferências e periódicos relacionadas aos resultados desta tese. As referências são listadas a seguir em ordem cronológica. Como forma de melhor demonstrar estas publicações, os capítulos da tese e quais publicações estão relacionadas a estas:

- **Artigo 1:** Silva, W., Steinmacher, I., e Conte, T. (2019). Students' and instructors' perceptions of five different active learning strategies used to teach software modeling. IEEE Access.
- **Artigo 2:** Silva, W., Gadelha, B., Steinmacher, I., e Conte, T. (2018). What are the differences between group and individual modeling when learning UML? *In Proceedings of the XXXII Brazilian Symposium on Software Engineering* (pp. 308-317).
- **Artigo 3:** Silva, W., Steinmacher, I., e Conte, T. (2017). Apoiando o Ensino de Diagrama de Atividades através de um jogo educacional. *In 25º Workshop sobre Educação em Computação (WEI 2017)* (Vol. 25, No. 1/2017). SBC.
- **Artigo 4:** Silva, W. A. F., Steinmacher, I. F., & Conte, T. U. (2017, November). Is it Better to Learn from Problems or Erroneous Examples?. *In 2017 IEEE 30th Conference on Software Engineering Education and Training (CSEE&T)* (pp. 222-231).

O artigo 1 apresenta os resultados do estudo experimental realizado com o objetivo de verificar a viabilidade de uso das estratégias de aprendizagem ativa através da percepção dos

estudantes e professores sobre as estratégias (Capítulo 5). Os artigos 2, 3 e 4 apresentam os resultados dos estudos exploratórios realizados com as estratégias de ensino (Capítulo 34).

Além disso, outros artigos publicados em colaboração com outros pesquisadores foram produzidos ao longo do doutorado e estão listados a seguir:

- **Artigo 5:** Damian, A. Marquez, A. B., Silva, W. Diniz, S., & Conte, T. (2020). Checklist-Based Techniques with Gamification and Traditional Approaches for Inspection of Interaction Models. *IET Software*.
- **Artigo 6:** Ferreira, B., Silva, W., Barbosa, S. D., & Conte, T. (2018). Technique for representing requirements using personas: a controlled experiment. *IET Software*, 12(3), 280-290.
- **Artigo 7:** Nascimento, E., Silva, W., de França, B. B. N., Gadelha, B., e Conte, T. (2017). Um Modelo sobre as Dificuldades para Especificar Casos de Uso. In *CibSE* (pp. 263-276).
- **Artigo 8:** Valentim, N. M. C., Silva, W., & Conte, T. (2017, May). The students' perspectives on applying design thinking for the design of mobile applications. In *Proceedings of the 39th International Conference on Software Engineering: Software Engineering and Education Track* (pp. 77-86). IEEE Press.
- **Artigo 9:** Nascimento, E., Silva, W., Conte, T., Steinmacher, I., Massollar, J., & Travassos, G. H. (2016). Is a Picture worth a Thousand Words?: A Comparative Analysis of Using Textual and Graphical Approaches to Specify Use Cases. In *Proceedings of the 30th Brazilian Symposium on Software Engineering* (pp. 93-102).
- **Artigo 10:** Silva, W., Valentim, N. M. C., & Conte, T. (2016, July). Designing Activity Diagrams Aiming at Achieving Usability in Interactive Applications: An Empirical Study. In *International Conference on Human-Computer Interaction* (pp. 208-219). Springer, Cham.
- **Artigo 11:** Nascimento, I., Silva, W., Gadelha, B., & Conte, T. (2016, July). Userbility: a technique for the evaluation of user experience and usability on mobile applications. In *International Conference on Human-Computer Interaction* (pp. 372-383). Springer, Cham.
- **Artigo 12:** Nascimento, I., Silva, W., Lopes, A., Rivero, L., Gadelha, B., Oliveira, E., & Conte, T. (2016). An empirical study to evaluate the feasibility

of a UX and usability inspection technique for mobile applications. In *28th International Conference on Software Engineering & Knowledge Engineering, California, USA*.

- **Artigo 13:** Ferreira, B., Silva, W., Oliveira Jr, E. A., & Conte, T. (2015, May). Designing Personas with Empathy Map. In *SEKE* (Vol. 152).
- **Artigo 14:** Silva, W., Valentim, N. M. C., & Conte, T. (2015, April). Integrating the Usability into the Software Development Process. In *Proceedings of the 17th International Conference on Enterprise Information Systems* (Vol. 3, pp. 105-113).

7.3 IMPLICAÇÕES DA PESQUISA

Estudantes de Engenharia de Software. De acordo com os relatos dos professores, após empregar as estratégias sugeridas pelo OpenSMALS, os estudantes ficaram mais engajados a participar das atividades. Além disso, alguns professores relataram que os estudantes tiveram um bom desempenho nas disciplinas, ou seja, projetando diagramas de forma mais corretos e completos. Em particular, I03 afirmou que - depois de corrigir os diagramas elaborados e aplicar a prova - *“olhando proporcionalmente o número de erros dos exercícios e na prova comparado às aulas anteriores, esses estudantes cometeram menos erros, considerando esses erros clássicos.”* Estudos relatam que os estudantes que aprendem por meio de estratégias de aprendizagem ativa têm habilidades mais variadas e aprofundadas do que os estudantes de outras universidades onde estratégias de aprendizado ativo não são adotadas/usadas (Shaw e Tomayko; 1991). De acordo com Kim (2017), essas estratégias também ajudam os estudantes a se auto avaliarem melhor e a se conscientizarem de que são mais uma personalidade focada no pensamento, na ação ou, melhor, nas pessoas. O *feedback* regular os ajuda a conhecer seus próprios pontos fortes e a trabalhar com seus pontos fracos. Portanto, esses pontos implicam que estratégias de aprendizagem ativas permitem que os estudantes expandam seu aprendizado e vão além de um curso convencional de Engenharia de Software.

Professores de Engenharia de Software: Usando as estratégias de aprendizagem ativas sugeridas pelo OpenSMALS, os professores podem atingir seus objetivos estratégicos de ter uma abordagem prática durante o ensino. A cada semestre, os professores enfrentam novos desafios e problemas. No entanto, ao empregar essas estratégias, o conhecimento transmitido aos estudantes enriquece muito o ensino diário. Em particular, I03 indicou que, ao usar essas

estratégias em sala de aula, “*sentiu-se atualizado, porque uma das coisas que você nota é que, depois de um tempo trabalhando na mesma coisa, todo ano a mesma coisa, da mesma maneira, dá uma sensação de estagnação. E esse sentimento de procurar uma maneira diferente de ensinar também me dá maior motivação. Além da motivação profissional, para melhorar o ensino, também sinto motivação pessoal.*” Além disso, o I02 afirmou que “*as estratégias de aprendizado ativo já trazem um enorme ganho simplesmente porque mudam a maneira como você ensina modelagem (...) Somente você apresenta de diferentes maneiras, já ajuda muito os estudantes com isso.*” Isso significa que mais professores podem ser motivados e incentivar o uso de estratégias ativas em suas aulas, aprendendo mais sobre os benefícios do uso dessas estratégias com os estudantes. Quando os professores introduzem esse tipo de estratégia na sala de aula, espera-se que os estudantes participem e se envolvam mais na aula. Essas estratégias podem facilitar a comunicação entre professores e estudantes por meio de *feedback* rápido e constante, que é uma característica deste tipo de estratégia de ensino. Dessa forma, o OpenSMALS pode auxiliar nesse processo, através do uso de estratégias ou questionários de *feedback* respondidos pelos estudantes.

Pesquisadores em Engenharia de Software: Existem muitas lacunas no mapeamento das estratégias e barreiras enfrentadas pelos professores, que podem ser exploradas em pesquisas futuras. Mais pesquisas são necessárias para investigar o momento de aplicação das estratégias e quais são as estratégias mais apropriadas para ensinar um diagrama específico. Conforme citado por I03, “*Existe uma maneira mais apropriada de usar exemplos negativos primeiro e depois exemplos errados para ensinar modelagem para estudantes?*” Seria de grande interesse analisar como essas estratégias influenciam positivamente a aprendizagem dos estudantes. Em particular, seria interessante entender os fatores de motivação e desmotivação que influenciam os professores no uso do OpenSMALS.

7.4 LIMITAÇÕES DA PESQUISA

Embora tenha-se analisado dados de várias fontes e diferentes salas de aula, provavelmente não se descobriu as percepções de todos os estudantes sobre estratégias de ensino e provavelmente não se evidenciou todos os desafios dos professores ou foram fornecidas explicações completas sobre esses desafios. Estamos cientes de que cada sala de aula que aplica a estratégia tem suas singularidades e que o universo de ensino de modelagem de software usando estratégias de aprendizado ativas é enorme. Isso significa que as percepções relatadas pelos estudantes e os desafios identificados podem diferir de acordo com a estratégia

ou a turma. Essa variedade nas respostas dos estudantes pode ter sido influenciada por fatores que influenciam diretamente a experiência real de aprendizagem, por exemplo, nível de experiência do professor (iniciante, iniciante, experiente ou especialista) e quantidade de orientação do estudante e do professor, treinamento e instrução com estratégias de aprendizado ativas. A estratégia de considerar diferentes métodos e diferentes turmas com diferentes perfis de estudantes teve como objetivo aliviar esse problema, identificando as percepções e os desafios que se repetem sob múltiplas perspectivas. Além disso, fornecer aos estudantes uma variedade de estratégias de aprendizagem ativa abordará suas muitas necessidades de aprendizado, com base em seus estilos de aprendizagem e posicionamento em seu processo de aprendizagem.

Outra ameaça à validade dos resultados é a subjetividade das classificações dos dados coletados. Para mitigar essa ameaça, foi empregada uma abordagem na qual todas as análises foram baseadas nos dados coletados. Além disso, foram conduzidas reuniões e foi-se discutida exaustivamente o processo de análise, juntamente com outros dois pesquisadores, para promover uma melhor validação das interpretações por meio de acordo mútuo.

Durante o Grupo Focal, ao pedir aos estudantes que anotem suas percepções em papéis coloridos e as leiam para outros estudantes, algumas dessas percepções podem ter sido influenciadas de alguma maneira. Para minimizar essa ameaça, os estudantes foram encorajados a fornecer suas opiniões e comentários o mais sincero possível. Além disso, a organização física da sala de aula utilizada e o grande número de discussões simultâneas entre equipes nos grupos focais provavelmente dificultaram a coleta de dados mais ricos. Nesse sentido, foi pedido aos estudantes que fizessem apenas um comentário de cada vez. Poder-se-ia dividir as discussões em outras sessões de Grupo Focal, mas não conseguimos garantir que os participantes estivessem disponíveis ou comprometidos em participar novamente.

Além disso, durante a entrevista, os professores foram solicitados a responder perguntas relacionadas à aplicação das estratégias escolhidas. O roteiro da entrevista foi semiestruturado e, portanto, continha questões abertas para captar as percepções dos professores sobre as estratégias. Para minimizar essa ameaça, o roteiro da entrevista foi avaliado por dois pesquisadores. Após a avaliação, foi realizada uma entrevista piloto com um professor com experiência anterior no ensino de modelagem, a fim de verificar se o script alcançaria seu objetivo. Os resultados da entrevista foram avaliados em conjunto com outro pesquisador e não houve necessidade de aprimorar o roteiro.

7.5 PERSPECTIVAS FUTURAS

A realização desta pesquisa possibilitou, além de outros resultados, na elaboração de um repositório aberto de ensino. Esse resultado abre novas perspectivas de pesquisa, que podem ser exploradas em trabalhos futuros. Alguns desses trabalhos futuros são relatados a seguir:

Inserção de novas estratégias/outros artefatos no repositório. Atualmente, o processo de inserção de novas estratégias ou outros artefatos (como os cenários para modelagem, ou os oráculos desses cenários) ainda é dependente da aprovação do pesquisador responsável pelo OpenSMALS. Com isso, espera-se elaborar um outro artefato (possivelmente um questionário) em que outros professores possam inserir informações sobre novas estratégias de forma colaborativa, e em seguida, estas serão adicionadas no OpenSMALS. Além disso, pretende-se elaborar um guia com informações referentes à inclusão de novas estratégias de aprendizagem ativa.

Melhoria na recomendação das estratégias. O processo de recomendação das estratégias ainda é restrito apenas aos pesquisadores dessa pesquisa. Isto significa que quando uma nova estratégia for inserida pelos professores, os pesquisadores precisaram verificar como que a mesma deve ser estruturada para poder ser recomendada da forma mais adequada possível. Com isso, espera-se que no futuro, esse processo de recomendação seja automatizado por meio de agente de inteligência artificial que utilize algumas regras de inclusão destas estratégias. Além disso, o processo de registro de informações e experiências de outros professores ainda é realizado de maneira manual. Pretende-se trabalhar em um algoritmo que agrupe as experiências de ensino relatadas pelos professores e as incorpore no repositório. Isso visa caracterizar melhor às discussões realizadas, a instância de uso das estratégias, a recomendação das mesmas para professores, experientes ou não, no ensino de modelagem e na aplicação de estratégias de aprendizagem ativa.

Melhoria na interação do professor com o questionário OpenSMALS. Atualmente, o OpenSMALS sugere uma estratégia baseado na resposta do professor no questionário. Uma sugestão para melhorar essa interação, seria transformar o questionário de recomendação em *chatbot*. Assim, *chatbot* do deveria sugerir a estratégia(s) necessária(s) de acordo com que foi perguntado pelo professor, além de direcionar o professor para o local em que se encontra informações detalhadas sobre a(s) estratégia(s) de ensino.

Melhoria da interface do OpenSMALS. Atualmente o OpenSMALS está em disponível na plataforma do Google Sites. Então, para melhorar a *teaching experience* durante o uso do repositório, pode-se utilizar técnicas de UX e Usabilidade para avaliar o portal, além

de utilizar técnicas de *Design Thinking* para auxiliar no reprojeto da interface do portal Web do OpenSMALS.

REFERÊNCIAS

- ACM, IEEE. Software Engineering 2004. Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering, 2014.
- Aglan, H.A., Ali, S.F. “Hands-on Experiences: An Integral Part of Engineering Curriculum Reform.” In *Journal of Engineering Education*, pp. 327-330, 1996.
- Agner, L. T., Lethbridge, T. C., Soares, I. W. “Student experience with software modeling tools.” In *Journal of Software & Systems Modeling*, pp. 1-23, 2019.
- Agner, T, Lethbridge, T. C. “A survey of tool use in modeling education”. In *20th Int. Conf. on Model Driven Engineering Languages and Systems (MODELS)*, pp. 303-311, 2017.
- Aksit, F., Niemi, H., e Nevgi, A. “Why is active learning so difficult to implement: The Turkish case”. In *Australian Journal of Teacher Education*, vol. 41, no. 4, pp. 94-109, 2016,
- Al-Busaidi, K., Al-Shihi, H. “Key factors to instructors’ satisfaction of learning management systems in blended learning”. In *Journal of Computing in Higher Education*, vol. 24, no. 1, pp. 18-39, 2012.
- Al-Tahat, K. “An innovative instructional method for teaching object-oriented modelling”. In *International Arab Journal Information Technology*, v.11, n.6, pp. 540-549, 2014.
- Anda, B., Hansen, K., Gullesen, I., Thorsen, H. K. “Experiences from introducing UML-based development in a large safety-critical project”. In *Empirical Software Engineering*, vol 11, no.4, pp. 555-581, 2006.
- Anderson, L. W., Krathwohl, D. R., Airasian, P. W., Cruikshank, K. A., Mayer, R. E., Pintrich, P. R., Raths, J., e Wittrock, M. C. “A taxonomy for learning, teaching, and assessing: A revision of Bloom’s taxonomy of educational”. Longman, 2009.
- Andrade, R. M. C., Santos, I. S., Araújo, I. L., Aragão, B. S., e Si-ewerdt, F. “Retrospective for the last 10 years of teaching software engineering in ufc’s computer department”. In *Proceedings of the 31st Brazilian Symposium on Software Engineering*, pages 358–367, 2017.
- Badreddin, O., Khandoker, R., Forward, A., Masmali, O., Lethbridge, T. C. “A Decade of Software Design and Modeling: A Survey to Uncover Trends of the Practice.” In *Proc. 21th MODELS*, Copenhagen, Denmark, 2018, pp. 245-255.

- Badreddin, O., Lethbridge, T. C., e Elassar, M. "Modeling practices in open source software". In *IFIP International Conference on Open Source Systems*, pp. 127-139, 2013.
- Bąk, K., Zayan, D., Czarnecki, K., Antkiewicz, M., Diskin, Z., Wąsowski, A., e Rayside, D. "Example-driven modeling: model= abstractions+ examples". In *International Conference on Software Engineering (ICSE'13)*, pp. 1273-1276, 2013.
- Baker, E., e Hill, S. "Investigating student resistance and student perceptions of course quality and instructor performance in a flipped information systems classroom". In *Information Systems Education Journal*, vol. 15, no.6, pp. 17-26, Nov. 2017.
- Barkley, E. F., Cross, K. P., e Major, C. H. "Collaborative learning techniques: A handbook for college faculty". John Wiley & Sons, 2014.
- Bartlett, M. S. "Tests of significance in factor analysis," *Brit. J. Stat. Psychol.*, vol. 3, no. 2, pp. 77-85, 1950.
- Baser, M. "Promoting conceptual change through active learning using open source software for physics simulations". In *Australasian Journal of Educational Technology*, vol. 22, no. 3, pp. 336-354, 2006.
- Bera, P. "Analyzing the cognitive difficulties for developing and using uml class diagrams for domain understanding". In *Journal of Database Management (JDM)*, vol. 23, no. 3, pp. 1-29, 2012.
- Bloom, B. S. "Taxonomy of educational objectives: The classification of educational goals: handbook I, cognitive domain". New York; Toronto: Longmans, Green, 1956.
- Bloom, B. S., *Taxonomy of educational objectives: The classification of educational goals, Handbook I, Cognitive Domain*, New York; Toronto: Longmans, Green. 1956.
- Boberic-Krstićev, D., e Tešendić, D. "Experience in teaching OOAD to various students". In *Informatics in Education*, vol. 12, no. 1, pp. 43-58, 2013.
- Boehm, B.W., Brown, J.R., e Lipow, M. "Quantitative evaluation of software quality". In *Proceedings of the 2nd international conference on Software engineering*, pp. 592-605, 1976.
- Bolchini, D., e Garzotto, F. "Quality of Web usability evaluation methods: an empirical study on MiLE+". In *International Workshop on Web Usability and Accessibility*, pp. 481 - 492, 2007.

- Bolloju, N. “Exploring quality dependencies among UML artifacts developed by novice systems analysts”. In *AMCIS 2006 Proceedings*, pp. 472, 2006.
- Bolloju, N., e Leung, F. “Assisting novice analysts in developing quality conceptual models with UML”. In *Communications of the ACM*, v. 49, pp. 108–112, 2006.
- Bolloju, N., Schneider, C., e Vogel, D. “Asymmetrical effects of using positive and negative examples on object modeling”. In: *18th International Conference on Information Systems Development*, pp 85-96, 2009.
- Bonwell, C. C., e Eison, J. A. “Active learning: Creating excitement in the classroom”. In Washington, DC: School of Education and Human Development, 1991.
- Booch G. “Object-Oriented Analysis and Design with Applications”. In Benjamin/Cummings, 1994.
- Booch, G., Rumbaugh, J., e Jacobson, I. “Modeling Language User Guide”. In *The Addison-Wesley Objectc Technology Series (2nd Edition)*, 2005.
- Borrego, M., Cutler, S., Prince, M., Henderson, C., e Froyd, J. E. “Fidelity of implementation of research-based instructional strategies in engineering science courses”. In *Journal of Engineering Education*, vol. 102, pp. 394–425, 2013.
- Boud, D., e Feletti, G. *The challenge of problem-based learning*, in Psychology Press, London: Kogan Page, 1997.
- Boud, D., Feletti, G. “The challenge of problem-based learning”. London: Kogan Page, 1998.
- Briggs, T. “Techniques for active learning in cs courses”. In *Journal of Computing Sciences in Colleges*, v. 21, n° 2, pp. 156–165, 2005.
- Brooks, D. “Active learning classrooms: The top strategic technology for 2017”. Acessado em 12 de Dezembro de 2019. Link: <<https://er.educause.edu/blogs/2017/3/active-learning-classrooms-the-top-strategic-technology-for-2017>>.
- Brown, J. D. “Choosing the right type of rotation in PCA and EFA.” In *JALT Testing & Evaluation SIG Newsletter*, vol. 13, no. 3, pp. 20-25, 2009.
- C. Thevathayan and M. Hamilton, Imparting software engineering design skills, in *Proc. Nineteenth Australasian Computing Education Conference*, pp. 95-102, 2017.

- Capuano, N., Pierri, A., Salerno, S., e Mangione, G. R. “Combining Individualization and Intuitive Guided Learning through Compound Learning Resources”. In *Proc. 6th Int. Conf. on Complex, Intelligent, and Software Intensive Systems*, pp. 905-910, 2012.
- Cattaneo, K. H. “Telling active learning pedagogies apart: From theory to practice.” In *Journal of New Approaches in Educational Research*, vol. 6, no. 2, pp. 144-152, 2017.
- Chen, J., Lu, H., An, L., e Zhou, Y. “Exploring teaching methods in software engineering education”. In 4th International Conference on Computer Science & Education (ICCSE'09), pp. 1733-1738, 2009.
- Chi, M. T., e Wylie, R. “The ICAP framework: Linking cognitive engagement to active learning outcomes”. In *Educational Psychologist*, vol. 49, no. 4, pp. 219-43, 2014.
- Cockburn, A. “*Writing effective use cases*”. Addison-Wesley Professional, 2000.
- Colucci, E. “Focus Groups can be fun’: the use of activity-oriented questions in Focus Group discussions,” In *Qualitative Health Research*, pp. 1422-1433, 2008.
- Connolly, T. M., Stansfield, M., e Hailey, T. “An application of games-based learning within software engineering”. In *British Journal of Educational Technology*, vol. 38, no. 3, pp. 416–428, 2007.
- Cooper, K., Liddle, S., Dascalu, S.: Experiences using defect checklists in software engineering education. In Proceedings of the ISCA 18th International Conference on Computer Applications in Industry and Engineering (CAINE 2005), pp. 402–409, 2005.
- Cronbach, L. J. “Essentials of psychological testing (5th ed.),” New York: Harper Collins Publishers, 1990.
- Da Silva, A. R. “Model-driven engineering: A survey supported by the unified conceptual model”. In *Computer Languages, Systems & Structures*, 43, 139-155, 2015.
- De França, B. B. N., Ribeiro, T. V., dos Santos, P. S. M., Travassos, G. H. “Using focus group in software engineering: lessons learned on characterizing software technologies in academia and industry”. In XVIII Ibero-American Conference on Software Engineering – Workshop de Engenharia de Software Experimental (CIbSE-ESELAW 2015), pp. 351-364 2015.
- De Freitas, S. A. A., Silva, W. C., Marsicano, G. “Using an Active Learning Environment to Increase Students' Engagement”. In 29th International Conference on Software Engineering Education and Training (CSEET'16), pp. 232-236, 2016.

- De Weerd, S., Corthouts, F., Martens, H., Bouwen, R. “Developing professional learning environments: model and application”. In *Studies in Continuing Education*, vol. 24, no. 1, pp. 25-38, 2002.
- Dennis, A. R., Hayes, G. S., Daniels Jr, R. M. “Business process modeling with group support systems”. In *Journal of Management Information Systems*, vol. 15, no. 4, pp. 115-142, 1999.
- Dennis, A. R., Hayes, G. S., Daniels Jr, R. M. “Re-engineering business process modeling”. In *Proceedings of the Twenty-Seventh Hawaii International Conference on System Sciences*, pp., 244 – 253, 1994.
- Dennis, S., Bowe, M., Ball, J., e Jensen, D. “A student-developed teaching demo of an automatic transmission”, In *Proceedings of the 2001 American Society for Engineering Education Annual Conference and Exposition*, pp. 104.1-104.13, 2001.
- Deterding, S., Sicart, M., Nacke, L., O’Hara, K., e Dixon D. “Gamification: Using Game Design Elements in Non-Gaming Contexts”. In *Annual Conference Extended Abstracts on Human Factors in Computing Systems*, p. 2425-2428, 2011.
- DeVellis, R. F. “*Scale development: Theory and application* (Vol. 26)”. Thousand Oaks, CA, USA: Sage,, 2016.
- Dewey, J. “Democracy and Education: An introduction to the philosophy of education,” The Macmillan Company, New York, 1916.
- Dewey, J. “Experience and education”. In: *The Educational Forum*, vol. 50, pp. 241–252 (1986).
- Dranidis, D., Stamatopoulou, I., e Ntika, M. “Learning and Practicing Systems Analysis and Design with StudentUML”. In *7th Balkan Conference on Informatics Conference*, pp 41-50, 2015.
- Eickholt, J. “Barriers to Active Learning for Computer Science Faculty”. In *arXiv preprint*, pp 1-13, 2018.
- Eickholt, J. “Barriers to Active Learning for Computer Science Faculty”. *arXiv preprint arXiv:1808.02426*, 2018.
- Ekahitanond, V. “Promoting university students’ critical thinking skills through peer feedback activity in an online discussion forum,” *Alberta Journal of Educational Research*, vol. 59, no. 2, pp. 247-265, 2013.

- Emara, M., Rajendran, R., Biswas, G., Okasha, M., Elbanna, A.A. “Do students’ learning behaviors differ when they collaborate in open-ended learning environments?”. In *Proceedings of the ACM on Human-Computer Interaction 2(CSCW)*, pp. 49, 2018.
- Erickson, F. “Taught cognitive learning in its immediate environments: A neglected topic in the anthropology of education”. In *Anthropology & Education Quarterly*, vol. 13, no. 2, pp. 149-180, 1982.
- Exposito, E. “yPBL methodology: A problem-based learning method applied to Software Engineering”. In *Proceeding IEEE Education Engineering Conference (EDUCON)*, pp. 1577–1583, 2010.
- Fabrigar, L. R., Wegener, D. T., MacCallum, R. C., & Strahan, E. J. “Evaluating the use of exploratory factor analysis in psychological research”. *Psychological methods*, vol 4, no. 3, pp. 272, 1999
- Fernandes Matsubara, P. G. Dealing with software estimates distortions from the perspective of negotiation theories. *ACM SIGSOFT Software Engineering Notes*, vol. 44, no. 3, pp. 22-22, 2019.
- Fernandez, A., Insfran, E., e Abrahão, S. “Usability evaluation methods for the web: A systematic mapping study”. In *Information and software Technology*, vol. 53, no. 8, pp. 789–817, 2011.
- Fernández-Sáez, A. M., Caivano, D., Genero, M., e Chaudron, M. R. “On the use of UML documentation in software maintenance: Results from a survey in industry”. In *2015 ACM/IEEE 18th International Conference on Model Driven Engineering Languages and Systems (MODELS)*, pp. 292-301, 2015.
- Ferreira, T., Viana, D., Fernandes, J., e dos Santos, R. P. “Identifying emerging topics and difficulties in software engineering education in Brazil.” In *SBES – Education Track*, pp. 230-239, 2018.
- Fioravanti, M. L., Sena, B., Paschoal, L. N., Silva, L. R., Allian, A. P., Nakagawa, E. Y., Souza, S. R., Isotani, S., e Barbosa, E. F. “Integrating Project Based Learning and Project Management for Software Engineering Teaching: An Experience Report”. In *Proc. of the 49th ACM Technical Symposium on Computer Science Education*, pp. 806-811, 2018.
- Fleiss, J.L., Levin, B., e Paik, M.C. “Statistical methods for rates and proportions”. In John Wiley & Sons (2013).

- Fraser, J. M., Tinman, A. L., Miller, K., Dowd, J. E., Tucker, L., e Mazur, E. “Teaching and physics education research: bridging the gap”. In *Reports on Progress in Physics*, vol, 77, no. 3, pp.1–17, 2014.
- Freeman, S., Eddy, L. S., McDonough, M., Smith, M. K., Okoroafor, N., Jordt, H., e Wenderoth, M. P. “Active learning increases student performance in science, engineering, and mathematics”. In *Proceedings of the National Academy of Sciences*, vol. 111, no. 23, pp., 8410–8415, 2014.
- Freeman, S., Eddy, S. L., McDonough, M., Smith, M. K., Okoroafor, N., Jordt, H., & Wenderoth, M. P. “Active learning increases student performance in science, engineering, and mathematics.” In *Proceedings of The National Academy of Sciences of the United States of America*, vol. 111, no. 23, pp. 8410-8415, 2014
- Froyd, J. E., Borrego, M. Cutler, S., Henderson, C., Prince, M. J. “Estimates of use of research-based instructional strategies in core electrical or computer engineering courses”. In *IEEE Transactions on Education*, vol. 56, no. 4, pp. 393–399, 2013.
- Garcia, F. J., e Moreno, M. N. “Software modeling techniques for a first course in software engineering: A workshop-based approach”. In *Transactions on Education*, vol. 47, no. 2, pp. 180-187, 2004.
- García-Holgado, A., García-Peñalvo, F. J., e Rodríguez-Conde, M. J. “Pilot experience applying an active learning methodology in a Software Engineering classroom.” In *Proc. EDUCON*, pp. 940-947, 2018.
- Gary, K. “Project-based learning”. In *Computer*, vol. 48, no. 9, pp. 98–100, 2015.
- Gorschek, T., Tempero, E., Angelis, L. “On the use of software design models in software development practice: An empirical investigation”. In *Journal of Systems and Software*, vol. 95, pp. 176–193, 2014.
- Granda, M., Condori-Fernández, N., Vos, T., e Pastor, O. “What do we know about the defect types detected in conceptual models?”. In *9th International Conference on Research Challenges in Information Science (RCIS'15)*, pp. 88-99, 2015.
- Grissom, S. Mccauley, R. Murphy, L. “How student centered is the computer science classroom? A survey of college faculty”. In *ACM Transactions on Computing Education (TOCE)*, vol 18, no. 1, pp. 5, 2017.
- Guedes, G. T. (2008). *UML: uma abordagem prática*. Novatec Editora.

- Guo, M., Zhang, C., Wang, F. “What is the further evidence about uml?-a systematic literature review”. In 2017 24th Asia-Pacific Software Engineering Conference Workshops (APSECW), pp. 106–113, 2017.
- Hainey, T., Connolly, T.M., Stansfield, M., e Boyle, E.A. “Evaluation of a game to teach requirements collection and analysis in software engineering at tertiary education level”. In Computers & Education, vol. 56, no. 1, pp. 21–35, 2011.
- Hausmann, R. G., Chi, M. T., e Roy, M. Learning from collaborative problem solving: An analysis of three hypothesized mechanisms. In Proceedings of the annual meeting of the cognitive science society, vol. 26, no. 26, 2004.
- Hazzan, O., e Kramer, J. “Abstraction in computer science & software engineering: A pedagogical perspective”. In Frontier Journal, v. 4, n° 1, pp. 6-14, 2007.
- Hevner, A. R. “A Three Cycle View of Design Science Research”. In *Scandinavian J. Inf. Syst*, vol 19, no. 2, pp. 87–92, 2007.
- Hevner, A. R., e Chatterjee, S. “Design Research in Information Systems”. In Ed. Springer, New York, 2010.
- Hilburn, T.B., Towhidnejad, M., Salamah, S.: Read before you write. In: Proceedings of the 24th Conference on Software Engineering Education and Training (CSEE&T), pp. 371–380, 2011.
- Holmes, R., Allen, M., & Craig, M. “Dimensions of experientialism for software engineering education”. In *Proceedings of the 40th International Conference on Software Engineering: Software Engineering Education and Training*, pp. 31-39, 2018.
- Houston, D.X., Mackulak, G.T., e Collofello, J.S. “Stochastic simulation of risk factor potential effects for software development risk management”. In Journal of Systems and Software, vol 59, no. 3, pp. 247–257, 2001.
- I. Hadar and E. Hadar, “An iterative methodology for teaching object oriented concepts,” Inform. Educ., vol. 6, no. 1, pp. 67–80, Jan. 2007
- Järvelä, S., Kirschner, P. A., Panadero, E., Malmberg, J., Phielix, C., Jaspers, J., e Järvenoja, H. “Enhancing socially shared regulation in collaborative learning groups: designing for CSCL regulation tools”. In *Educational Technology Research and Development*, vol. 63, no. 1, pp. 125-142, 2015.
- Jolliffe, I. “Principal component analysis,” Springer Berlin Heidelberg, pp. 1094-1096, 2011.

- Jorge, F. D. F., Bezerra, C. I., Coutinho, E. F., Monteiro, J. M., e Andrade, R. M. “A evolução do jogo itestlearning para o ensino das atividades de execução de testes de software”. In *XX Conferência Internacional sobre Informática na Educação (TISE 2015)-Nuevas Ideas en Informática Educativa TISE*, 2015.
- Junhua, M. L. “Innovations in Software Engineering Education: An Experimental Study of Integrating Active Learning and Design-based Learning”. In American Society for Engineering Education, pp. 01-45, 2014.
- Karamustafaoglu, O. “Active Learning Strategies in Physics Teaching,” In *Energy Education Science and Technology*, vol. 1, no. 1, pp. 27-50, Jan 2009.
- Karlsson, L., Thelin, T., Regnell, B., Berander, P., e Wohlin, C. “Pair-wise comparisons versus planning game partitioning—experiments on requirements prioritization techniques”. In *Empirical Software Engineering*, vol. 12, no. 1, pp. 3–33, 2007.
- Keeney-Kennicutt, W., Baris Gunersel, A., e Simpson, N. “Overcoming student resistance to a teaching innovation”. In *International Journal for the Scholarship of Teaching and Learning*, vol. 2, no. 1, pp. 5-, doi:10.20429/ijstl.2008.020105
- Kerrigan, J. “Active Learning Strategies for the Mathematics Classroom”. In *College Teaching*, vol. 66, no. 1, pp. 35-36, 2018.
- Kim, M.-S. “A team building algorithm based on successful record for capstone course”. In *Global Journal of Engineering Education*, pp. 243–248, 2017.
- King, A. “From sage on the stage to guide on the side”. In *College Teaching*, vol 41, pp. 30–35, 1993.
- Kinjo, T., Ohgame, Y., e Hazeyama, A. “An Object-Oriented Modeling Learning Support System Using Inspection Comments”. In *Journal of Information and Systems in Education*, vol. 5, no. 1, pp. 66–75, 2006.
- Kirschner, F., Paas, F., & Kirschner, P. A. Individual and group-based learning from complex cognitive tasks: Effects on retention and transfer efficiency. *Computers in Human Behavior*, vol. 25, no. 2, pp. 306-314, 2009.
- Kirschner, F., Paas, F., e Kirschner, P. A. “Individual and group-based learning from complex cognitive tasks: Effects on retention and transfer efficiency”. In *Computers in Human Behavior*, vol. 25, no. 2, pp. 306-314, 2009.

- Kitchenham, B. A., e Charters, S. “Guidelines For Performing Systematic Literature Reviews in Software Engineering”. In EBSE Technical Report EBSE-2007-01, 2007.
- Knauss, E., Schneider, K., e Stapel, K. “A game for taking requirements engineering more seriously”. In 2008 Third International Workshop on Multimedia and Enjoyable Requirements Engineering-Beyond Mere Descriptions and with More Fun and Games, pp. 22–26, 2008.
- Knight, R. D. “Five Easy Lessons: Strategies for Successful Physics Teaching”, Addison-Wesley, San Francisco, CA, 2004.
- Knobloch, J., Kaltenbach, J., e Bruegge, B. “Increasing student engagement in higher education using a context-aware Q&A teaching framework”. In *Proc. 40th ICSE-SEET*, pp. 136-145, 2018.
- Kolb, D.A.: *Experiential learning: Experience as the source of learning and development*. FT press (2014)
- Kontio, J., Lehtola, L., e Bragge, J. “Using the focus group method in software engineering: obtaining practitioner and user experiences”. In *International Symposium on Empirical Software Engineering (ISESE’04)*, pp. 271-280, 2004.
- Kopp, V., Stark, R., e Fischer, M. R. “Fostering diagnostic knowledge through computer-supported, case-based worked examples: effects of erroneous examples and feedback”. In *Journal of Medical education*, v. 42, n° 8, pp. 823-829, 2008.
- Koziolk, H., e Firus, V. “Empirical evaluation of model-based performance prediction methods in software development”. In *Quality of Software Architectures and Software Quality*, pp. 188-202, 2005.
- Kruus, H., Robal, T., e Jervan, G. “Teaching modeling in SysML/UML and problems encountered”. In 2014 25th EAEEIE Annual Conference (EAEEIE), pp. 33-36, 2014.
- Kuhrmann, M., Fernández, D.M., e Daneva, M. “On the pragmatic design of literature studies in software engineering: an experience-based guideline”. In *Empirical software engineering*, vol. 22, no. 6, pp. 2852–2891, 2017.
- Kurkovsky, S., Ludi, S., e Clark, L. “Active Learning with LEGO for Software Requirements”. In *50th ACM Technical Symposium on Computer Science Education*, pp. 218-224, 2019.
- Kuzniarz, L., e Börstler, J. “Teaching modeling: An initial classification of related issues”. In *Proceedings of the 7th Educators Symposium @ MODELS 2011*, pp. 1–10, 2011.

- Kuzniarz, L., e Staron, M. “Best practices for teaching UML-based software development”. In *Satellite Events at the MoDELS Conference*, pp. 320–332, 2006.
- Lacave, C., Molina, A. I., e Redondo, M. A. “A Preliminary Instrument for Measuring Students’ Subjective Perceptions of Difficulties in Learning Recursion”. In *IEEE Transactions on Education*, vol. 61, no. 2, pp. 119-126, 2017.
- Lai, C. L., e Hwang, C. J. “A self-regulated flipped classroom approach to improving students’ learning performance in a mathematics course”. In *Computers & Education*, vol. 100, pp. 126- 140, 2016.
- Laitenberger, O., & Dreyer, H. M. “Evaluating the usefulness and the ease of use of a web-based inspection data collection tool”. In *Proceedings Fifth International Software Metrics Symposium. Metrics (Cat. No. 98TB100262)*, pp. 122-132, 1998.
- Lin, Y. T. “Impacts of a flipped classroom with a smart learning diagnosis system on students’ learning performance, perception, and problem-solving ability in a software engineering course”. In *Computers in Human Behavior*, vol. 95, pp. 187-196, 2019.
- Lindland, O.I., Sindre, G., e Solvberg, A. “Understanding quality in conceptual modeling.” *IEEE software*, vol. 11, no. 2, pp. 42–49, 1994.
- Linsey, J., Tseng, I., Fu, K., Cagan, J., Wood, K., Schunn, C. “A study of design fixation, its mitigation and perception in engineering design faculty.” *ASME Journal of Mechanical*, v. 132, no. 4, pp. 041003-1- 041003-12, 2010
- Liu, E., Lin, S., Chiu, C., e Yuan, S. “Web-based peer review: The learner as both adapter and reviewer” In *Transactions on Education*, vol. 44, no. 3, pp. 246-251, 2001.
- Lopes, A., Steinmacher, I., e Conte, T. “UML Acceptance: Analyzing the Students' Perception of UML Diagrams”. In *Proceedings of the XXXIII Brazilian Symposium on Software Engineering*, pp. 264-272, 2019.
- Ma, Z. “An approach to improve the quality of object-oriented models from novice modelers through project practice”. In *Frontiers of Computer Science*, vol. 11, no. 3, pp. 485-498, 2017.
- Manotas, I., Bird, C., Zhang, R., Shepherd, D., Jaspan, C., Sadowski, C., Clause, J. An empirical study of practitioners' perspectives on green software engineering. In 38th International Conference on Software Engineering, pp. 237-248, 2016.

- Marques, M. R., Quispe, A., e Ochoa, S. F. “A systematic mapping study on practical approaches to teaching software engineering”. In IEEE Frontiers in Education Conference (FIE), pp. 1–8, 2014.
- McCrum-Gardner, E.: Which is the correct statistical test to use? *British Journal of Oral and Maxillofacial Surgery* vol. 46, no. 1, pp. 38–41, 2008.
- Medeiros, A. L., Fernandes, L., Batista, T., & Minora, L. Requisitos e Arquitetura de Software Orientada a Aspectos: Uma Integração Sinérgica. *Anais do XXI Simpósio Brasileiro de Engenharia de Software*, pp. 199-215, 2007.
- Mendes, J., Costa, Y., Frazão, K., Santos, R., Santos, D., & Rivero, L. “Identificação das Expectativas e Dificuldades de Alunos de Graduação no Ensino de Engenharia de Software.” In *Anais do XXVII Workshop sobre Educação em Computação*, pp. 334-347, 2019.
- Michael, J. (2007). Faculty perceptions about barriers to active learning. *College Teaching*, (2), 42
- Michael, J. “Faculty perceptions about barriers to active learning”. In *College teaching*, vol. 55, no. 2, pp. 42-47, 2007.
- Michael, J. “Where's the evidence that active learning works?” In *Advances in Physiology Education*, vol 30, no. 4, pp. 159-167, 2006.
- Misbhauddin, M., Alshayeb, M. “An integrated metamodel-based approach to software model refactoring”. *Software & Systems Modeling*, pp. 1-38, 2017.
- Moisan, S., e Rigault, J. P. “Teaching Object-Oriented Modeling and UML to Various Audiences”. In International Conference on Model Driven Engineering Languages and Systems Workshops, pp. 40-54, 2009.
- Nascimento, E. S., Silva, W. A. F., França, B. B. N. D.; Gadelha, B; Conte, T. U. Relatório Técnico: Um Modelo sobre as Dificuldades para Especificar Casos de Uso, Relatório de número 0007 (2017a). Disponível em: <https://uses.icomp.ufam.edu.br/relatorios-tecnicos/>.
- Nascimento, E. S., Silva, W. A. F., Lopes, Steinmacher, I., Conte, T. (2018) TR-USES-2018-0008. Reducing the challenges faced when learning use cases by using inspection. Disponível em: <https://uses.icomp.ufam.edu.br/relatorios-tecnicos/>.

- Nascimento, E.S., Silva, W.A.F., de França, B.B.N., Gadelha, B., Conte, T.U.: (2017b). A model on the difficulties to specify use cases. In: Proceedings of the XX Ibero-American Conference on Software Engineering (CIBSE), pp. 263–276.
- Nguyen, P. M., Terlouw, C., Pilot, A. “Culturally appropriate pedagogy: the case of group learning in a Confucian Heritage Culture context”. In *Intercultural Education*, vol. 17, no. 1, pp. 1-19, 2006.
- Nilson, L. B. “Teaching at its best: A research-based resource for college instructors”. In 4rd Edition, San Francisco: Wiley & Sons, 2016.
- Nunamaker, J., e Briggs, R. “Toward a Broader Vision for Information Systems”. In *ACM Transactions on MIS*, vol. 2, no. 4, pp. 20:1-20:12, 2011.
- Nunes, R. R., Pedrosa, D., Fonseca, B., Paredes, H., Cravino, J., Morgado, L., e Martins, P. “Enhancing students’ motivation to learn software engineering programming techniques: a collaborative and social interaction approach”. In *Intern. Conf. on Universal Access in Human-Computer Interaction*, 2015, pp. 189- 201, 2015.
- Ocak, M. A. (2011). Why are faculty members not teaching blended courses? Insights from faculty members. *Computers & Education*, 56(3), 689-699. doi:10.1016/j.compedu.2010.10.011
- OMG, The Unified Modeling Language. Documents Associated with UML Version 2.5, Object Management Group, 2015 < <http://www.omg.org/spec/UML/2.5/>>
- Orawiwatnakul, W., e Wichadee, S. “Achieving Better Learning Performance through the Discussion Activity in Facebook,” *Turkish Online Journal of Educational Technology*, vol. 15, no. 3, pp. 1-8, Jul. 2016.
- Park, E. L., Choi, B. K. “Transformation of classroom spaces: Traditional versus active learning classroom in colleges”. In *Higher Education*, vol 68, no. 5, pp. 749–771, 2014.
- Petersen, K., Vakkalanka, S., e Kuzniarz, L. “Guidelines for conducting systematic mapping studies in software engineering: An update”. *Information and Software Technology*, vol. 64, pp. 1–18, 2015.
- Petre, M. ““No shit” or “Oh, shit!”: responses to observations on the use of UML in professional practice”. In *Software & Systems Modeling*, v.13, n° 4, pp. 1225-1235, 2014.
- Petri, G., e Chiavegatti, N. C. (2015). “Um Role Playing Game para o Ensino de Elicitação e Análise de Requisitos”. In *Revista Novas Tecnologias na Educação*, v.13, n°1, p. 01-10.

- Port, D., e Kazman, R. “Laptop Enabled Active Learning in the Software Engineering Classroom: An Experience Report”. In *Proc. of the 20th CSEET*, pp. 262-274, 2007.
- Portillo, J. A. P. S., e Campos, P. G. “The jigsaw technique: experiences teaching analysis class diagrams”. In *2009 Mexican International Conference on Computer Science*, pp. 289-293, 2009.
- Prikladnicki, R., Albuquerque, A. B., von Wangenheim, C. G., e Cabral, R. “Ensino de engenharia de software: Desafios, estratégias de ensino e lições aprendidas”. In Fórum de Educação em Engenharia de Software (FEES), 2009.
- Prince, M. “Does active learning work? A review of the research”. In *Journal of Engineering Education*, v. 93, pp. 223–231, 2004.
- Pundak, D., e Rozner, S. “Empowering engineering college staff to adopt active learning methods”. In *Journal of Science Education and Technology*, vol. 17, no. 2, pp. 152-63, 2008.
- Quezada-Sarmiento, P. A., Enciso, L., Mayorga-Diaz, M. P., Mengual-Ándres, S., Hernandez, W., Vivanco-Ochoa, J. V., e Carrión, P. V. T. “Promoting innovation and entrepreneurship skills in professionals in software engineering training: An approach to the academy and bodies of knowledge context.” In *2018 IEEE Global Engineering Education Conference (EDUCON)* , pp. 796-799, 2018.
- Renkl, A. “Toward an instructionally oriented theory of example-based learning”. In *A Multidisciplinary Journal of Cognitive Science*, v. 38, nº 1, pp. 1-37, 2014.
- Richardson, I. , e Delaney, Y. “Problem based learning in the software engineering classroom,” in *22th Conference on Software Engineering Education and Training (CSEET 2009)*, pp. 174-181, 2009.
- Richardson, I., e Delaney, Y. “Problem based learning in the software engineering classroom”. In *22th Conference on Software Engineering Education and Training (CSEET'09)*, pp. 174-181, 2009.
- Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F. e Lorensen, W.: “Object-Oriented Modeling and Design”. Prentice Hall, 1991.
- Rumbaugh, J., Jacobson, I., Booch, G. “Unified Modeling Language Reference Manual”. Pearson Higher Education (2nd Edition), 2004.
- Runeson, P., Host, M. Guidelines for conducting and reporting case study research in software engineering. In *Empirical Software Engineering*, 14, n. 2, pp. 131-164, 2008.

- Sancho-Thomas, P., Fuentes-Fernández, R., e Fernández-Manjón, B. “Learning teamwork skills in university programming courses”. *Computers Education*, vol. 53, no. 2, pp. 517–531, 2009.
- Santos, G., Rocha, A. R., Conte, T., Barcellos, M. P., Prikładnicki, R., 2012. Strategic Alignment between Academy and Industry: A Virtuous Cycle to Promote Innovation in Technology. In 26° Simpósio Brasileiro de Engenharia de Software (SBES 2012), v. 1. p. 196-200.
- Sauer, C., Jeffery, D.R., Land, L., Yetton, P.: The effectiveness of software development technical reviews: A behaviorally motivated program of research. *IEEE Transactions on Software Engineering*, vol. 26, no. 1, pp. 1–14, 2000.
- Savi, R., Wangenheim, C., e Borgatto, A. “Um Modelo de Avaliação de Jogos Educacionais na Engenharia de Software”. In XXV Simpósio Brasileiro de Engenharia de Software (SBES 2011), pp. 194-203, 2011.
- Scanniello, G., e Erra, U. “Distributed modeling of use case diagrams with a method based on think-pair-square: Results from two controlled experiments,” In *Journal of Visual Languages & Computing*, vol. 25, no. 4, pp. 494-517, 2014.
- Schilling Jr, W., e Sebern, M. J. “Teaching Software Engineering: An Active Learning Approach”. In *Proc. 119th American Society for Engineering Education*, pp. 436-442, 2012.
- Schon, D.A.: *Educating the reflective practitioner*, 1987.
- Seaman, C. B. “Qualitative methods in empirical studies of software engineering”. *Transactions on Software Engineering*, vol. 25, no. 4, pp. 557-572, 1999.
- Segedy, J.R., Kinnebrew, J.S., Biswas, G. “Using coherence analysis to characterize self-regulated learning behaviours in open-ended learning environments”. In *Journal of Learning Analytics*, vol. 2, no. 1, pp. 13–48, 2015.
- Shadle, S. E., Marker, A., Earl, B. “Faculty drivers and barriers: laying the groundwork for undergraduate STEM education reform in academic departments”. In *International Journal of STEM Education*, vol. 4, no. 8, pp. 1-13, 2017.
- Shapiro, S., Wilk, M.: An analysis of variance test for normality. *Biometrika*, vol. 52, no. 3, pp. 591–611, 1965.
- Shaw, M., e Tomayko, J. E. “Models for Undergraduate Project Courses in Software Engineering”. In *SEI Conference on Software Engineering Education*, pp. 33-71, 1991.

- Shekhar, P., Prince, M., Finelli, C., Demonbrun, M., Waters, C. “Integrating quantitative and qualitative research methods to examine student resistance to active learning”. In *European Journal of Engineering Education*, vol. 44, no. 1, pp. 6-18, 2019,
- Shull, F., Carver, J., e Travassos, G. H. “An empirical methodology for introducing software processes”. In *9th ACM SIGSOFT International Symposium on Found. of Software Engineering Notes*, pp. 288 – 296, 2001.
- Siau, K., e Loo, P-P. “Identifying difficulties in learning UML”. In *Journal of Information Systems Management*, v. 23, n. 3, pp. 43-51, 2006.
- Sien, V. Y. “An investigation of difficulties experienced by students developing unified modelling language (UML) class and sequence diagrams”. In *Computer Science Education*, vol. 21, no. 4, pp. 317-342, 2011.
- Silva, W, Gadelha, B., Steinmacher, I, e Conte, T. “Relatório Técnico com os artefatos do estudo sobre realizar práticas de modelagem individual e em grupo”. In TR-USES-2018-009, 2018a. Disponível em: <http://uses.icomp.ufam.edu.br/relatorios-tecnicos/>.
- Silva, W., Steinmacher, I., Conte, T. Material de apoio utilizado no contexto da pesquisa “OpenSMALS: Um Repositório Aberto para auxiliar no Ensino de Modelagem de Software empregando Estratégias De Aprendizagem Ativa”. In Tech. rep., USES Research Group Technical Report 02 (2020). Disponível em: <http://uses.icomp.ufam.edu.br/relatorios-tecnicos/>.
- Silva, W., Steinmacher, I., Conte, T. Students’ and instructors’ perceptions of five different active learning strategies used to teach software modeling. *IEEE Access*, 2019.
- Silva, W., Steinmacher, I., e Conte, T. “ Um jogo educacional que apoia o ensino de Diagrama de Atividades”. In Relatório Técnico TR-USES-2017-0009, 2017b. Disponível em: <http://uses.icomp.ufam.edu.br/relatorios-tecnicos/>.
- Silva, W.A.F., Steinmacher, I.F., Conte, T.U.: Is it better to learn from problems or erroneous examples? In: *Proceedings of the 30th Conference on Software Engineering Education and Training (CSEE&T)*, pp. 222–231 (2017a).
- Silvab, W., Gadelha, B., Steinmacher, I., Conte, T. “What are the differences between group and individual modeling when learning UML?.” In *Proceedings of the XXXII Brazilian Symposium on Software Engineering*, pp. 308-317, 2018b.

- Sindre, G., e Moody, D. "Evaluating the Effectiveness of Learning Interventions: An Information Systems Case Study". In *European Conference on Information Systems (ECIS'03)*, pp.80-97, 2003.
- Smith, R., e Gotel, O. "Gameplay to introduce and reinforce requirements engineering practices." In *2008 16th IEEE International Requirements Engineering Conference*, pp. 95–104, 2008.
- Sommerville, I. "Engenharia de Software". 9^a edição. Pearson Prentice Hall, 2011. 529 p.
- Souza, M. V. R., e França, C. "O que Explica o Sucesso de Jogos no Ensino de Engenharia de Software? Uma Teoria de Motivação". In *24^o Workshop sobre Educação em Computação*, p. 2255-2264, 2016.
- Steinmacher, I.; Silva, M. A. G.; Gerosa, M. A.; e Redmiles, D. F. "A systematic literature review on the barriers faced by newcomers to open source software projects". In *Information and Software Technology*, v. 59, pp. 67-85, 2015.
- Stikkolorum, D.R., Ho-Quang, T., Karasneh, B., Chaudron, M.R. "Uncovering students' common difficulties and strategies during a class diagram design process: an online experiment". In: *EduSymp@ MoDELS*, pp. 29–42, 2015.
- Stoecklin, S., Williams, D. D., e Swain, R. "Understanding object-oriented systems specifications using familiar systems". In *Proceedings. 1998 International Conference Software Engineering: Education and Practice (Cat. No. 98EX220)*, pp. 10-15, 1998.
- Störrle, H. "How are Conceptual Models used in Industrial Software Development? A Descriptive Survey". In *21th International Conference on Evaluation and Assessment in Software Engineering (EASE'17)*, pp. 160-169, 2017.
- Strauss, A., e Corbin, J. "Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory". 2 ed. SAGE Publications, London, 1998.
- Szmurło, R. e Śmiałek, M. "Teaching software modeling in a simulated project environment". In *9th International Conference on Model Driven Engineering Languages and Systems (MODELS'06)*, pp. 301-310, 2006.
- Tamai, T. "How to Teach Software Modeling". In *27th International Conference on Software Engineering*, St. Louis, Mo, USA, pp. 609-610, 2005.
- Tanaka, T., Mori, K., Hashiura, H., Hazeyama, A., e Komiya, S. "Proposals of a Method Detectiong Learners' Difficult Points in Object Modeling Exercises and a Tool to Support

- the Method”. In IIAI 3rd International Conference on Advanced Applied Informatics (IIAIAAD), pp. 650-655, 2014.
- Tharayil, S., Borrego, M., Prince, M., Nguyen, K. A., Shekhar, P., Finelli, C. J., e Waters, C. “Strategies to mitigate student resistance to active learning” In *International Journal of STEM Education*, vol. 5, no. 1, pp. 1-16, 2018.
- Tiwari, S., Gupta, A.: A systematic literature review of use case specifications research. *Information and Software Technology*, vol. 67, pp. 128–158, 2015.
- Tomer, G., e Mishra, S. K. “Professional identity construction among software engineering students: A study in India.” In *Information Technology & People*, vol. 29, no. 1, pp. 146-172, 2016.
- Travassos, G., Shull, F., Fredericks, M., Basili, V.R.: Detecting defects in object-oriented designs: using reading techniques to increase software quality. In: *ACM Sigplan Notices*, vol. 34, pp. 47–5, 1999.
- Trochim, W. M., e Donnelly, J. P. “Research methods knowledge base,” 3rd. ed., 2008. Mason, OH: Atomic Dog Publishing.
- Turpen, C., Dancy, M., Henderson, C. “Perceived affordances and constraints regarding instructors of use of Peer Instruction: Implications for promoting instructional change”. In *Physical Review Physics Education Research*, vol. 12, no. 1, pp. 2016, 010116:01-010116:18.
- von Glasersfeld, E. “Radical constructivism: A Way of Knowing and Learning,” Routledge, 2013.
- Von Wangenheim, C. G., Savi, R., e Borgatto, A.F. “SCRUMIA—An educational game for teaching SCRUM in computing courses”. In *Journal of Systems and Software*, v. 86, n. 10, pp. 2675-2687, 2013.
- Vujović, V., Maksimović, M., Perišić, B. “The different active learning strategies in Software Engineering and their effectiveness”. In *Proc. 7th ICERI2014*, pp. 3183-3193, 2014.
- Wang, Y., Su, R., Li, G. “Research on PBL and LBL Double Track Teaching Model in Unified Modeling Language Teaching Based on Outstanding Engineers”. In *Informatics and Management Science*, pp. 381-386, 2013.
- Welch, R., e Klosky, J. L. “An Online Database and User Community For Physical Models in the Engineering Classroom”, In *Advances in Engineering Education*, vol. 1, no. 1, 2007.

- Wieringa, R. "Design science as nested problem solving". In *Proc. of the 4th Int. Conf. on Design Science Research in Information Systems and Technology*, pp. 8-20, 2009.
- Wohlin, C., Aurum, A. "Towards a decision-making structure for selecting a research design in empirical software engineering". In *Empirical Software Engineering*, vol. 20, no. 6, pp. 1427-1455, 2015.
- Wöhlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., Wesslén, A. "Experimentation in software engineering". Springer Science & Business Media. 2012.
- Yohannis, A. "Gamification of Software Modelling Learning". In *Satellite Events at the MoDELS Conference*, pp. 1-7, 2016.
- Zamanzadeh, V., Rassouli, M., Abbaszadeh, A., Majd, H. A., Nikanfar, A., Ghahramanian, A. "Details of content validity and objectifying it in instrument development". In *Nursing Practice Today*, vol. 1, no. 3, pp. 163-171, 2014.
- Zehetmeier, D., Böttcher, A., Brüggemann-Klein, A., Thurner, V. "Defining the Competence of Abstract Thinking and Evaluating CS-Students' Level of Abstraction". In *Proceedings of the 52nd Hawaii International Conference on System Sciences*, 2019.
- Zellweger, F. "Overcoming Subcultural Barriers in Educational Technology Support". In *18th annual conference of the Consortium of Higher Education Researchers*, pp. 1-14, 2005.
- Zhang, H., Kitchenham, B., Pfahl, D. "Software process simulation modeling: an extended systematic review". In *International Conference on Software Process*, pp. 309-320, 2010.



**PODER EXECUTIVO
MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO AMAZONAS
INSTITUTO DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA**



**MATERIAL DE APOIO UTILIZADO NO CONTEXTO DA PESQUISA
“OPENSMAIS: UM REPOSITÓRIO ABERTO PARA AUXILIAR NO
ENSINO DE MODELAGEM DE SOFTWARE EMPREGANDO
ESTRATÉGIAS DE APRENDIZAGEM ATIVA”**

WILLIAMSON ALISON FREITAS SILVA

Manaus
Janeiro, 2020

SUMÁRIO

PARECER CONSUBSTANCIADO DO CEP.....	3
PROTOCOLO DO MSL DO MAPEAMENTO SISTEMÁTICO DA LITERATURA	10
1.1. INTRODUÇÃO.....	10
1.2. QUESTÕES DE PESQUISA	10
1.3. STRING DE BUSCA.....	11
1.4. CRITÉRIOS PARA SELEÇÃO DOS ARTIGOS.....	12
1.5. ESTRATÉGIA PARA SELEÇÃO DOS ARTIGOS	13
1.6. DEFINIÇÃO DA ESTRATÉGIA DE EXTRAÇÃO DE DADOS	14
1.7. FASE DE CONDUÇÃO	16
DEFINIÇÃO DAS MÉTRICAS	17
DIFICULDADES IDENTIFICADAS A PARTIR DOS ESTUDOS DO MSL	19
LISTA COMPLETA DE ARTIGOS SELECIONADOS.....	27
FERRAMENTAS IDENTIFICADAS NO MSL	35
MAPEAMENTO ENTRE DIFICULDADES E TECNOLOGIAS IDENTIFICADAS NO MSL	37
OPEN REPOSITORY FOR TEACHING SOFTWARE MODELING APPLYING ACTIVE LEARNING STRATEGIES: MATERIAL DE APOIO	40
1. EVOLUÇÃO DO REPOSITÓRIO ABERTO	40
ESTRATÉGIAS DE ENSINO BASEADAS EM PRÍNCÍPIOS DE APRENDIZAGEM ATIVA VISANDO O ENSINO DE DIAGRAMAS DA UML.....	55

PARECER CONSUBSTANCIADO DO CEP



UNIVERSIDADE FEDERAL DO
AMAZONAS - UFAM



PARECER CONSUBSTANCIADO DO CEP

DADOS DO PROJETO DE PESQUISA

Título da Pesquisa: Investigando o uso de métodos de aprendizagem ativa durante o processo de ensino de modelos UML

Pesquisador: WILLIAMSON ALISON FREITAS SILVA

Área Temática:

Versão: 1

CAAE: 84445918.9.0000.5020

Instituição Proponente: Instituto de Computação

Patrocinador Principal: Financiamento Próprio

DADOS DO PARECER

Número do Parecer: 2.545.694

Apresentação do Projeto:

A Unified Modeling Language (UML) é um padrão para a modelagem que tem obtido ampla utilização na indústria de software. No entanto, estudantes encontram dificuldades para aprender a modelar os diagramas UML de forma correta e completa. Uma das razões é a maneira como a UML vem sendo ensinada. Para tornar a aprendizagem mais efetiva é necessário utilizar métodos em que os alunos desempenhem um papel mais ativo na aprendizagem. Diante desse contexto, este projeto tem por objetivo investigar os métodos de aprendizagem ativa que são mais adequados para o ensino de modelos da UML. Para isto, será realizada uma revisão sistemática da literatura com o objetivo de investigar quais as dificuldades percebidas pelos estudantes enquanto estes projetam os modelos UML e identificar os métodos de ensino propostos com o objetivo de minimizar estas dificuldades. Além disso, será proposta uma abordagem de ensino empregando princípios de aprendizagem ativa para apoiar o professor durante o processo de ensino de modelos UML e será realizado um estudo de caso com estudantes, maiores de 18 anos, visando compreender a influência da abordagem durante o processo de aprendizagem de modelos da UML. O resultado deste estudo visa identificar os fatores que influenciam positivamente e negativamente no aprendizado dos estudantes ao empregar a abordagem proposta.

Objetivo da Pesquisa:

Objetivo Primário:

Endereço: Rua Teresina, 495

Bairro: Adrianópolis

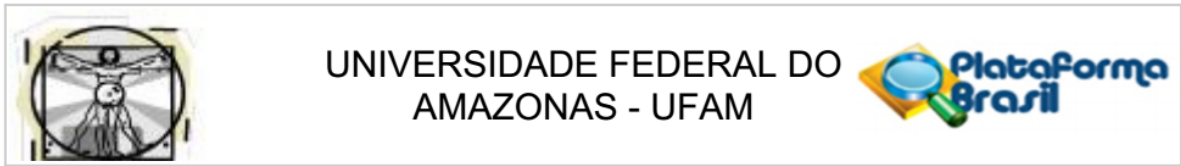
UF: AM

Município: MANAUS

Telefone: (92)3305-1181

CEP: 69.057-070

E-mail: cep.ufam@gmail.com



Continuação do Parecer: 2.545.694

O objetivo principal deste projeto consiste propor uma abordagem, que será composto por um conjunto de métodos de ensino com foco na aprendizagem ativa, para apoiar o ensino e facilitar o aprendizado dos estudantes em relação aos modelos da UML.

Objetivo Secundário:

- Identificação de dificuldades e oportunidades para ensinar e para construir os modelos, a partir de um mapeamento sistemático da literatura.
- Identificação de um conjunto de métodos de ensino com foco na aprendizagem ativa que podem apoiar o processo de ensino dos modelos UML.
- Desenvolvimento e adaptação de uma abordagem para o ensino de modelos que se adequem ao desenvolvimento de sistemas interativos com foco na qualidade de uso.

Avaliação dos Riscos e Benefícios:

Riscos:

Toda pesquisa realizada com seres humanos apresenta riscos. Como esta pesquisa é feita através da aplicação de questionários com estudantes, alguns riscos que podem ocorrer:

O participante pode se sentir cansado ou aborrecido ao responder os questionários, desconfortável ou constrangido durante a realização a gravação de áudio e vídeo. A fim de minimizar estes riscos, o pesquisador se compromete em esclarecer no início do estudo o conteúdo da pesquisa e a duração prevista para realizar a coleta dos dados. Além disso, será enfatizado que a participação é voluntária e será garantido ao participante o direito de não responder a qualquer pergunta, assim como de se retirar e encerrar a sua participação no estudo a qualquer momento sem qualquer tipo de penalização. Posteriormente à coleta de dados, o participante poderá ainda solicitar a exclusão do conteúdo fornecido, de forma integral ou parcial.

O participante pode pensar que não será mantida a sua confidencialidade. A fim de minimizar este risco, os pesquisadores também se comprometem a manter sigilo e confidencialidade a respeito de informações coletadas dos participantes, tanto no relatório final da pesquisa quanto em qualquer publicação que venha a ser feita sobre as descobertas do estudo. O pesquisador e o

Endereço: Rua Teresina, 495	CEP: 69.057-070
Bairro: Adrianópolis	
UF: AM Município: MANAUS	
Telefone: (92)3305-1181	E-mail: cep.ufam@gmail.com



UNIVERSIDADE FEDERAL DO
AMAZONAS - UFAM



Continuação do Parecer: 2.545.694

estudantes assinarão o termo de consentimento

livre e esclarecido (TCLE) associado a cada atividade, assegurando aos estudantes que os dados serão confidenciais. Com relação, ao vazamento

de dados informados durante o estudo, seja em forma de texto, imagem, áudio, vídeo, etc, os pesquisadores informarão aos estudantes que as

informações coletadas durante a pesquisa serão armazenadas em local seguro, sob a sua responsabilidade.

Além disso, o pesquisador ao realizar a

análise dos dados coletados em conjunto com outro(s) pesquisador(es) não deverá (ão) divulgar as identidades dos participantes. Somente os

pesquisadores mencionados no protocolo de pesquisa terão acesso aos dados individuais de cada participante. Será descrito também no TCLE que

não serão utilizados os nomes dos participantes para a identificação do arquivo nem seus nomes serão inseridos no conteúdo do documento.

O participante pensar que haverá uma compreensão incorreta dos questionamentos e respostas. Visando evitar a compreensão incorreta destas

informações, será esclarecido ao início da coleta de dados que quaisquer dúvidas serão sanadas pelo responsável pela coleta durante a mesma.

Além disso, o responsável pela coleta também pedirá autorização ao respondente para que dúvidas sejam esclarecidas durante a coleta.

Além dos pontos acima, vale ressaltar que todos os materiais necessários para a realização da pesquisa serão fornecidos pelos próprios

pesquisadores. Caso o participante venha a ter alguma despesa decorrente da sua participação na pesquisa, haverá o ressarcimento em espécie,

logo após ao término da pesquisa. Caso seja necessário, os participantes terão direito a indenizações e cobertura material para reparação aos

danos causados decorrentes da sua participação na pesquisa, esta indenização também será realizada via pagamento em espécie, logo após ao

término da pesquisa.

Benefícios:

Através dos resultados deste estudo, espera-se contribuir na aprendizagem dos participantes através do uso dos métodos de aprendizagem ativa,

melhorando assim sua formação acadêmica e profissional. Os benefícios desta pesquisa para os participantes são:

Endereço: Rua Teresina, 495

Bairro: Adrianópolis

UF: AM

Município: MANAUS

Telefone: (92)3305-1181

CEP: 69.057-070

E-mail: cep.ufam@gmail.com



UNIVERSIDADE FEDERAL DO
AMAZONAS - UFAM



Continuação do Parecer: 2.545.694

Fornecer aos participantes a oportunidade de praticar e aplicar os conhecimentos e habilidades adquiridas em sala de aula ao solucionar problemas e projetos reais.

Melhorar a capacidade dos participantes em compreender e projetar os modelos da UML de forma correta e completa.

Melhorar a formação dos estudantes para a indústria de software, tornando-os profissionais qualificados capazes de solucionar problemas e empregar os conceitos aprendidos em modelagem de software.

Melhorar as habilidades de trabalho em equipe dos participantes, seja no contexto acadêmico ou profissional.

Comentários e Considerações sobre a Pesquisa:

Metodologia Proposta:

Para alcançar os objetivos relacionados a este projeto será utilizada uma adaptação da metodologia experimental proposta em Mafra et al. (2006), que é uma extensão da metodologia de Shull et al. (2001) para o desenvolvimento e avaliação de tecnologias de software. A metodologia proposta por Shull et al. (2001) permite evoluir uma tecnologia emergente desde sua definição até a sua transferência para o contexto final. Mafra et al. (2006) identificaram a necessidade de executar estudos secundários antes dos estudos primários sugeridos em Shull et al. (2001), para que a definição da nova tecnologia pudesse estar baseada em evidências da literatura. Os estudos primários são aqueles estudos que o avaliador realiza para gerar novo conhecimento, isto é levantar hipóteses e prová-las. Já os estudos secundários não geram evidências, porém, visam identificar, avaliar, e interpretar todos os resultados relevantes a um determinado tópico de pesquisa, fenômeno de interesse ou questão de pesquisa. Nesta visão, a metodologia de Mafra et al. (2006) é dividida em duas partes, sendo que a primeira parte contém a extensão proposta, onde são realizados os estudos secundários, e a segunda contém a metodologia original definida por Shull et al. (2001), onde serão realizados os estudos primários. Para execução dos estudos primários, serão selecionados estudantes de graduação, maiores de 18 anos, regularmente matriculados em um Curso Superior em Ciência da Computação/ Sistemas de Informação/ Engenharia da Computação da

Endereço: Rua Teresina, 495

Bairro: Adrianópolis

UF: AM

Município: MANAUS

Telefone: (92)3305-1181

CEP: 69.057-070

E-mail: cep.ufam@gmail.com



UNIVERSIDADE FEDERAL DO
AMAZONAS - UFAM



Continuação do Parecer: 2.545.694

UFAM ou outra instituição de ensino do Brasil. Como instrumentos para coletas de dados dos estudos primários, serão utilizados questionários. Os estudantes responderam estes questionários após a realização do estudo em seu ambiente acadêmico. A seguir são descritos os passos metodológicos que servirão de base para compor a metodologia de pesquisa deste projeto.

Execução de estudos secundários – foram executadas revisões da literatura por meio de mapeamentos e revisões sistemáticas com o objetivo de criar uma base de conhecimento sólida para a pesquisa.

Design da Solução – está sendo elaborada uma proposta de solução tendo por base o conhecimento resultante dos estudos secundários baseados na literatura.

Execução de estudos primários – será realizada uma investigação, com estudantes de graduação, maiores de 18 anos, com o objetivo de (a) avaliar se a aplicação da abordagem de ensino é viável; (b) adquirir uma compreensão refinada sobre a abordagem; (c) identificar se a aplicação da abordagem de ensino possui alguma interação negativa; e (d) identificar os objetivos de aprendizagem que os estudantes podem alcançar após a aplicação da abordagem de ensino. Vale ressaltar que todos os estudos serão realizados com estudantes de graduação, com idades acima de 18 anos, que estejam regularmente matriculados em algum curso de Computação. Todos os estudantes, caso seja necessário, receberão treinamentos em forma de aulas expositivas, onde poderão esclarecer suas dúvidas sobre os materiais e sobre a abordagem que será utilizada.

Critério de Inclusão:

Os critérios de inclusão e exclusão foram desenvolvidos com base na Resolução 466/12. Abaixo são descritos os critérios de inclusão dos participantes do estudo:

1. Os participantes deverão ser estudantes de graduação regularmente matriculados em um Curso Superior em Ciência da Computação/ Sistemas de Informação/ Engenharia da Computação da UFAM ou outra instituição de ensino do Brasil.

Endereço: Rua Teresina, 495

Bairro: Adrianópolis

UF: AM

Município: MANAUS

Telefone: (92)3305-1181

CEP: 69.057-070

E-mail: cep.ufam@gmail.com



Continuação do Parecer: 2.545.694

Considerações sobre os Termos de apresentação obrigatória:

1. Projeto de Pesquisa: Apresentado.
2. TCLE - Termo de Consentimento Livre e Esclarecido: Apresentado e Adequado.
3. Instrumentos de coleta de dados (questionários, formulários, roteiros de entrevistas, etc): Apresentados e Adequados.
4. Folha de Rosto: Adequada. Assinada pela Coordenadora do PPGI.
5. Cronograma: Apresentado e Adequado.
6. Orçamento: Apresentado e Adequado.

Conclusões ou Pendências e Lista de Inadequações:

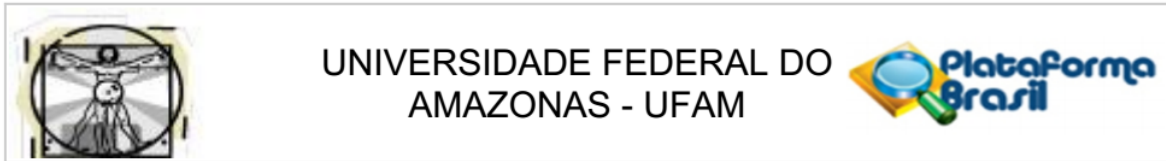
O Protocolo está de acordo com a Res. CNS 466/12 e satisfaz as exigências regulamentares do CEP. Nosso PARECER É PELA APROVAÇÃO DO PROJETO

Considerações Finais a critério do CEP:

Este parecer foi elaborado baseado nos documentos abaixo relacionados:

Tipo Documento	Arquivo	Postagem	Autor	Situação
Informações Básicas do Projeto	PB_INFORMAÇÕES_BÁSICAS_DO_PROJETO_1077929.pdf	19/02/2018 16:32:58		Aceito
Informações Básicas do Projeto	PB_INFORMAÇÕES_BÁSICAS_DO_PROJETO_1077929.pdf	19/02/2018 16:20:18		Aceito
TCLE / Termos de Assentimento / Justificativa de Ausência	TCLE_Projeto.pdf	19/02/2018 16:03:26	WILLIAMSON ALISON FREITAS SILVA	Aceito
Outros	Instrumentos_Coleta_Dados.pdf	19/02/2018 16:02:59	WILLIAMSON ALISON FREITAS SILVA	Aceito
Projeto Detalhado / Brochura Investigador	Projeto_Detalhado.pdf	19/02/2018 16:01:57	WILLIAMSON ALISON FREITAS SILVA	Aceito
Declaração de Instituição e Infraestrutura	Termo_De_anuencia_Preenchido.pdf	19/02/2018 16:01:20	WILLIAMSON ALISON FREITAS SILVA	Aceito

Endereço: Rua Teresina, 495
Bairro: Adrianópolis **CEP:** 69.057-070
UF: AM **Município:** MANAUS
Telefone: (92)3305-1181 **E-mail:** cep.ufam@gmail.com



Continuação do Parecer: 2.545.694

Folha de Rosto	Folha_De_rosto_Preenchida.pdf	19/02/2018 15:18:29	WILLIAMSON ALISON FREITAS SILVA	Aceito
----------------	-------------------------------	------------------------	---------------------------------------	--------

Situação do Parecer:

Aprovado

Necessita Apreciação da CONEP:

Não

MANAUS, 15 de Março de 2018

Assinado por:
Eliana Maria Pereira da Fonseca
(Coordenador)

Endereço: Rua Teresina, 495

Bairro: Adrianópolis

CEP: 69.057-070

UF: AM

Município: MANAUS

Telefone: (92)3305-1181

E-mail: cep.ufam@gmail.com

PROTOCOLO DO MSL DO MAPEAMENTO SISTEMÁTICO DA LITERATURA

1.1.Introdução

Para a condução do MSL, considerou-se as diretrizes fornecidas por Petersen *et al.* (2015) e Kitchenham e Charters (2007) para coletar as dificuldades enfrentadas pelos engenheiros de software (estudantes e profissionais de software) ao aprender modelagem de software com os diagramas UML. O MSL foi realizado em três etapas: planejamento, condução e relato de resultados. As atividades relacionadas as etapas de planejamento e condução do MSL são descritas nas subseções a seguir. Ao contrário das revisões informais da literatura, onde o pesquisador não segue um processo definido para sua condução, o MSL é executado de maneira formal obedecendo a um protocolo pré-estabelecido. Nas próximas seções, o protocolo para a condução do mapeamento sistemático é apresentado, baseado das diretrizes para condução de revisões sistemáticas proposto por Kitchenham e Charters (2007).

1.2. Questões De Pesquisa

Para este mapeamento, foi utilizada a seguinte Questão de Pesquisa (QP):

Questão de Pesquisa: Quais são as dificuldades identificadas na literatura que podem influenciar no ensino/aprendizagem de diagramas UML?

A partir da QP apresentada acima, buscou-se identificar três aspectos nas publicações selecionadas: quais as dificuldades que os engenheiros de software (estudantes ou profissionais) têm durante a aprendizagem de diagramas UML; quais tecnologias estão sendo usadas para apoiar o ensino e aprendizagem de diagramas UML; e, que tipo de estudos experimentais foram conduzidos usando diagramas UML ou as tecnologias propostas. Com base nesses três aspectos, foram definidas subquestões de pesquisa (SQ) para cada aspecto, a fim de responder a perguntas específicas, conforme pode ser visto na Tabela 1.

Tabela 1: Sub-Questões de Pesquisa.

Aspecto	Nº	Sub-questões de pesquisa
Dificuldades	SQ01	Quais são as dificuldades mencionadas no artigo?
Tecnologia	SQ02	Quais diagramas UML são abordados em cada artigo?
	SQ03	Há uma tecnologia ¹ que pode ser utilizada para apoiar no ensino/aprendizado dos diagramas?

¹ O termo “tecnologia” é utilizado como generalização para procedimentos, ferramentas, técnicas, metodologias e outras propostas realizadas na área de Engenharia de Software (Santos *et al.*, 2012).

	SQ04	A tecnologia é baseada em outra tecnologia existente?
	SQ05	Como a tecnologia pode ser utilizada (grupo ou individualmente)?
Experimentos	SQ06	O artigo apresenta uma avaliação experimental?
	SQ07	Qual o tipo de estudo experimental foi realizado?
	SQ08	Qual o perfil dos participantes do estudo?
	SQ09	Qual o tipo de resultado do estudo experimental?
	SQ10	O estudo experimental apresenta métrica(s) que visa(m) a avaliar os diagramas?

1.3. String de busca

O mecanismo de pesquisa disponível na maioria das bibliotecas digitais é baseado em expressões de pesquisa textuais. Assim, a definição da *string* de busca é essencial para a eficácia da etapa de busca e para o MSL como um todo (Steinmacher *et al.*, 2015). A *string* de busca usada na busca automática foi construída com base em três termos de pesquisa extraídos da questão geral de pesquisa (parte A da Figura 1): (1) dificuldades, (2) ensino e aprendizagem; e (3) diagramas UML. Após analisar os termos relacionados à questão de pesquisa, extraiu-se as palavras-chave e sinônimos dos artigos de controle (Tanaka *et al.*, 2015; Bera, 2012; Boberic-Krstićev e Tešendić, 2013; Kruus *et al.*, 2014; Stikkolorum *et al.*, 2015; Moisan e Rigault, 2009; Sien, 2011). O processo de construção da *string* de busca foi desenvolvido iterativamente para garantir que fosse usado um conjunto abrangente de sinônimos, permitindo uma alta cobertura de artigos. Além disso, foi utilizado um processo de refinamento semelhante ao realizado por Zhang *et al.* (2010) para incluir novos termos de pesquisa dos artigos selecionados anteriormente e verificar se os artigos de controle estavam retornando nas sequências de pesquisa sendo testadas. Após as iterações, os pesquisadores alcançaram um nível razoável de cobertura de artigos com a sequência de pesquisa nas bibliotecas digitais. A *string* de busca final que foi utilizada neste mapeamento é apresentada na parte B da Figura 1.

As bibliotecas digitais devem indexar os artigos sobre as dificuldades no ensino/aprendizagem em modelar diagramas UML escritos em inglês, apoiar a pesquisa usando expressões booleanas e fornecer acesso ao texto completo do artigo. Com base nesses critérios, quatro bibliotecas digitais foram selecionadas para conduzir a busca: *Scopus*², *Engineering Village*³, *Web of Science*⁴ e *ACM Digital Library*⁵.

² <http://www.scopus.com/> .

³ <https://www.engineeringvillage.com/home.url>.

⁴ <https://webofknowledge.com/>.

⁵ <http://dl.acm.org/>.

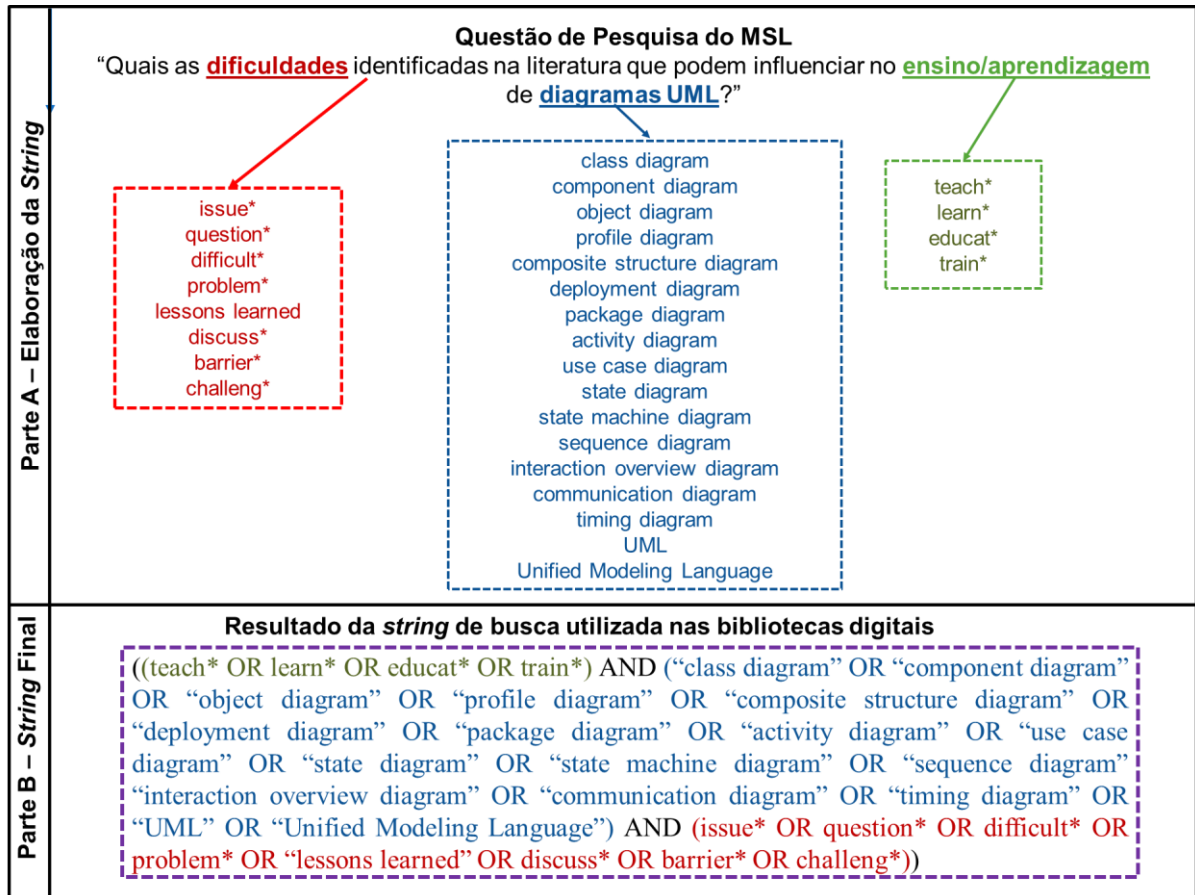


Figura 1. String de busca com base na questão de pesquisa do mapeamento.

1.4. Critérios para seleção dos artigos

Independentemente do mecanismo utilizado para realizar a busca, os artigos foram selecionados de acordo com vários critérios relevantes para a questão de pesquisa. Os critérios de inclusão e exclusão foram construídos com base nas diretrizes sugeridas por Kuhrmann *et al.* (2017). Os critérios para inclusão dos artigos são:

- **CI1.** Artigos discutindo dificuldades/lições aprendidas no ensino/aprendizagem de diagramas UML.
- **CI2.** Artigos que apresentam estudos experimentais que discutem o ensino/aprendizagem de diagramas UML.
- **CI3.** Artigos que discutam métricas que podem ser usadas para avaliar diagramas UML.
- **CI4.** Artigos que apresentam tecnologias que podem ser usadas para apoiar o ensino/aprendizagem de diagramas UML.
- **CI5.** Artigos que discutem aspectos relacionados ao ensino/aprendizagem de diagramas UML.

Os artigos que atenderam a pelo menos um dos seguintes critérios de exclusão foram excluídos:

- **CE1.** Artigos duplicados (por exemplo, um artigo com um estudo publicado em locais diferentes ou em datas diferentes). Nesse caso, será considerada apenas a versão mais completa e mais recente.
- **CE2.** Artigos que apresentam o uso de diagramas no desenvolvimento de software, mas não mencionam dificuldades / lições aprendidas sobre o uso desses diagramas.
- **CE3.** Artigos que não estão escritos em inglês.
- **CE4.** Os seguintes tipos de publicação: livros, teses de doutorado, dissertações de mestrado, patentes, tutoriais, propostas de oficinas ou pôsteres.
- **CE5.** O texto completo do artigo não está disponível para os autores (mesmo após contato com os autores).

1.5. Estratégia para seleção dos artigos

A *string* de busca apresentada na Subseção 1.3 foi utilizada para recuperar os artigos das bibliotecas digitais em abril de 2019. O processo de seleção foi iterativo e incremental, no qual cada artigo passou por duas etapas diferentes de filtro.

Na primeira etapa, os artigos resultantes foram filtrados com base em seu título e resumo. Quando os títulos e resumos dos artigos não estavam relacionados às dificuldades no ensino/aprendizagem de diagramas UML, eles eram excluídos. Para esse processo, aplicava-se os critérios de inclusão e exclusão descritos na subseção anterior. A decisão de incluir/excluir e classificar um artigo na primeira etapa foi conduzida e discutida por três pesquisadores. Para garantir a confiabilidade da inclusão de um trabalho candidato no MSL, os pesquisadores compararam seus resultados usando a medida Kappa (Landis e Koch, 1997). Esta medida de concordância tem como valor máximo o 1, onde este valor 1 representa total concordância e os valores próximos e até abaixo de 0, indicam nenhuma concordância, ou a concordância foi exatamente a esperada pelo acaso. Três avaliadores independentes classificaram uma amostra aleatória de 35 artigos. Cada pesquisador revisou os artigos individualmente e atribuiu um critério de seleção. Em seguida, os resultados dos pesquisadores foram comparados através do teste Kappa. O resultado foi um nível de concordância de 0,75, indicando uma concordância significativa entre os pesquisadores. Um resultado do Kappa acima de 0,70 mostra que houve

uma boa concordância entre os avaliadores. Além disso, isso tende a indicar que o processo não está ambíguo e foi usado de forma coerente pelos avaliadores.

Como as duas primeiras informações (título e resumo) não foram suficientes para identificar se o estudo é realmente relevante para a pesquisa realizada, a segunda etapa foi baseada na leitura do texto completo dos estudos que foram incluídos na primeira etapa. Assim, a segunda fase teve como objetivo garantir uma análise mais precisa dos estudos. Os estudos foram selecionados também de acordo com os critérios de inclusão e exclusão. Nesta etapa, os resultados foram verificados com outros dois pesquisadores e as divergências foram discutidas e resolvidas entre os pesquisadores.

A Tabela 2 resume o total de artigos encontrados e selecionados em cada fonte de dados.

Tabela 2. Artigos encontrados e incluídos por biblioteca digital.

Biblioteca digital	Artigos Encontrados	Artigos Relevantes
Scopus	881	108
Engineering Village	884	42
Web of Science	457	18
ACM Digital Library	581	13
Soma dos artigos	2.803	181
Artigos repetidos	1.331	0
Total	1.472	181

1.6. Definição da estratégia de extração de dados

Os trabalhos selecionados na segunda etapa foram submetidos ao processo de extração de dados. Para extrair os dados de forma sistemática, foi elaborado um formulário para registrar as informações necessárias para cada artigo, apoiando a resposta à pergunta da pesquisa. Com base nisso, adotou-se a estratégia de fornecer um conjunto de respostas possíveis durante a extração de papéis. Fernandez *et al.* (2011) afirmaram que essa estratégia garante a aplicação dos mesmos critérios de extração de dados em todos os artigos selecionados e facilita sua classificação. Neste sentido, o formulário de extração deste mapeamento sistemático foi definido conforme descrito na Tabela 3.

Tabela 3: Tabela para extração de dados

Aspecto	SubQuestões de Pesquisa	Possíveis Respostas
Informações gerais	Nome do Artigo	O título do artigo.
	Autor (es)	O nome dos autores do artigo.
	Tipo de publicação	O tipo de publicação do artigo (por exemplo, periódico)
	Ano	Ano em que o artigo/periódico foi publicado.

Aspecto	SubQuestões de Pesquisa	Possíveis Respostas
	Local de Publicação	O nome do local de publicação em que o artigo/periódico foi publicado.
Dificuldades	Dificuldade (s) identificada(s)/ discutida(s) no artigo	Frases qualitativas do artigo relacionadas a dificuldades no ensino / aprendizagem de diagramas UML.
	Diagramas(s) UML em que a(s) dificuldade(s) foi/foram percebida(s)	Diagrama de atividades, Diagrama de classes, Diagrama de colaboração, Diagrama de comunicação, Diagrama de componentes, Diagrama de implantação, Diagrama de implementação, Diagrama de objetos, Diagrama de pacote, Diagrama de sequência, Diagrama da máquina de estado, Diagramas UML (em geral), Descrição do caso de uso, Diagrama de casos de uso.
Tecnologia	O artigo apresenta uma tecnologia que auxilie na melhoria do ensino dos diagramas?	Sim ou Não.
	A tecnologia é baseada em outra existente?	Sim ou Não.
	A tecnologia foi proposta para ser utilizada em grupo, individualmente ou em pares?	Grupo, Individualmente, Pares, Combinado (Grupo e/ou Individualmente e/ou Pares) ou Não informado.
Estudo experimental	O artigo apresenta um estudo experimental?	Sim ou Não.
	Qual o método de pesquisa experimental realizado?	Informar o método de pesquisa utilizado: Estudo de caso, Relato de Experiência, Experimento Controlado, <i>Survey</i> (Pesquisa de Opinião), <i>Action Research</i> (pesquisa-ação), Métodos Mistos, Design Science, Estudo secundário (Revisão Sistemática da Literatura, Mapeamento Sistemático da Literatura/ Meta-Análise), Estudo Terciário (Revisão Terciária) ou não especificado.
	Qual o tipo de participante do estudo?	Não especificado, Estudantes, Profissionais, Estudantes e Profissionais.
	Qual o tipo de resultado do estudo realizado experimental?	Quantitativo, Qualitativo ou Ambos (Quanti e Quali).
	Foram utilizadas métricas para avaliar os diagramas?	Informar se foram utilizadas métricas para avaliar os diagramas? Caso houver métricas, especificar a métrica utilizada no artigo.

Por fim, coletou-se as informações fornecidas nos trabalhos relacionadas às dificuldades identificadas ao aprender/executar a modelagem usando os diagramas da UML. Assim, catalogou-se as dificuldades identificadas. Cada dificuldade estava relacionada às informações sobre o estudo, com o diagrama UML ao qual ele se relaciona e ao tipo de evidência usada para

indicá-lo. Após essa identificação, cada dificuldade foi vinculada aos segmentos de texto que a sustentavam nos estudos em que foram identificados. Utilizando os segmentos de texto, aplicando uma abordagem baseada em procedimentos de codificação classificou-se as dificuldades identificadas (Strauss e Corbin, 1998). Ao analisar os dados, criou-se códigos (conceitos relevantes para a compreensão das dificuldades) relacionados aos segmentos de texto (informações descritas nos artigos). Em seguida, os códigos foram analisados e agrupados de acordo com suas propriedades, formando conceitos que representam categorias e subcategorias. A análise foi realizada por um pesquisador e discutida com outros dois pesquisadores com experiência em análise qualitativa.

1.7. Fase de condução

181 artigos foram selecionados de um conjunto inicial de 1.472 artigos. Alguns trabalhos apareceram em mais de uma biblioteca digital. Nesse caso, consideramos o artigo apenas uma vez, de acordo com a ordem de pesquisa (primeiro *Scopus*, seguido por *Engineering Village*, *Web of Science* e *ACM Digital Library*). Todos os artigos em potencial foram acessados usando a conexão com a Internet e a autenticação fornecidas pela universidade do primeiro autor. Quando um artigo potencial não estava disponível gratuitamente na rede da instituição, o autor correspondente era contatado para solicitar a versão do autor.

DEFINIÇÃO DAS MÉTRICAS

A definição das métricas identificadas nos estudos empíricos abordados no mapeamento sistemático é apresentada abaixo. Vale ressaltar que todos os estudos que utilizam medidas que medem o mesmo conceito são agrupados (mesmo que tenham sido originalmente apresentados com nomes diferentes).

A. Corretude

Fórmula utilizada: $Q_{cr}(S_p) = |S_p \cap S_e| / |S_p|$, onde

S_e o conjunto de artefatos de um certo tipo produzido por um grupo de especialistas no assunto e S_p o conjunto de tais artefatos produzidos por um participante do experimento.

Artigos em que esta métrica é utilizada: S01.

B. Completude

Fórmula utilizada: $Q_{cp}(S_p) = |S_p \cap S_e| / |S_e|$, onde

S_e o conjunto de artefatos de um certo tipo produzido por um grupo de especialistas no assunto e S_p o conjunto de tais artefatos produzidos por um participante do experimento.

Artigos em que esta métrica é utilizada: S01.

C. Atributos de Qualidade

Esta é uma métrica subjetiva obtida como um ranking da compreensão percebida pelo profissional (professor) sobre um determinado diagrama. O questionário possuía seis atributos de qualidade, em que cada atributo era medido usando uma escala ordinal de Likert de -4 a 4 ou 1 a 8.

Artigos em que esta métrica é utilizada: S02.

D. F-Measure

Definição: é uma métrica agregada que é uma combinação padrão de *recall* e *precision*, definida como sua média harmônica.

Fórmula: $F\text{-Measure} = (2 * precision_{s/i} * recall_{s/i}) / (precision_{s/i} + recall_{s/i})$

○ *Recall*

Definição: O *recall* mede a fração do número de problemas corretos identificados com respeito ao número de problemas de modelagem no diagrama.

Fórmula: $= |A_{s/i} \cap C_i| / |C_i|$, onde

$A_{s/i}$ é o conjunto de elementos mencionados em uma resposta para uma questão i pelo participante s e C_i é o conjunto correto de elementos esperados para a questão i

○ *Precision*

Definição: O *precision* mede a fração do número de problemas corretos identificados com relação ao número de problemas identificados.

Fórmula: $= |A_{s/i} \cap C_i| / |A_{s/i}|$, onde

$A_{s/i}$ é o conjunto de elementos mencionados na resposta para a questão i pelo participante s e o C_i é o conjunto correto de elemtnos esperados para a questão i .

Artigos em que esta métrica é utilizada: S03 e S16.

E. RUBRIC

Definição: a métrica RUBRIC é um conjunto de critérios que avaliam a qualidade dos modelos que são elaborados. É necessário avaliar o diagrama e verificar qual critério o diagrama se encontra.

Artigos em que esta métrica é utilizada: S10.

F. Erros

Definição: esta é uma métrica que conta o número de erros cometidos na resolução de uma tarefa específica.

Artigos em que esta métrica é utilizada: S15, S32, S55, S69, S77.

G. Eficiência

Fórmula: tempo necessário para especificar informações (aspectos do processo) contidas em uma determinada descrição do processo (tempo / # aspectos do processo especificado)

Artigos em que esta métrica é utilizada: S27.

H. Eficácia

Fórmula: esta métrica é calculada a partir da quantidade de defeitos identificados nos diagramas. Alguns destes defeitos são: (a) quantidade de erros sintáticos, (b) quantidade de aspectos do processo modelado incorretamente e (c) quantidade de aspectos do processo que estão faltando.

Artigos em que esta métrica é utilizada: S27 e S73.

I. Compreensibilidade

Definição: esta métrica é utilizada para (a) calcular a quantidade de respostas incorretas informada pelos participantes para questões relacionadas ao conteúdo referente a um dado artefato e para (b) calcular a quantidade de respostas dadas pelos participantes ao encontrar erros em artefatos especificados erroneamente.

Artigos em que esta métrica é utilizada: S27.

J. Adequação

Definição: esta métrica utiliza um conjunto de critérios que avalia a competência dos participantes na construção dos diagramas.

Artigos em que esta métrica é utilizada: S31.

Artigos que usam esta medida com nomes diferentes:

- *Eficácia Pedagógica: S37*
- *Eficácia cognitiva: S46, S47*

DIFICULDADES IDENTIFICADAS A PARTIR DOS ESTUDOS DO MSL

A seguir será apresentada as dificuldades relacionadas aos modelos retiradas dos estudos identificados a partir do mapeamento sistemático da literatura.

DIFICULDADES RELACIONADAS AO DIAGRAMA DE ATIVIDADES

Categoria	Subcategoria	Citações
Dificuldades relacionadas ao diagrama de atividades	Dificuldade em compreender / utilizar o diagrama	(S68 – Dobing, B, e Parsons, J.; 2006): “A lack of understanding by analysts is the primary factor among the few not using Class Diagrams (50%). Similar concerns were expressed by 48% of respondents about Activity Diagrams.” (S71 – Siau, K, e Loo, P-L.; 2006): “Students are also confused about when to use the sequence and when to use the collaboration/ communication diagrams. Similar issues exist for the activity and the statechart/state diagrams. These issues need to be addressed by the instructors and hopefully alleviated in future versions of UML.”
	Dificuldade em compreender que cada cenário está relacionado a um fluxo do diagrama.	(S24 – Boberic-Krsticev, D., and Tesendic, D.; 2013): “some students didn’t understand that those scenarios relate to one use case, so they create separate diagrams for each scenario.
	Dificuldade em definir o nível de granularidade	(S40 – Moisan, S., e Rigault, J-P; 2008): “it presents a serious danger, identical to one of the previously mentioned use case risk: some attendees may continue the decomposition up to very ne grained activities, ending up with a functional approach and thus, here again, missing the point of object-oriented modeling.”

DIFICULDADES RELACIONADAS AO DIAGRAMA DE COLABORAÇÃO

Categoria	Subcategoria	Citações
Dificuldades relacionadas diagrama de colaboração	Dificuldade em compreender/ utilizar o diagrama	(S68 – Dobing, B e Parsons, J.; 2006): “Respondents in our survey clearly prefer Sequence Diagrams to Collaboration Diagrams. The latter are used less frequently (Figure 1), are less useful for various purposes (Table 1), and are more likely to be seen as redundant (Table 2) when compared to the former.” (S71 – Siau, K, e Loo, P-L.; 2006): “Students are also confused about when to use the sequence and when to use the collaboration / communication diagrams. Similar issues exist for the activity and the statechart / state diagrams. These issues need to be addressed by the instructors and hopefully alleviated in future versions of UML.”
	Dificuldade em avaliar se a informação é relevante	(S68 – Dobing, B e Parsons, J.; 2006): “Similar concerns were expressed by 48% of respondents about Activity Diagrams. Leading concerns for the remaining components are about usefulness (Statechart), value (Sequence and Use Case Diagrams and Narratives), and redundancy (Collaboration)”

DIFICULDADES RELACIONADAS AO DIAGRAMA DE COMUNICAÇÃO

Categoria	Subcategoria	Citações
Dificuldades relacionadas ao diagrama de comunicação	Dificuldade em compreender/ utilizar o diagrama	(S71 – Siau, K, e Loo, P-L.; 2006): “Students are also confused about when to use the sequence and when to use the collaboration / communication diagrams. Similar issues exist for the activity and the statechart/state diagrams. These issues need to be addressed by the instructors and hopefully alleviated in future versions of UML.”

DIFICULDADES RELACIONADAS AO DIAGRAMA DE ESTADOS

Categoria	Subcategoria	Citações
Dificuldades relacionadas ao diagrama de estados	Dificuldade em compreender/ utilizar o diagrama	(S68 – Dobing, B e Parsons, J.; 2006): “Statechart Diagrams seem to be less useful most of the time (Table 2) but are rated highly for providing new information in some situations (Figure 1) and have low redundancy (Table 2). As one interview subject put it, “When they are useful, they are very useful.” (S68 – Dobing, B e Parsons, J.; 2006): “Statechart Diagrams are very useful for their intended purpose but are not critical for many systems.” (S71 – Siau, K, e Loo, P-L.; 2006): “Students are also confused about when to use the sequence and when to use the collaboration / communication diagrams. Similar issues exist for the activity and the statechart/state diagrams. These issues need to be addressed by the instructors and hopefully alleviated in future versions of UML.” (S68 – Dobing, B e Parsons, J.; 2006): “Leading concerns for the remaining components are about usefulness (Statechart), value (Sequence and Use Case Diagrams and Narratives), and redundancy (Collaboration).”

DIFICULDADES RELACIONADAS AO DIAGRAMA DE COMPONENTES

Categoria	Subcategoria	Citações
Dificuldades relacionadas ao	Dificuldade em perceber os blocos funcionais do sistema	(S24 – Boberic-Krsticev, D., and Tesendic, D.; 2013): “Students of both programmers regarded component diagram as the most difficult diagram and we think that this is

Categoria	Subcategoria	Citações
diagrama de componentes		particularly because they were not able to perceive different functional blocks of the system.”

DIFICULDADES RELACIONADAS AO DIAGRAMA DE OBJETOS

Categoria	Subcategoria	Citações
Dificuldade relacionadas ao diagrama de objetos	Dificuldade em identificar os objetos	(S40 – Moisan, S., e Rigault, J-P; 2008): “identifying objects and classes at the correct abstract level remains a difficult issue, even for people used to data base modeling. Indeed, if there can be "ricks", there are no absolute "recipes" to do it.”

DIFICULDADES RELACIONADAS A ESPECIFICAÇÃO DE CASOS DE USO

Categoria	Subcategoria	Citações
Dificuldades relacionadas ao diagrama de estados	Dificuldade em verificar se o seu custo de especificar o caso de uso justifica o seu uso	(S68 – Dobing, B e Parsons, J.; 2006): “Similar concerns were expressed by 48% of respondents about Activity Diagrams. Leading concerns for the remaining components are about usefulness (Statechart), value (Sequence and Use Case Diagrams and Narratives), and redundancy (Collaboration)”
	Dificuldade em definir o detalhamento da descrição	(S67 - Bolloju, N., 2006): “Excessive use of low level steps in use case descriptions is possibly due to the tendency of team members in beefing up the use case description without attempting to acquire or gather sufficient domain knowledge pertaining to the underlying use case scenario. It appears that the novice analysts have tried to compensate for their lack of domain expertise by introducing unnecessary steps or splitting simple steps.”
	Dificuldade em descrever os fluxos do caso de uso	(S40 – Moisan, S., e Rigault, J-P; 2008): “Some people have problems in writing. Other are (too) proficient in this domain and can hardly respect the restricted style required by scenarios”

DIFICULDADES RELACIONADAS AO DIAGRAMA DE CASOS DE USO

Categoria	Subcategoria	Citações
Dificuldades relacionadas ao diagram de casos de uso	Dificuldade para verificar se o diagrama justifica o custo de sua construção	(S68 – Dobing, B e Parsons, J.; 2006): “Similar concerns were expressed by 48% of respondents about Activity Diagrams. Leading concerns for the remaining components are about usefulness (Statechart), value (Sequence and Use Case Diagrams and Narratives), and redundancy (Collaboration)” (S24 – Boberic-Krsticev, D., and Tesendic, D.; 2013): “More than half of students of mathematics said that use case diagrams were moderately difficult.”
	Dificuldade em evitar ambiguidade	(S79 – Hallinan, S. and Paul Gibson, J. 2005): “The challenge in using actors is knowing where and when to use them, and to use them in such a way that does not result in ambiguity. Through investigation it was found that the system delivers value “to an actor” and an actor “represents a role that a human, a hardware device, or even another system plays with a system”. This may result in confusion in the Use Case diagram. For example, the image of an actor is that of a stick-man (which may be associated directly with people and hence users); and secondly, if there are also actors representing other systems then the exact context in which that actor is interacting with another system needs to be correctly understood. This type of contextual understanding is difficult to learn from traditional lectures, and comes only through experience of actually doing design as part of the problem solving process.” (S79 – Hallinan, S. and Paul Gibson, J. 2005): “Wherever ambiguity is encountered, a challenge is presented in terms of how to resolve this ambiguity. In the case of Use Case diagrams, it was found to be how to correctly encapsulate system behaviour, so that the minimum of system activity is displayed, and the maximum of system functionality is delivered to the user. This also must be performed in such a manner that neither too much nor too little functionality is attributed per Use Case, as it may result in an unbalanced view of the value delivered.”
	Dificuldade de identificar as relações entre UC	(S24 – Boberic-Krsticev, D., and Tesendic, D.; 2013): “it is noticed that there was problem relating to identification of relationships between use cases. During the use case modelling it is possible to identify include and extend relationships and in some situations, students couldn’t decide whether to use include or extend relationship.”
	Dificuldade em definir o nível adequado de granularidade do UC	(S24 – Boberic-Krsticev, D., and Tesendic, D.; 2013): “There was a problem with identifying use cases and determining the right level of use case granularity. Namely, students tended to derive smaller, finer grained use cases which represent atomic functionalities. This approach results in losing idea of main system functionalities and relations among them. Use case diagrams became overwhelmed and hard to follow. Students who created small use cases more often used include relationship. On the other side, some students had tendency to create large, more abstract use cases, which leads to ambiguous use case boundaries and problems in understanding of system. In addition, the next problem relates to students’ tendency to equal use cases with specification of user interface. In other words, a number of students regarded identification of use cases as defining screens of the user interface and for every button on the screen they defined new use case, which leads to very fine grained use case model” (S79 – Hallinan, S. and Paul Gibson, J. 2005): “One of the basic challenges presented to the beginner with the move into the detailed requirements stage is the manner in which the increasing number and type of diagram should be managed. In the case of the project the challenge was that there were multiple end-user types with different functionality

Categoria	Subcategoria	Citações
		associated with each. It was decided that it would be of value to the developer if the layout and flow of detail in the UML Use Case diagrams mapped well to the Functional Requirements specification.” (S40 – Moisan, S., e Rigault, J-P; 2008): “The second risk with use cases is to go too far in the functional decomposition, entering into low level system details, forgetting about the external actors, and missing the point of object-orientation.”
	Dificuldade em identificar os casos de uso	(S24 – Boberic-Krsticev, D., and Tesendic, D.; 2013): “There was a problem with identifying use cases and determining the right level of use case granularity.” (S79 – Hallinan, S. and Paul Gibson, J. 2005): “The first challenge encountered with the use of Use Case diagrams was correct application so that there was a minimum of unnecessary functionality or elaboration that was non-value-added and hindered the objective that they were intended to assist. In this regard it took a number of iterations of diagrams before the core set of functionality to be delivered to each user type was agreed upon. The early iterations of diagrams contained too much information: • unnecessary functionality in terms of what was being delivered to the user, • complexity in terms of inter-relationships of actors and flow of control with regard to the user, and; • system functionality that was not involved in delivering value to the user.”
	Dificuldade em compreender/ utilizar o diagrama	(S71 – Siau, K, e Loo, P-L.; 2006): “(...) the use-case diagram is probably the most controversial diagramming technique in UML. The role and uses of the use-case diagram in UML need to be clarified and explained.”

DIFICULDADES RELACIONADAS AO DIAGRAMA DE SEQUÊNCIA

Categoria	Subcategoria	Citações
	Dificuldade em verificar se o diagrama justifica o custo de sua construção	(S68 – Dobing, B e Parsons, J.; 2006): “Similar concerns were expressed by 48% of respondents about Activity Diagrams. Leading concerns for the remaining components are about usefulness (Statechart), value (Sequence and Use Case Diagrams and Narratives), and redundancy (Collaboration).”
	Dificuldade em entender a sintaxe	(S40 – Moisan, S., e Rigault, J-P; 2008): “However, the syntax becomes complex and may require programming experience. It is sometimes so complex that a programming language (or pseudo-code) representation may be clearer and more natural. Moreover, many tools do not handle nicely fragment edition or update. Another problem about sequence diagrams is that some attendees wonder why using these diagrams since few tools generate code, simulation, or test from them.”
	Dificuldade em entender os conceitos	(S24 – Boberic-Krsticev, D., and Tesendic, D.; 2013): “Acquiring concepts of sequence diagram was difficult for 50% of students of computer science, while on the other hand almost all students of mathematics said that sequence diagrams are difficult or even very difficult.”
	Dificuldade em desenhar / rascunhar o diagrama	(S31 – Sien, V. Y.; 2011): “Twenty-one students from Group 1 did not produce any sequence diagrams – they claimed that they did not know how to draw them. However, two of their IT lecturers who were present at the session said that sequence diagrams have been covered in their OOAD course. Only one student from Group 2 did not produce a sequence diagram. His/her reason for not drawing a sequence diagram was not made known to the facilitator. Our analysis of sequence diagrams is therefore based on the 20 models produced by Group 1 and 10 models produced by Group 2.”
Objetos	Dificuldade em empregar os objetos de controle	(S31 – Sien, V. Y.; 2011): “Only 40% of the sequence diagrams included an appropriate controller object (representing the facade controller) in their sequence diagrams.” (S31 – Sien, V. Y.; 2011): “Only 40% of the sequence diagrams defined an appropriate controller object for delegating tasks to other objects that play a part in the use case scenario.”
	Dificuldades em utilizar os objetos identificados a partir do diagrama de classes	(S31 – Sien, V. Y.; 2011): “As shown in Table 7, 70% of the students were able to produce at least one or two appropriate objects, e.g. Project, Staff and Client. A significant proportion (85%) of sequence diagrams had objects that did not relate to corresponding classes identified in the class diagram. Some examples of these inappropriate objects are CreateProjectSchedule, AssignStaff, and NewStaff. This difficulty provides evidence that some students do not understand that they should use the objects that have been identified from the problem domain (defined in the class diagram) in their sequence diagram.” (S31 – Sien, V. Y.; 2011): “There were 85% of sequence diagrams that contained objects that did not relate to corresponding classes defined in the class diagram.” (S31 – Sien, V. Y.; 2011): “Fulfilling the responsibilities of use cases by getting objects to pass messages to each other. The participants were unaware that the sequence diagrams are dependent on the analysis class diagram in terms of its classes, associations and multiplicities.” (S62 – Sin, T; 2007): “Incorrect and missing entity classes even when going through data modeling”
	Dificuldade em modelar os multi-objetos	(S31 – Sien, V. Y.; 2011): “A multi-object/container represents a set of instances and corresponds to a class defined in the class diagram with a multiplicity of ‘*’. We expect three multi-objects to be defined in the sequence diagram for Use Case 2. However, only 10% of diagrams have at least one multi-object defined correctly”
Mensagens	Dificuldade em definir as mensagens/parâmetros	(S31 – Sien, V. Y.; 2011): “Only 35% of the sequence diagrams produced included parameters in the messages – we do not however consider this a serious design fault as parameters can be optionally defined in UML. However, these sequence diagrams had only one or two appropriate parameters defined in their respective messages. The more common

Categoria	Subcategoria	Citações
		parameters that have been appropriately identified are found in the message for creating a ProjectSchedule object, addSchedule (estDate, estTime, estCost)." (S31 – Sien, V. Y.; 2011): “As shown in Table 8, only 48% of sequence diagrams displayed some appropriate evidence of responsibilities delegated to the appropriate objects. None of the student diagrams fulfilled all the responsibilities of the use case. (S31 – Sien, V. Y.; 2011): “There were 48% of sequence diagrams produced that displayed some appropriate evidence of responsibilities delegated to the appropriate objects. None of the sequence diagrams fulfilled all the responsibilities of the use case.”
Fluxos	Dificuldades para expressar os fluxos de controle	(S40 – Moisan, S., e Rigault, J-P; 2008): “For traditional sequence diagrams, they are simple to use but their power of expression is limited, especially to express complex control flows. Thus many diagrams are needed to express the scenarios and their variants, often with common and thus repeated parts.”
Passos do Diagrama de Sequência	Identificar todos os passos para o diagrama de sequência	(S62 – Sin, T; 2007): “Missing certain vital steps, most importantly, data modeling and going straight to modeling the sequence diagram” (S62 – Sin, T; 2007): “A sense of vagueness about how the pieces connect to one another. There is a sense of vagueness of subgoals. Designers don’t know how to move systematically from one activity to another” (S62 – Sin, T; 2007): “Not following each activity in the use case to completion.” (Sin, 2007) (S62 – Sin, T; 2007): “Lack of an overall strategy and inability to break down the procedure into discrete steps. There are too many choices, which seem to overwhelm the designers. The breadth of the task overwhelms designers” (S62 – Sin, T; 2007): “Inability to distinguish the “transaction” in the task from the transaction in the training example which resulted in borrowing and “copying” objects and logic from the example”

DIFICULDADES RELACIONADAS AO DIAGRAMA DE CLASSES

Categoria	Subcategoria	Citações
	Dificuldade em compreender os conceitos do diagrama	(S24 – Boberic-Krsticev, D., and Tesendic, D.; 2013): “Mastering concepts of class diagram was moderately difficult for students of computer science programme, while students of mathematics had more difficulties to understand complex concepts of class diagrams.” (S24 – Boberic-Krsticev, D., and Tesendic, D.; 2013): “In addition, students who have previous knowledge of database modelling had difficulties in accepting concept of object-oriented model. They avoided using many-to-many associations, aggregation, composition, association classes as well as interfaces and abstract classes. Namely, they considered entity classes as a database tables instead of regarding them as layer which communicates with database. Also, as a result of that they didn’t conceive the value of using operations in the entity classes. Generally speaking, it is observed that students have a little or no experience in object-oriented programming and because of that they cannot comprehend how some particular model will be implemented. It was very hard to explain to students that, for example, many-to-many association between two classes should be implemented in such way that one class has an attribute which is a collection of objects of the second class and otherwise.” (S68 – Dobing, B e Parsons, J.; 2006): “A lack of understanding by analysts is the primary factor among the few not using Class Diagrams (50%).” (S31 – Sien, V. Y.; 2011): “Identification of appropriate concepts and relationships within the problem domain. Only 6% of the participants were able to produce class diagrams with all of the expected classes. There were 36% of class diagrams that included classes with inappropriate class names. The participants failed to include essential attributes in the identified classes – 58% of the class diagrams produced had missing attributes. There were 59% of class diagrams with inappropriate associations and all the class diagrams had missing associations. Only 52% of the participants attempted to include generalisation–specialization hierarchies in their diagrams. Among these attempts, only 59% were defined appropriately. A significant number of participants (94%) did not attempt to identify whole–part associations in their diagrams.”
	Dificuldade em compreender a notação do diagrama	(S30 – Boustedt, J.; 2012): “S17 continues to describe different forms of arrows, but cannot tell what they mean: ‘... and then there are of course different kinds of arrows, filled and non filled, dotted or, what is it called, not dotted, straight lines, and you can also see, if you think that this is classes or modules or anything, you can see which ones are most important or most central. G here, for example, is in fact, seems in fact to be in the middle of the chart, and also has many arrows towards it, and above all here, well, it is difficult to say when you do not know what the arrows mean or what it is all about...[S17]”
Classes	Dificuldade para identificar as classes	(S31 – Sien, V. Y.; 2011): “Among the total group of 52 students, only 6% managed to identify all eight expected candidate classes, 31% managed to identify seven expected candidate classes and 4% were able to identify only one suitable class in their class diagrams. Some examples of missing classes are illustrated in Figure 4. They represent essential information about the system that has been omitted from the class diagrams e.g. the ProjectSchedule class contains information relating to the schedule of each project (e.g. estimated date started, estimated date ended, actual date started, actual date ended, etc.) and the Task class contains information of each specific task related to the project (e.g. description, date started, estimated cost, etc).”

Categoria	Subcategoria	Citações
		(S40 – Moisan, S., e Rigault, J-P; 2008): “identifying objects and classes at the correct abstract level remains a difficult issue, even for people used to data base modeling. Indeed, if there can be “ricks”, there are no absolute “recipes” to do it.”
	Dificuldade para entender / identificar as classes de controle	(S31 – Sien, V. Y.; 2011): “Before the exercise, Group 1 expressed that they had no knowledge on Controller classes during the revision session. Consequently, the facilitator provided them with some further explanations on the role of Controller classes. However, 66% of the students in this group did not include a Controller class in their class diagrams even though they had been advised to include a class to represent the overall system.” (Sien, 2011) (S24 – Boberic-Krsticev, D., and Tesendic, D.; 2013): “Students usually have some previous knowledge of two tier architecture and by creating analyses class model they are introducing with three tier architectures. Adopting of that new concept was difficult for them. Mainly they only identified entity and boundary classes while failed to observe control classes. So, operations which should belong to control classes were assigned to boundary classes. Result of this is that students created methods of boundary classes which directly call methods of entity classes.” (S31 – Sien, V. Y.; 2011): “Controller class. Before the exercise, Group 1 expressed that they had no knowledge on Controller classes during the revision session. Consequently, the facilitator provided them with some further explanations on the role of Controller classes. However, 66% of the students in this group did not include a Controller class in their class diagrams even though they had been advised to include a class to represent the overall system.”
	Dificuldade em entender as classes de fronteira	(S24 – Boberic-Krsticev, D., and Tesendic, D.; 2013): “there were some cases where students ignored boundary classes and created only control and entity classes. Reason for this lays in the fact that they didn’t understand role of boundary classes. We are dealing with this problem by drawing sketches and mock-ups of user interface and based on that we create boundary classes and make connections between user interface actions and boundary classes’ methods.”
	Dificuldade em descrever o nome das classes da forma adequada	(S31 – Sien, V. Y.; 2011): “There are 15 (36%) class diagrams with class names that do not represent a real-world concept. For example, one student produced some classes as shown in Figure 5 containing use case names as class names.”
Atributos	Dificuldade para alocar os atributos nas classes corretas	(S31 – Sien, V. Y.; 2011): “Figure 6 illustrates examples of attributes produced by students e.g. personName, taskStartDate and taskEndDate defined in a Project class that should be defined in the respective Person and Task classes.”
	Dificuldade para definir os atributos das classes	(S31 – Sien, V. Y.; 2011): “Students commonly fail to include essential attributes in classes. For example, in the CreateProjectSchedule expanded use case, the information for the schedule has been defined as estimated date started, estimated date ended, estimated time, estimated cost, etc. However, some students produced a ProjectSchedule class without some of these attributes.”
Relacionamentos	Dificuldade em definir as associações	(S31 – Sien, V. Y.; 2011): “The results in Table 4 show that identification of associations is a serious problem for the participants; 59% of students provided inappropriate associations and all the class diagrams had at least one missing association. None of the students provided the reflexive association required for the Task class even though the requirements stated that current tasks may depend on tasks that have been completed earlier. Another common fault was the omission of the association between the Partner class and the Project class, where information is required for a particular partner object managing its project objects (see Figure 7).”
	Dificuldade em utilizar especialização e generalização	(S31 – Sien, V. Y.; 2011): “We found that the class diagrams produced by the students in our study did not include many generalization–specialization hierarchies – only 52% of students tried incorporating a generalization–specialization hierarchy in their diagrams. Among these 27 attempts, 16 class diagrams had the hierarchy defined appropriately, 3 class diagrams had a mixture of appropriate and inappropriate inheritance hierarchies, and all the hierarchies defined in the other 8 diagrams were inappropriate. Detailed analysis on the appropriate and inappropriate use of generalization–specialization hierarchies are shown in Table 5. An example of an inappropriately defined generalization–specialization hierarchy is illustrated in Figure 9 where the ProjectSchedule and Task classes are defined as subclasses of the Project class. A more appropriate relationship to present the relationships between these three classes is a whole–part association.” (S57 -Alonso, M, Py, M., Lemeunier, T.; 2008): “Relationships’ orientation errors are especially interesting because we noticed that novices often have difficulties in understanding the meaning of relationships’ orientation.”
	Dificuldade no uso de agregação e composição	(S31 – Sien, V. Y.; 2011): “only one student managed to provide both whole–part associations correctly. A high percentage (94%) of students did not attempt to identify this type of association in their class diagrams (see Table 6).” (S30 – Boustedt, J.; 2012): “Only a minority of the informants could tell the difference between the white and black diamond and many informants could not explain what the diamonds mean when they did not have help from descriptive class names. The diamond symbol (and the term aggregation) was often (vaguely) described as a “has”-relationship, but it was rarer that informants described it in terms of a “whole/part”-relation. The “has”- relation was sometimes explained by saying that a class has an instance variable that can refer to another class. This is true, but is also true for other types of relations than aggregation – for example, an ordinary association”

DIFICULDADES RELACIONADAS A UML

Categoria	Citações
<p>Dificuldade em entender a notação da UML</p>	<p>(S24 – Boberic-Krsticev, D., and Tesendic, D.; 2013): “As it can be seen, the most of students of computer science had no problems with UML notation relating to use case diagrams”</p> <p>(S26 – Fernández-Sáez, A. M., Chaudron, M. R. V., Genero, M.; 2013): “Some of the disadvantages mentioned (like “No semantics”, “Unclear syntactics”, “Difficulties in understanding the notation”) might be caused by a poor understanding of UML diagrams.”</p> <p>(S09 – Petre, M; 2014): “The overheads of understanding the notation: UML is considered “unnecessarily complex.” Issues of comprehension included both software developers and stakeholders. Several informants reported variations in understanding and interpretation among developers that led to problems. Others noted that the complexities of the notation limited its utility—or demanded targeted use—in discussions with stakeholders (including highly technical stakeholders).”</p> <p>(S09 – Petre, M; 2014): “Most of the discussion refers to the complexity of the notation, e.g.: “UML has failed as a specification tool. The details of the notation are complex (do you remember what all the different arrow tips in a UML class diagram mean exactly?)” (d0k, Reddit discussion [18]). In particular, users are frustrated when the apparent accessibility of UML dissipates outside the context of the originating discussions, when different users have different “styles,” or when diagrams become intractably large, e.g.: “You see the boxes and arrows and it all seems so approachable! Then you ask three people who were not in the room when the diagrams were created what those boxes and arrows represent on more than a trivial level (what inputs/outputs for a system are they describing? what business rules are being described here? how would you make this happen?) and you discover that you’ve just spent half a week churning out the world’s most boring Rorschach test.” (48 klocs, Reddit discussion [18]). The increase in complexity with size affects the cost-benefit balance, e.g.: “I am at that tipping point myself. The use cases, class diagrams, and sequence diagrams have been instrumental on deciding what needs to be done. But management pushes for complete traceability from requirements to tests, and the overhead really starts to bog the entire thing down.” (cdcformatc,Reddit discussion [18]). However, complexity can trade-off with selective use, with many using UML selectively in order to exploit particular advantages while managing complexity. Tools may help, but they carry their own overheads and complexities: “I prefer tools to generate UML for my documentation... Making them by hand with the tools available suck. Either they’re so simplistic as to be useless (“What does this class do again?” or “How does this class do that thing?”, the context doesn’t tell us.), or you get so involved in trying to fill out the verbose details you get frustrated with the time you’re wasting just fucking with the tools. Further, some of the class diagrams tend to blur together and I can’t tell which diagram represents what; usually my trip up is between hierarchy and dependency.” (mredding, Wilson blog discussion [27]).”</p> <p>(S40 – Moisan, S., e Rigault, J-P; 2008): “UML is a big language, even bigger with UML 2. Moreover, the problem is not the language itself, but the concepts of modeling. The arcane notations of UML 2 are not suitable for non-developers and for this population we usually restrict to classical (UML 1) class, sequence, and activity diagrams.”</p> <p>(S71 – Siau, K, e Loo, P-L.; 2006): “In addition, subjects pointed out that a number of concepts could not be captured using UML notations. They also claimed that sometimes they could not express what they were trying to model. These difficulties can result from the inadequacy of UML itself. The findings for Region #3 thus show that these weaknesses are the source of at least some of the difficulties in learning UML.”</p> <p>(S71 – Siau, K, e Loo, P-L.; 2006): “Subjects had problems memorizing the notations and constructs in UML. The difficulties mentioned are included in Table 3. The large number of constructs in UML increases the difficulty in learning, because humans are constrained by the size of their short-term memory (Miller, 1956). To make learning UML easier, we need to reduce the number of constructs, resolve inconsistencies among constructs and diagrams, remove ambiguities in UML, and increase hands-on practice during the learning process”</p> <p>(S71 – Siau, K, e Loo, P-L.; 2006): “The semantics of some UML constructs and concepts are poorly defined. This creates learning difficulties for students — especially inexperienced and novice users of UML. For example, the definitions of stereotypes <<Include>> and <<Extend>> in Booch <i>et al.</i> (2005, p. 234) are beyond the comprehension of novice UML users. UML constructs and concepts should be clearly defined and discussed in an easily understandable way (rather than using formal technical jargon) to avoid ambiguities and misunderstanding.”</p>
<p>Dificuldade para compreender as relações entre diagramas da UML</p>	<p>(S24 – Boberic-Krsticev, D., and Tesendic, D.; 2013): “Sometimes, students do not even seem to know that the different kinds of UML diagrams, such as sequence and class diagrams, are related to one another although each of them addresses a particular aspect of the same system. The main problem lays in the fact that students only learn notations of particular diagrams and see diagrams isolated from each other.”</p> <p>(S24 – Boberic-Krsticev, D., and Tesendic, D.; 2013): “Students from both programmes had problems with the adoption of sequence diagrams. Consequence of low comprehension of class diagrams resulted in problems with adoption of sequence diagrams. Sequence diagrams may be the best indicators of the level of acceptance of class diagrams and three tier architecture. It is noticed that even though students created correct class diagrams and it seemed that they understood the concepts, they still created bad solution of related sequence diagrams. Sometimes, students do not even seem to know that the different kinds of UML diagrams, such as sequence and class diagrams, are related to one another although each of them addresses a particular aspect of the same system. The main problem lays in the fact that students only learn notations of particular diagrams and see diagrams isolated from each other.”</p> <p>(S31 – Sien, V. Y.; 2011): “Understanding the relationships between the various structural and behavioural UML diagrams – within the context of our study, they are unable to adequately connect</p>

Categoria	Citações
	<p>class and sequence diagrams. There were 85% of sequence diagrams that contained objects that did not relate to corresponding classes defined in the class diagram.”</p> <p>(S40 – Moisan, S., e Rigault, J-P; 2008): “UML proposes a wealth of different views on the system, at different levels of abstraction, corresponding to different phases of development. This is good! But these views are not independent, they have strong and consistent relationships. Most of these relations are ignored by tools and cannot be introduced by the modeler. As a matter of example, virtually all tools handle correctly a relation between sequence diagrams and class definitions. When drawing a message in the sequence diagram, one can conjure up a menu with the list of all operations of the receiving object. But when drawing a state machine representing the "protocol" of a particular class, where most of the transitions are labeled with one of the class operations, most tools, if any, do not show the same magic menu.”</p> <p>(S71 – Siau, K, e Loo, P-L.; 2006): “The current textbooks and training materials on UML do a poor job discussing the “linkages” between various diagrams. The various UML diagrams are used to present different aspects of a problem. However, students find it difficult to integrate and synthesize the information depicted in different diagrams. They also find it difficult to see the connections between the various UML diagrams. For example, how are the use-case and class diagrams linked to the interaction diagrams. The linkages between various diagrams need to be explicitly discussed in the textbooks, training materials, and classrooms.”</p>
Dificuldade na abstração dos diagramas	<p>(S26 – Fernández-Sáez, A. M., Chaudron, M. R. V., Genero, M.; 2013): “Note that “high level of abstraction” is mentioned as an advantage and a disadvantage at the same time. This may be because architects feel abstraction is beneficial, but developers need diagrams which are closer to the source code.”</p>
Dificuldade em entender a semântica da UML	<p>(S71 – Siau, K, e Loo, P-L.; 2006): “The pattern displayed in Region 4 indicates that subjects were unsure about some of the semantics of UML. Table 2 details the typical statements for this region and the interpretations. These difficulties are mainly due to those UML semantics that are not precisely defined. This is an inherent problem of the unifying process, which produces UML from many different object-oriented modeling techniques (Selic <i>et al.</i>, 2002). Furthermore, the ambiguity of UML semantics poses problems for novices and for forward engineering (i.e., generating codes from models).”</p> <p>(S26 – Fernández-Sáez, A. M., Chaudron, M. R. V., Genero, M.; 2013): “Some of the disadvantages mentioned (like “No semantics”, “Unclear syntactics”, “Difficulties in understanding the notation”) might be caused by a poor understanding of UML diagrams.”</p>
Dificuldade devido à falta de contexto	<p>(S09 – Petre, M; 2014): “Informants remarked that UML deals primarily with software architecture rather than the “whole” system, and hence that it lacks context.”</p>
Dificuldade em questões de sincronização e consistência	<p>(S09 – Petre, M; 2014): “A number of informants identified issues of synchronization or consistency, both in terms of transformation and coordination between different views/diagrams, and in terms of maintaining the coordination between the model and the implementation.”</p> <p>(S31 – Sien, V. Y.; 2011): “Inconsistent and confusing diagrams and constructs are obstacles in learning UML. (...) the inconsistency and confusing diagrams in UML impact the effectiveness of compiling declarative knowledge (e.g., definitions, meanings of diagrams, and other constructs of UML) into procedural knowledge (e.g., how to draw UML diagrams to effectively model an aspect of a system).”</p>
Dificuldade devido à necessidade de um conhecimento anterior	<p>(S71 – Siau, K, e Loo, P-L.; 2006): “According to the information processing theory, when learners acquire new information, they attempt to relate this new piece of information to their existing knowledge structure. Thus, when subjects are introduced to the concepts of UML, the foremost action is to search and recall similar concepts, knowledge, or production rules from their knowledge pool (long-term memory) and then try to create a new production rule that combines the new concepts learned with the old knowledge. If learners have very little or no experience in object orientation or UML, they can encounter difficulties in processing UML concepts and constructs in their memory and combining knowledge. In addition, prior knowledge of a procedural and functional paradigm may interfere with the learning of UML, which is based on an object-oriented paradigm.”</p> <p>(S67 - Bolloju, N.; 2006): “The majority of errors can be attributed to the inexperience of novice analysts in problem-structuring skills such as decomposition and to difficulties in applying object-oriented concepts related to distribution of responsibilities across participating objects.”</p> <p>(S71 – Siau, K, e Loo, P-L.; 2006): “The subjects claimed that insufficient course information, a crowded classroom, lack of a good textbook, and the user-unfriendliness of the CASE (computer-aided software engineering) tool created moderate difficulty in learning UML. Typical statements about this region include: “Spend too much time on class diagrams and not enough on others”, “I think we should have learned more about the unified process”, “Not seeing a completely finished product”, “Bugs in UML software (i.e., Rational Rose)”, “UML software is too expensive and slow to access”, “How to link diagram in Rose”, “Many UML examples contain errors”, “Textbook is hard to follow””</p>

COLABORAÇÃO PARA MODELAGEM DOS DIAGRAMAS UML

Categoria	Subcategoria	Citações
UML	Trabalho em Equipe ajuda a melhorar o entendimento dos conceitos UML	<p>(S24 – Boberic-Krsticev, D., and Tesendic, D.; 2013): “We asked students what they think about usefulness of team work, do they have problems to work in team and does the work on the project help them to better understand UML concepts. Generally speaking students of both programmes did not have problems to work in team, however, students of mathematics regarded team work more useful than students of computer science. For instance, 80% of students of mathematics regarded team work as very useful and only 61% of students of computer science agree with that statement.”</p> <p>(S40 – Moisan, S., e Rigault, J-P; 2008): “Some advocate a sort of "linguistic approach", reading an informal specification or design document</p>

Categoria	Subcategoria	Citações
		(depending on which phase you are in) and selecting nouns as candidates for classes and verbs as candidates for functions or methods. In our experience this does not work so well and nothing replaces a good understanding of the problem or its solution. For this, we simply promote group work, discussing and brainstorming in common.”
Diagrama de Classes	Trabalho colaborativo as vezes atrapalha	(S30 – Boustedt, J.; 2012): “Several informants describe that it is a good idea to develop software designs in collaboration with the project team, but collaborative work can also be a source of hard-to-handle problems. S01 points out a problem he experienced when his group designed a solution to a project assignment. After long discussions, the group agreed on a solution model (using class diagrams). However, when they started to implement the code, the subgroups started to do their own solutions that deviated from the design documents. And they did not go back and discuss this with the entire group and the design documents were never changed. Everybody knew this, but nobody wanted to start a new design discussion.”

LISTA COMPLETA DE ARTICULOS SELECCIONADOS

- S01** Kung, D., Lei, J.: An object-oriented analysis and design environment. In: 2016 IEEE 29th International Conference on Software Engineering Education and Training (CSEET), pp. 91–100 (2016).
- S02** Karasneh, B., Jolak, R., Chaudron, M.R.: Using examples for teaching software design: an experiment using a repository of UML class diagrams. In: 2015 Asia-Pacific Software Engineering Conference (APSEC), pp. 261–268 (2015).
- S03** Balaban, M., Maraee, A., Sturm, A., Jelnov, P.: A pattern-based approach for improving model quality. *Software & Systems Modeling*14(4), 1527–1555 (2015).
- S04** Stikkolorum, D.R., Ho-Quang, T., Chaudron, M.R.: Revealing students' UML class diagram modelling strategies with webUML and logviz. In: 2015 41st Euromicro Conference on Software Engineering and Advanced Applications, pp. 275–279 (2015).
- S05** Dranidis, D., Stamatopoulou, I., Ntika, M.: Learning and practicing systems analysis and design with studentUML. In: Proceedings of the 7th Balkan Conference on Informatics, p. 41, (2015).
- S06** Weerasinghe, A., Evans, B.: UML-it: An its to teach multiple modelling tasks. In: International Conference on Artificial Intelligence in Education, pp. 816–819 (2015).
- S07** Stikkolorum, D.R., Ho-Quang, T., Karasneh, B., Chaudron, M.R.: Uncovering students' common difficulties and strategies during a class diagram design process: an online experiment. In: EduSymp@MoDELS, pp. 29–42 (2015).
- S08** Bruel, J.M., Maggi, B., Letavernier, C.: Papyrus for education. In: 2nd International Workshop on Open Source Software for Model Driven Engineering, OSS4MDE2015, vol. 1541, pp. 38–43 (2015).
- S09** Petre, M.: “no shit” or “oh, shit!”: responses to observations on the use of UML in professional practice. *Software & Systems Modeling*13(4), 1225–1235 (2014).
- S10** Yamazaki, S., Jiromaru, T.: Instructional design of exercise-centric teaching materials on UML modeling. In:2014 IIAI 3rd International Conference on Advanced Applied Informatics, pp. 342–346 (2014).
- S11** Fitsilis, P., Gerogiannis, V.C., Anthopoulos, L.: Role of unified modelling language in software development in greece—results from an exploratory study. *IET software* 8(4), 143–153 (2013).
- S12** Mich, L.: Teaching requirements analysis: A student project framework to bridge the gap between business analysis and software engineering. In: REET, pp. 20–25 (2014).
- S13** Gallardo, J., Molina, A.I., Bravo, C., Gallego, F.: A system for collaborative building of use case models: Communication analysis and experiences: Experiences of use and lessons learned from the use of the space-design tool in the domain of use case diagrams. In: 2014 9th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE), pp. 1–10 (2014).
- S14** Reggio, G., Leotta, M., Ricca, F., Clerissi, D.: What are the used activity diagram constructs? a survey. In: 2nd International Conference on Model Driven Engineering and Software Development (MODELSWARD), pp. 87–98 (2014).
- S15** Kayama, M., Ogata, S., Masymoto, K., Hashimoto, M., Otani, M.: A practical conceptual modeling teaching method based on quantitative error analyses for novices learning to create errorfree simple class diagrams. In:2014 IIAI 3rd International Conference on Advanced Applied Informatics, pp. 616–622 (2014).
- S16** Scanniello, G., Erra, U.: Distributed modeling of use case diagrams with a method based on think-pair-square: Results from two controlled experiments. *Journal of Visual Languages & Computing* 25(4), 494–517 (2014).
- S17** Tanaka, T., Mori, K., Hashiura, H., Hazeyama, A.,Komiya, S.: Proposals of a method detecting learners' difficult points in object modeling exercises and a tool to support the method. *International Journal of Software Innovation (IJSI)* 3(1), 63–74 (2015).
- S18** Reggio, G., Leotta, M., Ricca, F.: Who knows/uses what of the UML: a personal opinion survey. In: International Conference on Model Driven Engineering Languages and Systems, pp. 149–165 (2014).
- S19** Jurgelaitis, M., Drungilas, V.,Ceponiene, L.: Gamified moodle course for teaching UML. *Baltic Journal of Modern Computing* 6(2), 119–127 (2018).
- S20** Kruus, H., Robal, T., Jervan, G.: SysML in systems engineering course. In: 10th European Workshop on Microelectronics Education (EWME), pp. 177–181 (2014).
- S21** Mitra, S.: Using UML modeling to facilitate three-tier architecture projects in software engineering courses. *ACM Transactions on Computing Education (TOCE)* 14(3), p. 17 (2014).
- S22** Barros, J.P.: From concrete to abstract: about teaching UML class diagrams to novice programmers. In: Proceedings of the 8th International Joint Conference on Software Technologies, pp. 278–283 (2013).

- S23** Akayama, S., Kuboaki, S., Hisazumi, K., Futagami, T., Kitasuka, T.: Development of a modeling education program for novices using model-driven development. In: 2012 Workshop on Embedded and Cyber-Physical Systems Education, WESE 2012 (2013).
- S24** Boberic-Krsticev, D., Telendic, D.: Experience in teaching ooad to various students. *Informatics in Education* 12(1), 43–58 (2013).
- S25** Karasneh, B., Chaudron, M.R.: Online img2UML repository: An online repository for UML models. In: EESS-MOD@ MoDELS, pp. 61–66 (2013).
- S26** Fernandez-Saez, A.M., Chaudron, M.R., Genero, M.: Exploring costs and benefits of using UML on maintenance: Preliminary findings of a case study in a large it department. In: EESSMOD@MoDELS, pp. 33–42 (2013).
- S27** Gross, A., Jurkiewicz, J., Doerr, J., Nawrocki, J.: Investigating the usefulness of notations in the context of requirements engineering. In: 2012 Second IEEE International Workshop on Empirical Requirements Engineering (EmpiRE), pp. 9–16 (2012).
- S28** Garbe, H.: Intelligent assistance in a problem-solving environment for UML class diagrams by combining a generative system with constraints (2012).
- S29** Ogata, S., Matsuura, S.: Training of requirements analysis modeling with UML-based prototype generation tool. In: Proceedings of the 5th India Software Engineering Conference, pp. 105–108 (2012).
- S30** Boustedt, J.: Students' different understandings of class diagrams. *Computer Science Education* 22(1), pp. 29–62 (2012)
- S31** Sien, V.Y.: An investigation of difficulties experienced by students developing unified modelling language (UML) class and sequence diagrams. *Computer Science Education* 21(4), 317–342 (2011).
- S32** Briand, L., Labiche, Y., Madrazo-Rivera, R.: An experimental evaluation of the impact of system sequence diagrams and system operation contracts on the quality of the domain model. In: 2011 International Symposium on Empirical Software Engineering and Measurement, pp. 157–166 (2011).
- S33** Sikkel, K., Daneva, M.: Getting the client into the loop in information system modelling courses. In: 2011 6th international Workshop on Requirements Engineering Education and Training, pp. 1–4 (2011).
- S34** Vogel-Heuser, B., Braun, S., Obermeier, M., Sommer, K., Seidel, T., Johannes, C.: Modeling order effects on errors in object oriented modeling for machine and plant automation from an educational point of view. In: ETFA 2011, pp. 1–4 (2011).
- S35** Broy, M., Cengarle, M.V.: UML formal semantics: lessons learned. *Software and Systems Modeling* 10(4), 441–446 (2011).
- S36** Holland, J., Baghaei, N., Mathews, M., Mitrovic, A.: The effects of domain and collaboration feedback on learning in a collaborative intelligent tutoring system. In: International Conference on Artificial Intelligence in Education, pp. 469–471 (2011).
- S37** Sien, V.Y.: Implementation of the concept-driven approach in an object-oriented analysis and design course. In: International Conference on Model Driven Engineering Languages and Systems, pp. 55–69 (2010).
- S38** Phalp, K., Adlem, A., Jeary, S., Vincent, J., Kanyaru, J.: The role of comprehension in requirements and implications for use case descriptions. *Software Quality Journal* 19(2), 461–486 (2011).
- S39** Sikkel, K., Daneva, M.: Teaching consistency of UML specifications. In: 2010 5th International Workshop on Requirements Engineering Education and Training, pp. 17–19 (2010).
- S40** Moisan, S., Rigault, J.P.: Teaching object-oriented modeling and UML to various audiences. In: International Conference on Model Driven Engineering Languages and Systems, pp. 40–54 (2009).
- S41** Andersson, H., Herzog, E., Johansson, G., Johansson, O.: Experience from introducing unified modeling language/systems modeling language at saab aerosystems. *Systems Engineering* 13(4), 369–380 (2010).
- S42** Tang, W., Ning, B., Xu, T., Zhao, L.: Scenario-based modeling and verification of system requirement specification for the european train control system. *WIT Transactions on the Built Environment* 114, 759–770 (2010).
- S43** Zhang, L.: Applying case method approach to a unified modeling language curriculum. In: 2010 2nd International Conference on Education Technology and Computer, vol. 4, pp. V4–418 (2010).
- S44** Borstler, J., Dranidis, D., Ramollari, E., Trapp, S., Heintz, M., Weber, S.: Sharing and discussing UML modeling exercises in a PLE. In: ITiCSE, p. 301 (2010).
- S45** Sharif, B., Maletic, J.I.: The effects of layout on detecting the role of design patterns. In: 2010 23rd IEEE Conference on Software Engineering Education and Training, pp. 41–48 (2010).
- S46** Figl, K., Mendling, J., Strembeck, M., Recker, J.: On the cognitive effectiveness of routing symbols in process modeling languages. In: International Conference on Business Information Systems, pp. 230–241 (2010).

- S47** Figl, K., Mendling, J., Strembeck, M., Recker, J.: On the cognitive effectiveness of routing symbols in process modeling languages. In: International Conference on Business Information Systems, pp. 230–241 (2010).
- S48** Ishii, N., Nagao, Y., Suzuki, Y., Fujiyoshi, H., Fujii, T.: Designing a programming course to foster creativity using UML modeling template. In: CSEDU (2), pp. 104–109 (2009)
- S49** Labiche, Y.: The UML is more than boxes and lines. In: International Conference on Model Driven Engineering Languages and Systems, pp. 375–386 (2008).
- S50** Demuth, B., Weigel, D.: Web based software modeling exercises in large scale software engineering courses. In: 22nd Conference on Software Engineering Education and Training, pp. 138–141 (2009).
- S51** Hai, L.: The role of collaboration diagrams in OO software engineering student projects. In: 2009 22nd Conference on Software Engineering Education and Training, pp. 93–100 (2009).
- S52** VanderMeer, D., et al.: Applying learner-centered design principles to UML sequence diagrams. *Journal of Database Management (JDM)* 20(1), 25–47 (2009).
- S53** Song, I.Y., Khare, R., An, Y., Hilsbos, M.: A multi-level methodology for developing UML sequence diagrams. In: International Conference on Conceptual Modeling, pp. 114–127 (2008).
- S54** Lin, H., Liu, C., Lin, P.: A method of elicitation teaching for object-oriented analysis and design curriculum. In: 2008 The 9th International Conference for Young Computer Scientists, pp. 2495–2500 (2008).
- S55** Paterson, J.H., Cheng, K.F., Haddow, J.: Patterncoder: A programming support tool for learning binary class associations and design patterns. *ACM Transactions on Computing Education (TOCE)* 9(3), 16 (2009).
- S56** Tourtoglou, K., Virvou, M.: User stereotypes concerning cognitive, personality and performance issues in a collaborative learning environment for UML. In: *New Directions in Intelligent Interactive Multimedia*, pp. 385–394 (2008).
- S57** Alonso, M., Py, D., Lemeunier, T.: A learning environment for object-oriented modeling, supporting metacognitive regulations. In: 2008 Eighth IEEE International Conference on Advanced Learning Technologies, pp. 69–73 (2008).
- S58** Auxepaules, L., Py, D., Lemeunier, T.: A diagnosis method that matches class diagrams in a learning environment for object-oriented modeling. In: 2008 Eighth IEEE International Conference on Advanced Learning Technologies, pp. 26–30 (2008).
- S59** Debusse, J.C., Stiller, T.: Technologies and strategies for integrating object-oriented analysis and design education with programming. In: 19th Australian Conference on Software Engineering (ASWEC 2008), pp. 97–103 (2008).
- S60** Tourtoglou, K., Virvou, M.: User stereotypes concerning cognitive, personality and performance issues in a collaborative learning environment for UML. In: *New Directions in Intelligent Interactive Multimedia*, pp. 385–394 (2008).
- S61** Baghaei, N., Mitrovic, A.: From modelling domain knowledge to metacognitive skills: extending a constraint-based tutoring system to support collaboration. In: International Conference on User Modeling, pp. 217–227 (2007).
- S62** Sin, T.: Improving usability of analysis sequence diagram in transaction - oriented applications. *AMCIS 2007 Proceedings* p. 419 (2007).
- S63** Siau, K., Nah, F., Eschenbrenner, B., Guru, A.: An augmented approach to support collaborative distance learning of unified modeling language. *AMCIS 2007 Proceedings* p. 301 (2007).
- S64** Szmurlo, R., Smialek, M.: Teaching software modeling in a simulated project environment. In: International Conference on Model Driven Engineering Languages and Systems, pp. 301–310 (2006).
- S65** Baghaei, N., Mitrovic, A., Irwin, W.: Supporting collaborative learning and problem-solving in a constraint-based cscl environment for UML class diagrams. *International Journal of Computer Supported Collaborative Learning* 2(2-3), 159–190 (2007).
- S66** Wrycza, S., Marcinkowski, B.: UML 2 academic course — methodological background and survey benchmarking. In: 23rd Annual Conference for Information Systems Educators, vol. 23 (2006).
- S67** Bolloju, N.: Exploring quality dependencies among UML artifacts developed by novice systems analysts. *AMCIS 2006 Proceedings* p. 472 (2006).
- S68** Dobing, B., Parsons, J.: How UML is used. *Communications of the ACM* 49(5), 109–113 (2006).
- S69** Kuzniarz, L., Staron, M.: Best practices for teaching UML based software development. In: International Conference on Model Driven Engineering Languages and Systems, pp. 320–332 (2005).
- S70** Giese, H., Roques, P., Lethbridge, T.C.: Summary of the educator’s symposium. In: International Conference on Model Driven Engineering Languages and Systems, pp. 302–305 (2005).
- S71** Siau, K., Loo, P.P.: Identifying difficulties in learning UML. *Information Systems Management* 23(3), 43–51 (2006)

- S72** Lázaro Carrascosa, C., Velazquez Iturbide, A., Hijon Neira, R., Hernan Losada, I.: TextOO: An object-oriented learning tool based on enunciates (2006).
- S73** Lange, C.F., DuBois, B., Chaudron, M.R., Demeyer, S.: An experimental investigation of UML modeling conventions. In: International Conference on Model Driven Engineering Languages and Systems, pp. 27–41 (2006).
- S74** Pavlov, V.L., Yatsenko, A.: Using pantomime in teaching OOA & OOD with UML. In: 18th Conference on Software Engineering Education & Training (CSEET'05), pp. 77–84 (2005).
- S75** Cheng, B.H., Stephenson, R., Berenbach, B.: Lessons learned from automated analysis of industrial UML class models (an experience report). In: International Conference on Model Driven Engineering Languages and Systems, pp. 324–338 (2005).
- S76** Baghaei, N., Mitrovic, A.: Collect-UML: supporting individual and collaborative learning of UML class diagrams in a constraint-based intelligent tutoring system. In: International Conference on Knowledge-Based and Intelligent Information and Engineering Systems, pp. 458–464 (2005).
- S77** Wrycza, S., Marcinkowski, B.: UML 2 teaching at post-graduate studies—prerequisites and practice. In: Proceedings of ISECON, vol. 22 (2005).
- S78** Ciancarini, P.: On the education of future software engineers. In: 27th International Conference on Software Engineering (ICSE 2005), pp. 649–650 (2005).
- S79** Hallinan, S., Gibson, J.P.: A graduate's role in technology transfer: From requirements to design with UML. In: IASTED Conf. on Software Engineering, pp. 94–99 (2005).
- S80** Bunse, C., Grützner, I., Ochs, M., Peper, C., Steinbach-Nordmann, S.: Applying a blended learning strategy for software engineering education. In: Proceedings of the 18th Conference on Software Engineering Education and Training (CSEE&T) (2005).
- S81** Kinjo, T., Ohgame, Y., Hazeyama, A.: An object-oriented modeling learning support system using inspection comments and its evaluation (2006).
- S82** Pettit, R.G., Street, J.A.: Lessons learned applying UML in the design of mission critical software. In: International Conference on the Unified Modeling Language, pp. 129–137 (2004).
- S83** Grossman, M., Aronson, J.E., McCarthy, R.V.: Does UML make the grade? insights from the software development community. *Information and Software Technology* 47(6), 383–397 (2005).
- S84** Irani, P.: Notations for software engineering class structures. In: International Conference on Theory and Application of Diagrams, pp. 441–445 (2004).
- S85** Hilsbos, M., Song, I.Y.: Use of tabular analysis method to construct UML sequence diagrams. In: International Conference on Conceptual Modeling, pp. 740–752 (2004).
- S86** Auer, M., Tschurtschenthaler, T., Biffl, S.: A flyweight UML modelling tool for software development in heterogeneous environments (2003).
- S87** Alphonse, C., Ventura, P.: QuickUML: a tool to support iterative design and code development. In: Companion of the 18th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications, pp. 80–81 (2003).
- S88** Hansen, K.M., Ratzner, A.V.: Tool support for collaborative teaching and learning of object-oriented modeling. *ACM SIGCSE Bulletin* 34(3), 146–150 (2002).
- S89** Alphonse, C., Ventura, P.: Object orientation in cs1-cs2 by design. *ACM SIGCSE Bulletin* 34(3), 70–74 (2002).
- S90** Petrie, H., Schlieder, C., Blenkhorn, P., Evans, G., King, A., O'Neill, A.M., Ioannidis, G.T., Gallagher, B., Crombie, D., Mager, R., et al.: Tedub: A system for presenting and exploring technical drawings for blind people. In: International Conference on Computers for Handicapped Persons, pp. 537–539 (2002).
- S91** Isoda, S.: Object-oriented real-world modeling revisited. *Journal of Systems and Software* 59(2), 153–162 (2001).
- S92** Moritz, S.H., Wei, F., Parvez, S.M., Blank, G.D.: From objects first to design-first with multimedia and intelligent tutoring. In: *ACM SIGCSE Bulletin*, vol. 37, pp. 99–103 (2005).
- S93** Kruus, H., Robal, T., Jervan, G.: Sysml in systems engineering course. In: 10th European Workshop on Microelectronics Education (EWME), pp. 177–181 (2014).
- S94** Hankley, W.: On teaching software architecture and design. *age8*, 1 (2003).
- S95** Kanakaraddi, S.G., Naragund, J.G., Chikaraddi, A.K.: Active learning methods for teaching ooad course. In: IEEE International Conference in MOOC, Innovation and Technology in Education (MITE), pp. 47–52 (2013).
- S96** Vachharajani, V., Pareek, J., Gulabani, S.: Effective label matching for automatic evaluation of use-case diagrams. In: IEEE Fourth International Conference on Technology for Education, pp. 172–175 (2012).

- S97** Gallant, R.: Thinking fast: Patterns of cognitive error in software engineering education and practice. In: IEEE International Conference on Software Science, Technology and Engineering (SWSTE), pp. 43–50 (2016).
- S98** Bunse, C., Grutzner, I., Peper, C., Steinbach-Nordmann, S., Vollmers, C.: Coaching professional software developers – an experience report. In: 19th Conference on Software Engineering Education & Training (CSEET'06), pp. 123–130 (2006).
- S99** Dori, D., Wengrowicz, N., Dori, Y.J.: A comparative study of languages for model-based systems-of-systems engineering (MBSSE). In: World Automation Congress (WAC), pp. 790–796 (2014).
- S100** Pavlov, V.L., Yatsenko, A.: Using pantomime in teaching OOA& OOD with UML. In: 18th Conference on Software Engineering Education & Training (CSEET'05), pp. 77–84 (2005).
- S101** Dixon, M.: A single case environment for teaching and learning. In: ACM SIGCSE Bulletin, vol. 36, pp. 271–271 (2004).
- S102** Duschl, K., Obermeier, M., Vogel-Heuser, B.: An experimental study on UML modeling errors and their causes in the education of model driven plc programming. In: Global Engineering Education Conference (EDUCON), pp. 119–128 (2014).
- S103** Frezza, S., Andersen, W.: Interactive exercises to support effective learning of UML structural modeling. In: Proceedings 36th Annual Conference on Frontiers in Education, pp. 7–12 (2006).
- S104** Vogel-Heuser, B., Obermeier, M., Braun, S., Sommer, K., Jobst, F., Schweizer, K.: Evaluation of a UML-based versus an iec 61131-3-based software engineering approach for teaching plc programming. IEEE Transactions on Education 56(3), 329–335 (2013).
- S105** Scholtz, B., Wesson, J.: How 'learnable' are case tools in diverse user communities? In: IFIP Human-Computer Interaction Symposium, pp. 83–97 (2008).
- S106** Tabrizi, M.H., Collins, C., Ozan, E., Li, K.: Implementation of object-orientation using UML in entry level software development courses. In: Proceedings of the 5th conference on Information technology education, pp. 128–131 (2004).
- S107** Liu, X.: The research on teaching methods of object-oriented approach in management information systems curriculum. In: Emerging Technologies for Information Systems, Computing, and Management, pp. 1107–1113 (2013).
- S109** Hasker, R.W., Rowe, M.: UMLint: Identifying defects in UML diagrams. In: American Society for Engineering Education. American Society for Engineering Education (2011).
- S110** Parvez, S.M., Blank, G.D.: A pedagogical framework to integrate learning style into intelligent tutoring systems. Journal of Computing Sciences in College 22(3), 183–189 (2007).
- S111** Beheshti, R., Dado, E.: Simplified UML techniques for system development in an educational setting. In: 6th International Conference on Information Technology Based Higher Education and Training, pp. S2C–1 (2005).
- S112** Hasker, R.W.: Teaching basic class diagram notation with UMLgrader. age24, 1.
- S113** Dagdeviren, H., Juric, R., Lees, P.: Experiences of teaching UML within the information systems curriculum. In: 26th International Conference on Information Technology Interfaces, pp. 381–386 (2004).
- S114** Pettit, R.G., Street, J.A.: Lessons learned applying UML in the design of mission critical software. In: International Conference on the Unified Modeling Language, pp. 129–137 (2004).
- S115** Alkoshman, M.M.: Unified modeling language and enhanced entity relationship: an empirical study. International Journal of Database Theory and Application 8(3), 215–227 (2015).
- S116** Wang, Y., Su, R., Li, G.: Research on pbl and lbl double track teaching model in unified modeling language teaching based on outstanding engineers. In: Informatics and Management Science V, pp. 381–386 (2013).
- S117** Lewis, T.L., Perez-Quinones, M.A., Rosson, M.B.: A comprehensive analysis of object-oriented design: towards a measure of assessing design ability. In: 34th Annual Frontiers in Education, pp. S3H–16 (2004).
- S118** Doerschuk, P.: Incorporating team software development and quality assurance in software engineering education. In: 34th Annual Frontiers in Education, 2004.FIE 2004., pp. F1C–7 (2004).
- S119** Giordano, D., Maiorana, F.: Object oriented design through game development in xna. In: 2013 3rd Interdisciplinary Engineering Design Education Conference, pp. 51–55 (2013).
- S120** Baar, T.: Improving software engineering education by modeling real-world implementations. In: Proceedings of the 8th edition of the Educators' Symposium, pp. 36–39 (2012).
- S121** Demuth, B.: How should teaching modeling and programming intertwine? In: Proceedings of the 8th edition of the Educators' Symposium, pp. 32–35 (2012).

- S122** Sedrakyan, G., De Weerd, J., Snoeck, M.: Process mining enabled feedback: “tell me what i did wrong” vs. “tell me how to do it right”. *Computers in human behavior* 57, 352–376 (2016).
- S123** Sedrakyan, G., Snoeck, M., Poelmans, S.: Assessing the effectiveness of feedback enabled simulation in teaching conceptual modeling. *Computers & Education* 78, 367–382 (2014).
- S124** Byrne, B.M., Qureshi, Y.S.: UML class diagram or entity relationship diagram? an object-relational conceptual impedance mismatch. In: *Proceedings of 6th International Conference of Education, Research and Innovation (ICERI2013)*, pp. 3594–3604 (2013).
- S125** Sedrakyan, G., Snoeck, M.: Technology-enhanced support for learning conceptual modeling. In: *Enterprise, Business Process and Information Systems Modeling*, pp. 435–449 (2012).
- S126** Hutchinson, J., Whittle, J., Rouncefield, M., Kristoffersen, S.: Empirical assessment of mde in industry. In: *Proceedings of the 33rd international conference on software engineering*, pp. 471–480 (2011).
- S127** Hutchinson, J., Rouncefield, M., Whittle, J.: Model-driven engineering practices in industry. In: *Proceedings of the 33rd International Conference on Software Engineering*, pp. 633–642 (2011).
- S128** Boberic-Krsticev, D., Tešendić, D.: Teaching object-oriented modeling using UML. In: *AIP Conference Proceedings*, vol. 1389, pp. 810–812. AIP (2011).
- S129** Tourtoglou, K., Virvou, M.: User stereotypes concerning cognitive, personality and performance issues in a collaborative learning environment for UML. In: *New Directions in Intelligent Interactive Multimedia*, pp. 385–394 (2008).
- S130** Baghaei, N., Mitrovic, A., Irwin, W.: Supporting collaborative learning and problem-solving in a constraint-based CSCL environment for UML class diagrams. *International Journal of Computer-Supported Collaborative Learning* 2(2–3), 159–190 (2007).
- S131** Copperman, E., Beeri, C., Ben-Zvi, N.: Visual modelling of learning processes. *Innovations in Education and Teaching International* 44(3), 257–272 (2007).
- S132** Anda, B., Hansen, K., Gulleisen, I., Thorsen, H.K.: Experiences from introducing UML-based development in a large safety-critical project. *Empirical Software Engineering* 11(4), 555–581 (2006).
- S133** Ogata, S., Kayama, M., Okano, K.: Smart-learning: State machine simulators for developing thinking skills. In: *17th International Conference on Advanced Learning Technologies (ICALT)*, pp. 81–83 (2017).
- S134** Bera, P., Poels, G.: The effects of construct redundancy on readers’ understanding of conceptual models. *Journal of Database Management (JDM)* 28(3), 1–25 (2017).
- S135** Dittmar, A., Buchholz, G., Kühn, M.: Effects of facilitation on collaborative modeling sessions with a multitouch UML editor. In: *2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering Education and Training Track (ICSE-SEET)*, pp. 97–106 (2017).
- S136** Outair, A., Lyhyaoui, A., Tanana, M.: Towards an automatic evaluation of UML class diagrams by graph transformation. *International Journal of Computer Applications* 95(21) (2014).
- S137** Thevathayan, C., Hamilton, M.: Imparting software engineering design skills. In: *Proceedings of the Nineteenth Australasian Computing Education Conference*, pp. 95–102 (2017).
- S138** Stikkolorum, D.R., Chaudron, M.: Teaching of agile UML modeling: Recommendations from students’ reflections. In: *CIBSE*, pp. 127–140 (2017).
- S139** Reischmann, T., Kuchen, H.: Towards an assessment tool for advanced software engineering skills. In: *Proceedings of the 16th Koli Calling International Conference on Computing Education Research*, pp. 81–90 (2016).
- S140** Stikkolorum, D.R., Chaudron, M.R.: A workshop for integrating UML modelling and agile development in the classroom. In: *Proceedings of the Computer Science Education Research Conference 2016*, pp. 4–11 (2016).
- S141** Rodrigues, C.S.C., Werner, C.M., Landau, L.: Visar3d: an innovative 3d visualization of UML models. In: *Proceedings of the 38th International Conference on Software Engineering Companion*, pp. 451–460 (2016).
- S142** Hidayat, A., Utomo, V.G.: Adaptive online module prototype for learning unified modelling language (UML). *International Journal of Electrical and Computer Engineering* 6(6), 2931 (2016).
- S143** Miyashita, H., Tai, H., Amano, S.: Controlled modeling environment using flexibly-formatted spreadsheets. In: *Proceedings of the 36th International Conference on Software Engineering*, pp. 978–988 (2014).
- S144** Box, R., Whitelaw, M.: Experiences when migrating from structured analysis to object-oriented modelling. In: *Proceedings of the Australasian conference on Computing education*, pp. 12–18 (2000).
- S145** Tiwari, S., Gupta, A.: Investigating comprehension and learnability aspects of use cases for software specification problems. *Information and Software Technology* 91, 22–43 (2017).

- S146** Kung, D., Lei, J.: An object-oriented analysis and design environment. In: IEEE 29th International Conference on Software Engineering Education and Training (CSEET), pp. 91–100 (2016).
- S147** Mouenis, A.T., Souhaib, A., Mohamed, K.: Development of baseyan networks from unified modeling language for learner modelling. *International Journal of Advanced Computer Science and Applications* 6(2) (2015).
- S148** Tsarmpou, P., Tambouris, E.: Using learning analytics to enhance UML use case diagrams assimilation in a distance education course. *International Journal of Learning Technology* 10(4), 274–290 (2015).
- S149** Bolloju, N., Schneider, C., Vogel, D.: Asymmetrical effects of using positive and negative examples on object modeling. In: *Information Systems Development*, pp.85–96 (2011).
- S150** Stoecklin, S., Williams, D.D., Swain, R.: Understanding objectoriented systems specifications using familiar systems. In: *Proceedings. 1998 International Conference Software Engineering: Education and Practice (Cat. No.98EX220)*, pp. 10–15 (1998).
- S151** de la Vara, J.L., Espinoza, H.: Dealing with software model quality in practice: Experience in a research project. In: *13th International Conference on Quality Software*, pp. 396–405 (2013). S152 Kuhn, A., Murphy, G.C.: Lessons learned from evaluating MDE abstractions in an industry field study. In: *Proceedings of the Second Edition of the International Workshop on Experiences and Empirical Studies in Software Modelling*, p. 3 (2012).
- S153** Siau, K., Cao, Q.: How complex is the unified modeling language? In: *Advanced Topics in Database Research, Volume 1*, pp. 294–306. Igi Global (2002).
- S154** Portillo, J.A.P.S., Campos, P.G.: The jigsaw technique: experiences teaching analysis class diagrams. In: *Mexican International Conference on Computer Science*, pp. 289–293 (2009).
- S155** Kuriyama, K., Sakai, K.: An agent oriented environment for collaborative learning-lessons learned through vocational training on software design with UML. In: *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, pp. 567–574 (2007) .
- S156** VanderMeer, D., et al.: Applying learner-centered design principles to UML sequence diagrams. *Journal of Database Management (JDM)*20(1), 25–47 (2009).
- S157** Burton, P.J., Bruhn, R.E.: Using UML to facilitate the teaching of object-oriented systems analysis and design. *Journal of Computing Sciences in Colleges*19(3), 278–290 (2004).
- S158** Baghaei, N., Mitrovic, A.: From modelling domain knowledge to metacognitive skills: extending a constraint-based tutoring system to support collaboration. In: *International Conference on User Modeling*, pp. 217–227 (2007).
- S159** Mlikinen, E., Systa, T.: Masan interactive synthesizer to support behavioral modeling in UML. In: *Proceedings of the 23rd International Conference on Software Engineering (ICSE 2001)*, pp. 15–24 (2001).
- S160** Domeshek, E.A., Holman, E.: Oodle: a prototype object-oriented design learning environment. In: *Companion of the 17th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications*, pp. 64–65 (2002).
- S161** Mitra, S., Rao, T.: Discovering design patterns in software behavior models. *Journal of Computing Sciences in Colleges* 32(6), 120–129 (2017).
- S162** Hadjerrouit, S.: Constructivism as guiding philosophy for software engineering education. *Acm SIGCSE Bulletin* 37(4), 45–49 (2005).
- S163** Gogolla, M., Vallecillo, A.: On explaining modeling principles with modeling examples: a classification catalog. In: *Proceedings of the 8th edition of the Educators’Symposium*, pp. 28–31M (2012).
- S164** Parvez, S.M., Blank, G.D.: A pedagogical framework to integrate learning style into intelligent tutoring systems. *Journal of Computing Sciences in Colleges* 22(3), 183–189 (2007).
- S165** El Ahmar, Y., Le Pallec, X., G’erard, S., Ho-Quang, T.: Visual variables in UML: a first empirical assessment. In: *Human Factors in Modeling* (2017).
- S166** Tazin, A.: UML class diagram composition using software requirements specifications. In: *MODELS (Satellite Events)*, pp. 511–515 (2017).
- S167** Badreddin, O., Khandoker, R., Forward, A., Masmali, O., Lethbridge, T.C.: A decade of software design and modeling: A survey to uncover trends of the practice. In: *Proceedings of the 21th ACM/IEEE International Conference on Model Driven Engineering Languages and Systems*, pp. 245–255 (2018).
- S168** K`astner, A., Gogolla, M., Selic, B.: From (imperfect) object diagrams to (imperfect) class diagrams: New ideas and vision paper. In: *Proceedings of the 21th ACM/IEEE International Conference on Model Driven Engineering Languages and Systems*, pp. 13–22 (2018).
- S169** Jurgelaitis, M., Ceponien, L., ^ Ceponis, J., Drungilas, V.: Implementing gamification in a university-level UML modeling course: A case study. *Computer Applications in Engineering Education* 27(2), 332–343 (2019).

- S170** Silva, W., Gadelha, B., Steinmacher, I., Conte, T.: What are the differences between group and individual modeling when learning UML? In: Proceedings of the XXXII Brazilian Symposium on Software Engineering, pp. 308–317 (2018)
- S171** Felisbino, C.M., Neto, A.G.S.S., Bastos, L.C.: Supporting to the teaching and learning process in object orientation during the construction of class diagrams. In: Brazilian Symposium on Software Engineering, pp. 338–347 (2018).
- S172** Yu, Z., Xiong, Z.: Comparative analyses for the performance of rational rose and visio in software engineering teaching. In: Journal of Physics: Conference Series, vol. 1087, p. 062041, IOP Publishing (2018).
- S173** Shen, Z., Tan, S., Siau, K.: Challenges in learning unified modeling language: From the perspective of diagrammatic representation and reasoning. Communications of the Association for Information Systems 43(1), 30 (2018).
- S174** Grant, E.S., Ajjimaporn, P.: Pedagogical benefits from an exercise in reverse engineering for an aviation software systems. In: CSEDU (2), pp. 179–188 (2018).
- S175** Zayan, D., Sarkar, A., Antkiewicz, M., Maciel, R.S.P., Czarnecki, K.: Example-driven modeling: on effects of using examples on structural model comprehension, what makes them useful, and how to create them. Software & Systems Modeling pp. 1–27 (2018).
- S176** Silva, W.A.F., Steinmacher, I.F., Conte, T.U.: Is it better to learn from problems or erroneous examples? In: 30th Conference on Software Engineering Education and Training (CSEE&T), pp. 222–231 (2017).
- S177** Beimel, D., Kedmi-Shahar, E.: Improving the identification of functional system requirements when novice analysts create use case diagrams: the benefits of applying conceptual mental models. Requirements Engineering, pp. 1–20 (2018).
- S178** Ozkaya, M.: Do the informal & formal software modeling notations satisfy practitioners for software architecture modeling? Information and Software Technology 95, 15–33 (2018).
- S179** Burgueño, L., Vallecillo, A., Gogolla, M.: Teaching UML and OCL models and their validation to software engineering students: an experience report. Computer Science Education 28(1), 23–41 (2018).
- S180** Bogdanova, D., Snoeck, M.: Learning from errors: Error-based exercises in domain modelling pedagogy. In: IFIP Working Conference on The Practice of Enterprise Modeling, pp. 321–334 (2018).
- S181** Ondik, J., Olejar, M., Rastocny, K., Bielikova, M.: Activity-based model synchronization and defects detection for small teams. In: 2017 International Conference on Software Quality, Reliability and Security Companion (QRS-C), pp. 8–15 (2017)

FERRAMENTAS IDENTIFICADAS NO MSL

Ferramentas		
Subcategoria	Nome da Tecnologia	ID do artigo
Ferramentas CASE	IDE for OOAD	S01
	LogViz	S04
	UML editor WebUML	S07
	Papyrus	S08
	UMLet	S86
	QuickUML	S96
	Use – Case Diagram Editor	S04
	SMart-Learning	S133
	Multi-Touch UML Editor	S135
	Spreadsheet-based modeling approach	S143
	Minimally Adequate Synthesizer (MAS)	S156
	Rational Rose	S169
Vision	S169	
Sistemas de avaliação dos diagramas	Graph matching system for class diagram	S136
	A prototype for the automated assessment of UML class diagrams	S139
	Method to detect learners' difficult points.	S17
	Approach and algorithms for automating the selection of the best sub-diagrams	S163
	SmallTEAmHelper	S179
	Tool to detect learners' difficult points.	S17
Sistemas de tutoria de modelagem	StudentUML	S05
	UML-IT	S06, S108
	SPACE-DESIGN	S13
	Tool to detect learners' difficult points.	S17
	Intelligent Problem-Solving Environment for UML Class Diagrams	S28
	COLLECT-UML	S36, S61, S65, S76, S130
	Courseware for UML/SysML modelling	S44
	UML Checker	S50
	UML Tutor	S50
	PatternCoder	S55
	AUTO-COLLEAGUE	S56, S60, S129
	Diagram UML	S57
	diagnostic method + Diagram	S58
	TextOO	S72
	Ideogramic UML	S88
	TeDUB	S90
	QSEE-SuperLite	S101
	Interactive Exercise	S103
	UMLGrader	S112
	Agent-Oriented Collaborative Learning System	S154
	A Prototype Object-Oriented Design Learning Environment (OODLE)	S157
	CIMEL-ITS	S161
	Method to detect learners' difficult points.	S17
	Tool to detect learners' difficult points.	S17
	Tool of Model-Driven Requirements Analysis Using Unified Modeling Language	S29
	MERODE	S122, S123, S125
	Tablet PCs e Web Conference Software	S63
	Geon	S84
	VisAr3D environment	S141
	Prototype of adaptive online module based on the student's learning style categories	S142
	Online Learning System	S162, S166, S180
	Tool for create the class diagram	S163
Online Img2UML Repository	S25	
Métodos Educacionais	Problem-based learning	S116, S189
	Learning from Erroneous Examples	S180
	Teaching Method for Conceptual Modeling Based on Error Analyses of Class Diagrams	S15
	Teaching Methods of Object-Oriented Approach in MIS Curriculum	S107

Ferramentas		
Subcategoria	Nome da Tecnologia	ID do artigo
	Method based on think-pair-square	S16
	Method to detect learners' difficult points	S17
	Rules-based pedagogical method for identifying defects in models	S49
	PBL + LBL Double Track Teaching Method	S116
	Pseudo Real-World Modeling Methods	S91
	Lecture based learning	S116
	Project-based learning with weekly quizzes, tests and active learning tasks	S137
	Positive and Negative Examples	S148
	Learning by Similar Systems	S149
	Jigsaw Technique	S153
	Learning Methods using inspection data	S81
Diretrizes educacionais	Set of Guidelines for Teaching OO Modelling Using Concept Mapping Techniques	S37
	Methodology for developing Sequence Diagrams	S53
	Principles for teaching object-oriented design	S117
	Design Pattern in software behavior models	S158
	minimalCD — A set of translation rules	S22
	Use Case Writing Guidelines	S38
	Guidelines for conducting training	S40
	A set of mapping rules between models	S42
	Best Practices for Teaching UML	S69
	Classification catalogue for examples	S160
	Tabular Analysis Method	S85
	Patterns of Cognitive Error	S97
	UML-Based Team Project	S118
SaUML	S52	
Abordagens Educacionais	Pantomime	S100, S110
	Blended Learning Approach	S80
	Antipatterns for addressing correctness problems in modeling	S03
	Educational approach based on peer reflection	S138
	Interactive lecture workshop	S140
	Reverse Engineering Techniques	S100
	Repository of UML Class Diagrams	S02
Design Instrucional	Format of a course	S33, S39, S59, S64, S89, S92, S94, S95, S98, S99, S111, S119, S121
	Elicitation Teaching for Object-Oriented Analysis and Design	S54
	Process of UML 2 Teaching	S66, S77
	Framework for student projects	S12
	Instructional Design of Exercise-Centric Teaching Materials on UML Modeling	S10
	Modeling Education Program for Novices	S23
	Case Method Approach	S43
UML modeling template	S48	

MAPEAMENTO ENTRE DIFICULDADES E TECNOLOGIAS IDENTIFICADAS NO MSL

Cat.	Dificuldades principais	Tecnologias propostas para minimizar as dificuldades		
		Ferramentas	Métodos, Diretrizes e Abordagens	Design Instrucional
Dificuldades relacionadas aos diagramas UML	<ul style="list-style-type: none"> • Dificuldade em compreender a sintaxe dos diagramas UML • Dificuldade em compreender a semântica dos diagramas UML • Dificuldade para compreender os relacionamentos entre os diagramas da UML • Dificuldade em definir o diagrama mais adequado para o projeto • Definição do nível de detalhe dos diagramas UML • Necessidade de conhecimento prévio de OO 	<p>Ferramentas Case:</p> <ul style="list-style-type: none"> • Papyrus (S08) • Rational Rose S169 • Vision (S169) <p>Sistemas de Avaliação dos diagramas:</p> <ul style="list-style-type: none"> • Method to detect learners' difficult points (S17) • Tool to detect learners' difficult points. (S17) • SmallTEamsHelper (S179) <p>Sistemas de tutoria de modelagem:</p> <ul style="list-style-type: none"> • diagnostic method + Diagram (S58) • TextOO (S72) • Ideogramic UML (S88) • TeDUB (S90) • QSEE-SuperLite (S101) • Agent-Oriented Collaborative Learning System (S154) • A Prototype Object-Oriented Design Learning Environment (OODLE) (S157) • Online Img2UML Repository (S25) 	<p>Métodos Educacionais:</p> <ul style="list-style-type: none"> • Problem-based learning (S116, S189) • Learning from Erroneous Examples (S180) • Method to detect learners' difficult points (S17) • PBL + LBL Double Track Teaching Method (S116) • Project-based learning with weekly quizzes, tests and active learning tasks (S137) <p>Diretrizes Educacionais:</p> <ul style="list-style-type: none"> • Principles for teaching object-oriented design (S117) • Best Practices for Teaching UML (S69) <p>Abordagens Educacionais:</p> <ul style="list-style-type: none"> • Pantomime (S100, S110) • Blended Learning Approach (S80) • Interactive lecture workshop (S140) • Reverse Engineering Techniques (S100) 	<ul style="list-style-type: none"> • Format of a course (S33, S59, S89, S98, S99) • Process of UML 2 Teaching (S66, S77) • Instructional Design of Exercise-Centric Teaching Materials on UML Modeling (S10) • Modeling Education Program for Novices (S23)
Dificuldades relacionadas ao diagrama de casos de uso	<ul style="list-style-type: none"> • Dificuldades relacionadas ao ator • Dificuldades relacionadas aos Casos de uso • Dificuldades relacionadas às relações entre casos de uso 	<p>Ferramentas Case:</p> <ul style="list-style-type: none"> • IDE for OOAD (S01) • Papyrus (S08) • UMLet (S86) • QuickUML (S96) <p>Sistemas de tutoria de modelagem:</p> <ul style="list-style-type: none"> • StudentUML (S05) • UML-IT (S06, S108) • SPACE-DESIGN (S13) • Tablet PCs e Web Conference Software (S63) • Prototype of adaptive online module based on the student's learning style categories (S142) • Online Learning System (S162, S166, S180) 	<p>Métodos Educacionais:</p> <ul style="list-style-type: none"> • Teaching Methods of Object-Oriented Approach in MIS Curriculum (S107) • Method based on think-pair-square (S16) • Rules-based pedagogical method for identifying defects in models (S49) • Learning Methods using inspection data (S81) <p>Diretrizes Educacionais:</p> <ul style="list-style-type: none"> • Use Case Writing Guidelines (S38) • Guidelines for conducting training (S40) • Best Practices for Teaching UML (S69) • UML-Based Team Project (S118) 	<ul style="list-style-type: none"> • Format of a course (S33, S64, S92, S94, S111) • Elicitation Teaching for Object-Oriented Analysis and Design (S54) • Framework for student projects (S12) • Case Method Approach (S43) • UML modeling template (S48)

<p>Dificuldades relacionadas ao diagrama de classes</p>	<ul style="list-style-type: none"> • Dificuldades relacionadas às classes • Dificuldades relacionadas aos atributos • dificuldades relacionadas às associações (relacionamentos) • Dificuldades relacionadas aos métodos 	<p>Ferramentas Case:</p> <ul style="list-style-type: none"> • IDE for OOAD (S01) • LogViz (S04) • UML editor WebUML (S07) • UMLet (S86) • Multi-Touch UML Editor (S135) • Spreadsheet-based modeling approach (S143) <p>Sistemas de Avaliação dos diagramas:</p> <ul style="list-style-type: none"> • Graph matching system for class diagram (S136) • A prototype for the automated assessment of UML class diagrams (S139) • Method to detect learners' difficult points (S17) • Approach and algorithms for automating the selection of the best sub-diagrams (S163) <p>Sistemas de tutoria de modelagem:</p> <ul style="list-style-type: none"> • StudentUML (S05) • UML-IT (S06, S108) • Intelligent Problem-Solving Environment for UML Class Diagrams (S28) • COLLECT-UML (S36, S61, S65, S76, S130) • Courseware for UML/SysML modelling (S44) • UML Checker (S50) • UML Tutor (S50) • PatternCoder (S55) • AUTO-COLLEAGUE (S56, S60, S129) • Diagram UML (S57) • diagnostic method + Diagram (S58) • TextOO (S72) • Ideogramic UML (S88) • Interactive Exercise (S103) • UMLGrader (S112) • CIMEL-ITS (S161) • Tool of Model-Driven Requirements Analysis Using Unified Modeling Language (S29) • MERODE (S122, S123, S125) • Geon (S84) • VisAr3D environment (S141) • Prototype of adaptive online module based on the student's learning style categories (S142) 	<p>Métodos Educacionais:</p> <ul style="list-style-type: none"> • Teaching Method for Conceptual Modeling Based on Error Analyses of Class Diagrams (S15) • Teaching Methods of Object-Oriented Approach in MIS Curriculum (S107) • Rules-based pedagogical method for identifying defects in models (S49) • Positive and Negative Examples (S148, S149) • Jigsaw Technique (S153) • Learning Methods using inspection data (S81) <p>Diretrizes Educacionais:</p> <ul style="list-style-type: none"> • Set of Guidelines for Teaching OO Modelling Using Concept Mapping Techniques (S37) • minimalCD — A set of translation rules (S22) • Guidelines for conducting training (S40) • Best Practices for Teaching UML (S69) • Classification catalogue for examples (S160) • Patterns of Cognitive Error (S97) • UML-Based Team Project (S118) <p>Abordagens Educacionais:</p> <ul style="list-style-type: none"> • Antipatterns for addressing correctness problems in modeling (S03) • Educational approach based on peer reflection (S138) • Interactive lecture workshop (S140) • Repository of UML Class Diagrams (S02) 	<ul style="list-style-type: none"> • Format of a course (S92, S94, S95 S111 S119, S121) • Elicitation Teaching for Object-Oriented Analysis and Design (S54) • Case Method Approach (S43)
---	--	--	--	--

Cat.	Dificuldades principais	Tecnologias propostas para minimizar as dificuldades		
		Ferramentas	Métodos, Diretrizes e Abordagens	Design Instrucional
		<ul style="list-style-type: none"> • Online Learning System (S162, S166, S180) • Tool for create the class diagram (S163) • Online Img2UML Repository (S25) 		
Dificuldades relacionadas ao diagrama de sequência	<ul style="list-style-type: none"> • Dificuldades relacionadas aos objetos • Dificuldades relacionadas às mensagens 	<p>Ferramentas Case:</p> <ul style="list-style-type: none"> • Papyrus (S08) • UMLet (S86) • Minimally Adequate Synthesizer (MAS) (S156) <p>Sistemas de tutoria de modelagem:</p> <ul style="list-style-type: none"> • StudentUML (S05) • Courseware for UML/SysML modelling (S44) • TextOO (S72) • Online Learning System (S162, S166, S180) 	<p>Métodos Educacionais:</p> <ul style="list-style-type: none"> • Teaching Methods of Object-Oriented Approach in MIS Curriculum (S107) • Rules-based pedagogical method for identifying defects in models (S49) <p>Diretrizes Educacionais:</p> <ul style="list-style-type: none"> • Set of Guidelines for Teaching OO Modelling Using Concept Mapping Techniques (S37) • Methodology for developing Sequence Diagrams (S53) • Design Pattern in software behavior models (S158) • Guidelines for conducting training (S40) • A set of mapping rules between models (S42) • Best Practices for Teaching UML (S69) • Tabular Analysis Method (S85) • UML-Based Team Project (S118) • SaUML (S52) <p>Abordagens Educacionais:</p> <ul style="list-style-type: none"> • Interactive lecture workshop (S140) 	<ul style="list-style-type: none"> • Format of a course (S64, S94, S95 S119, S121), • Elicitation Teaching for Object-Oriented Analysis and Design (S54) • Case Method Approach (S43)
Dificuldades relacionadas	<ul style="list-style-type: none"> • Dificuldade relacionada aos fluxos • Dificuldades no nó de união 	<p>Ferramentas Case:</p> <ul style="list-style-type: none"> • Papyrus (S08) <p>Sistemas de tutoria de modelagem:</p> <ul style="list-style-type: none"> • StudentUML (S05) • UML-IT (S06, S108) 	<p>Diretrizes Educacionais:</p> <ul style="list-style-type: none"> • Guidelines for conducting training (S40) • UML-Based Team Project (S118) 	<ul style="list-style-type: none"> • Format of a course (S33, S39, S94) • Elicitation Teaching for Object-Oriented Analysis and Design (S54)

OPEN REPOSITORY FOR TEACHING SOFTWARE MODELING APPLYING ACTIVE LEARNING STRATEGIES: MATERIAL DE APOIO

Este é o material de suporte que foi usado para desenvolver, avaliar e evoluir um repositório aberto para o ensino de modelagem de software, empregando estratégias de aprendizado ativas. Além disso, apresentamos a evolução feita da primeira versão para sua versão atual.

1. EVOLUÇÃO DO REPOSITÓRIO ABERTO

Para apoiar os instrutores na escolha de estratégias de ensino apropriadas, consideramos as estratégias identificadas e selecionadas em um Mapeamento Sistemático. Identificamos os métodos de ensino que alavancam qualquer estratégia de aprendizado ativo em seus antecedentes teóricos. Com base nisso, definimos a primeira versão do repositório. Todas as estratégias disponibilizadas no repositório são apresentadas em Tabela 4.

Tabela 4. Estratégias selecionadas para compor o OpenSMALS.

Nome das estratégias	Descrição das estratégias	Autores
<i>Learning based on similar systems</i>	O método Aprendizagem Baseada em Sistemas Familiares permite que os estudantes investiguem um sistema semelhante ao que se pretende modelar e, com isso, realizar a modelagem do novo sistema utilizando como base as características identificadas no sistema escolhido.	Stoecklin <i>et al.</i> (1998)
<i>Jigsaw</i>	O <i>Jigsaw</i> é uma técnica que consiste em dividir o material de aprendizagem em tarefas parciais. Cada estudante da equipe é como uma peça de quebra-cabeças que executará uma tarefa parcial, que ao final do processo será compartilhada com todos os outros membros da equipe.	Portilho e Campos (2009)
<i>Learning from Erroneous Examples</i>	O <i>Learning from Erroneous Examples</i> (ErrEx) é um método baseado em Aprendizagem Ativa que engaja os estudantes em um exercício de aprendizagem ativa, oferecendo a oportunidade para uma <i>learning experience</i> construtiva e colaborativa através dos exemplos errados. Este método instrui os estudantes a encontrar, explicar e corrigir os erros.	Silva <i>et al.</i> (2017a)
<i>Inspection-Based strategy</i>	O <i>Inspection-Based strategy</i> tem por objetivo apoiar os estudantes que estão aprendendo como modelar os diagramas corretamente. Para isso, os estudantes utilizam comentários de inspeção de diagramas previamente inspecionados por outros estudantes.	Kinjo <i>et al.</i> (2006)
<i>Multimodal Approach</i>	Esta é uma abordagem multimodal que visa fomentar as habilidades de modelagem dos estudantes que são relevantes para a indústria, combinando uma abordagem orientada por projeto com várias atividades que focam em princípios de aprendizagem ativa.	Thevathayan and Hamilton (2017)

Nome das estratégias	Descrição das estratégias	Autores
<i>Negative Examples</i>	O <i>Negative Examples</i> é um método que fornece diretrizes associadas a descrições e exemplos negativos de modelagem, para que estes sejam incorporados no ambiente de ensino de modelagem, fazendo com que os estudantes aprendam observando como se deve fazer uma modelagem de forma correta a partir de exemplos negativos trabalhados em sala de aula.	Bolloju <i>et al.</i> (2011)
<i>PBL + LBL Double Track Teaching Method</i>	Esse método de ensino baseia-se na <i>Lecture based learning</i> (LBL) e <i>Problem based learning</i> (PBL). Neste método de ensino o professor ensina a base teórica relacionada ao problema real e combina enquanto os estudantes estão trabalhando para a resolução de problemas real.	Wang <i>et al.</i> (2013)
<i>Positive Examples</i>	O <i>Positive Examples</i> é um método que fornece diretrizes associadas a descrições e exemplos positivos de modelagem, para que estes sejam incorporados no ambiente de ensino de modelagem, fazendo com que os estudantes aprendam observando como se deve fazer uma modelagem de forma correta.	Bolloju <i>et al.</i> (2011)
<i>Problem Based Learning Adapted</i>	O <i>PBL Adapted</i> é um método pedagógico que enfatiza o papel de um problema, no qual os estudantes são responsáveis pela sua aprendizagem. Este método ajuda os estudantes a desenvolver estratégias e construir conhecimento. Formação de equipe e distribuição de papéis são essenciais para o uso de PBL.	Silva <i>et al.</i> (2017a)
Think-Pair-Square	O Método <i>Think-Pair-Square</i> foi concebido para promover aprendizagens, discussões ativas e para auxiliar na resolução de problemas de forma cooperativa entre os estudantes. O método Think-Pair-Square é considerado idealmente quando se deseja que os estudantes obtenham experiências de trabalho colaborativo.	Scanniello e Erra (2014)

Após identificar as estratégias, organizamos o conhecimento sobre cada uma das estratégias identificadas. Para isso, primeiro definimos o domínio e o escopo que devem ser abordados pelo conhecimento gerado. O domínio é a representação semântica e a formalização de estratégias de ensino baseadas em princípios ativos de aprendizagem **Erro! Fonte de referência não encontrada.** O escopo está relacionado ao suporte dado a essas estratégias, por meio do conhecimento organizado e da representação semanticamente, facilitando sua difusão e uso. Em seguida, realizamos uma especificação em linguagem natural, na qual organizamos as informações sobre as estratégias, através dos artigos que relataram o uso das estratégias. Em seguida, estruturamos as informações coletadas em um modelo conceitual descrito como um diagrama de classes (ver Figura 2).

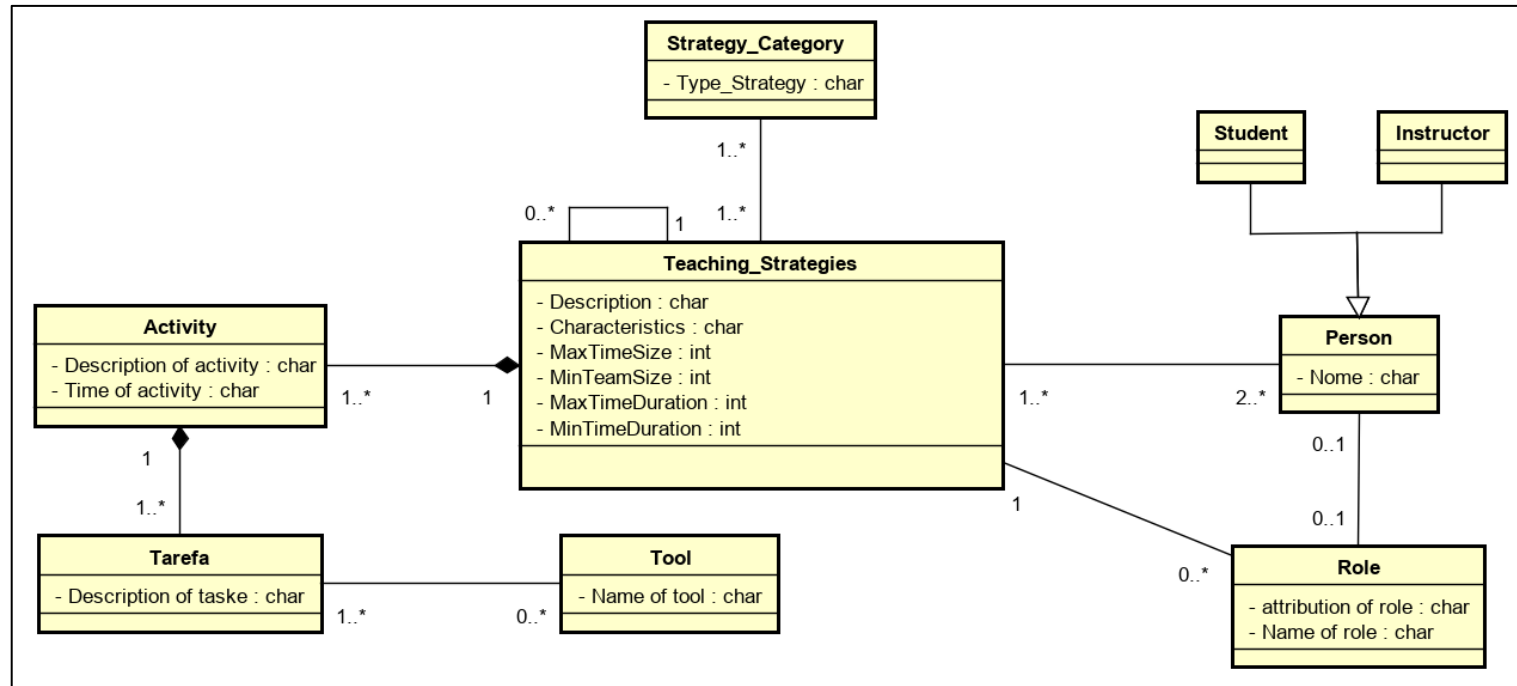


Figura 2. Modelo conceitual elaborado com as informações coletadas das estratégias.

A classe *Teaching_Strategies* representa o conjunto de estratégias de aprendizado ativas. Uma estratégia pode ser usada em conjunto com outra, complementando ou melhorando seus resultados. Isso é apresentado pelo auto-relacionamento na classe *Teaching_Strategies*. A classe *Teaching_Strategies* está associada à classe *Strategy_Category* que, conforme detalhado por Barkley *et al.* pode ser um desses tipos: organizadores de discussões, jogos, redação, ensino recíproco, solução de problemas e informações gráficas. Além disso, uma estratégia consiste em um conjunto de atividades (classe *Activity*). Cada atividade tem uma descrição do que deve ser realizado e seus objetivos, além da duração esperada. Uma atividade é composta por uma ou mais tarefas (classe *Tarefa*). Uma tarefa é descrita por seus procedimentos. Algumas dessas tarefas podem exigir o uso de uma ferramenta de suporte para sua execução; por exemplo, uma tarefa pode estar discutindo um determinado assunto e o instrutor pode definir que deve ser

executada uma ferramenta de comunicação síncrona. Os participantes das estratégias estão representados na classe Pessoa e podem ser de dois tipos: Instrutores e Alunos. Essas pessoas podem desempenhar papéis diferentes (classe de função) durante uma atividade. Por exemplo, na estratégia com base em inspeção, os participantes podem executar as funções de modelagem e / ou inspetor. Como forma de explicar o modelo conceitual proposto, desenvolvemos um questionário baseado no conhecimento das estratégias representadas no modelo conceitual. Com base nas respostas fornecidas pelos instrutores no questionário, o questionário do repositório recomenda um conjunto de estratégias de acordo com as necessidades do instrutor interessado em aplicá-las. Para isso, construímos uma árvore de decisão com três níveis. Cada nível tem perguntas que orientam o instrutor a escolher uma estratégia apropriada: Nível 01 - Escolha dos diagramas (que diagramas você deseja ensinar?) Nível 02 - Realização de atividades (Como você deseja que as atividades sejam executadas? ou ambos?) Nível 03 - Suporte de ferramentas (você deseja suporte de ferramentas?). Figura 3 (item a) apresenta uma parte da árvore de decisão criada. O questionário do repositório mostra ao instrutor o nome da estratégia (Figura 3.b item A), uma breve descrição da estratégia (Figura 3b item B) e um link onde o instrutor pode clicar para obter mais informações (Figura 3.b item C).

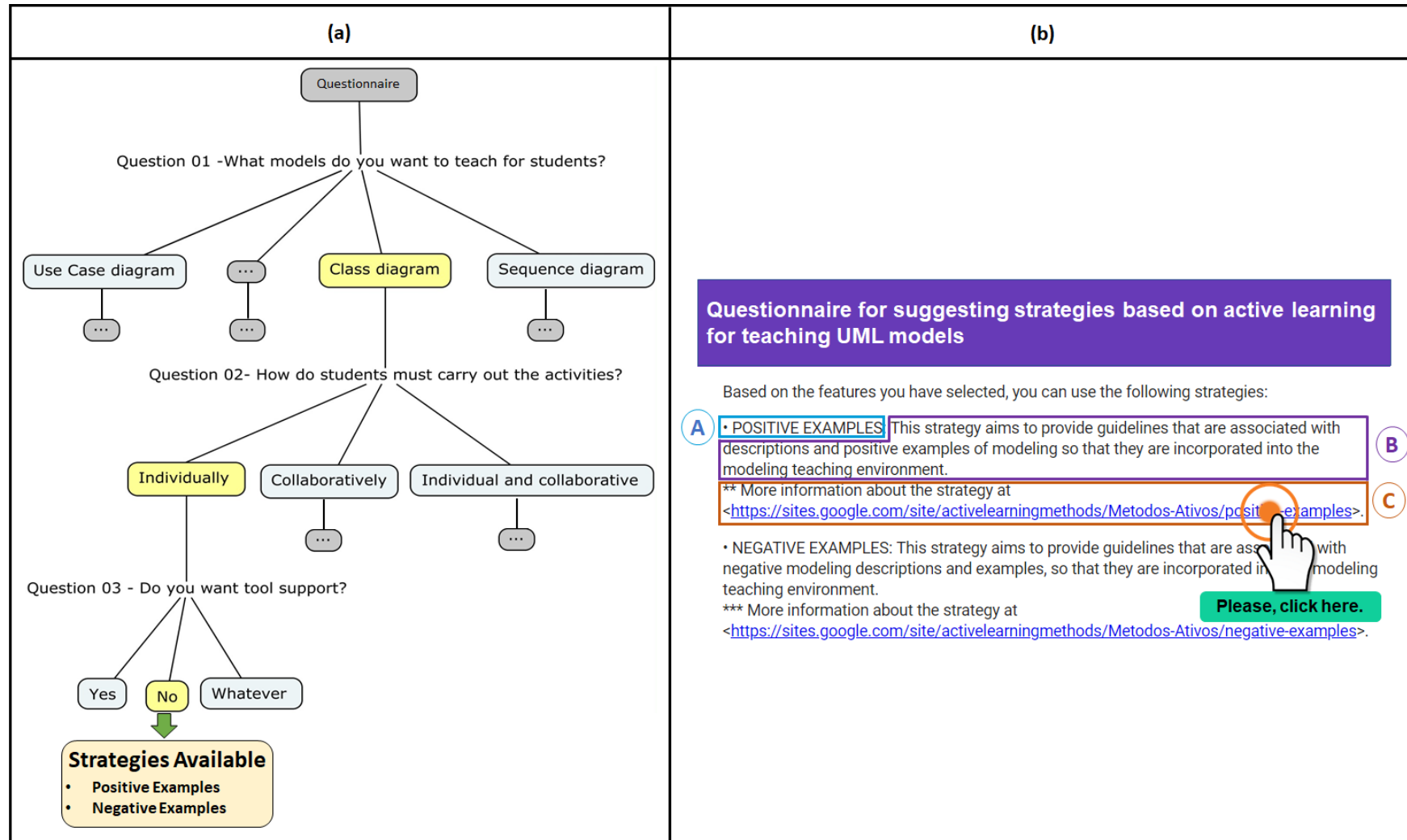


Figura 3. Exemplo de grupos de perguntas da primeira versão da árvore de decisão para o questionário e estratégia sugerida pelo questionário.

Todas as perguntas e respostas possíveis para essa árvore de decisão podem ser vistas na Figura 4

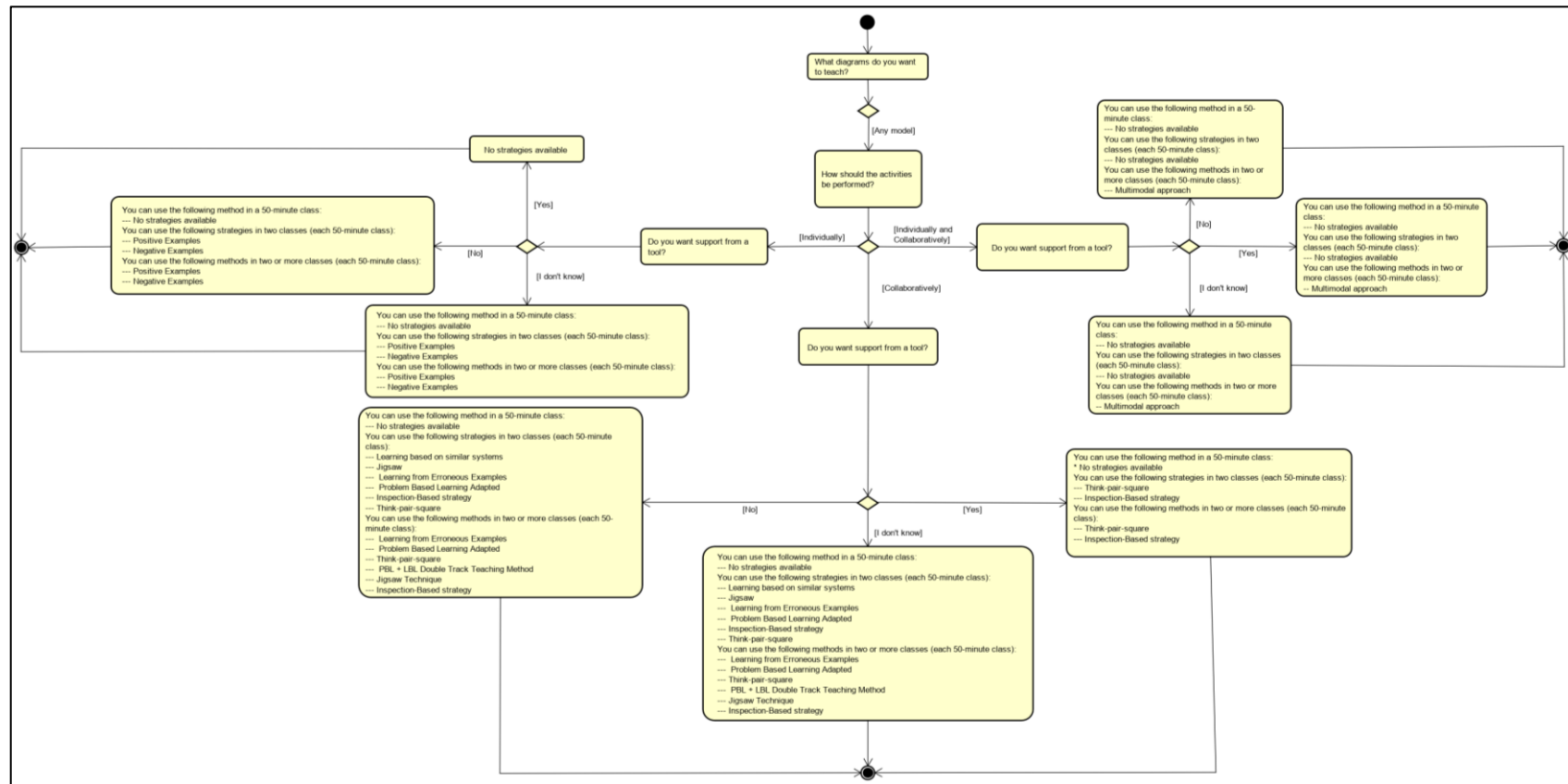


Figura 4. Grupos de perguntas da primeira versão da árvore de decisão.

No portal da Web (Figura 5), agrupamos algumas informações importantes para que o instrutor possa empregar a estratégia de aprendizado ativa escolhida: o nome da estratégia (Figura 5 elemento A); uma breve descrição da estratégia (Figura 5 elemento B); o modelo no qual a estratégia pode ser

empregada (Figura 5 elemento C); as principais características das estratégias, bem como os principais benefícios que os alunos obterão após o uso da estratégia (Figura 5 elemento D); o tipo de categoria a que a estratégia pertence (Figura 5 elemento E); o tempo mínimo necessário para a aplicação da estratégia (Figura 5 elemento F); o número apropriado de alunos necessários para a formação dos grupos, se necessário (Figura 5 elemento G); as etapas de uso que podem ser seguidas pelo instrutor ((Figura 5 elemento H); o nome da ferramenta, se houver, e se o instrutor desejar usá-la com os alunos (Figura 5 elemento I); exemplos de uso anterior da estratégia (Figura 5 elemento J); e, finalmente, as referências de onde as informações foram coletadas (Figura 5 elemento K).

Home Questionnaire Teaching Strategies based on Active Learning for teaching of UML Researchers' Contact

Navigation

Home
Questionnaire
Teaching Strategies based on Active Learning for teaching of UML models
Think-Pair Square
Learning based on similar systems
Jigsaw
Learning from Erroneous Example
Inspection-Based strategy
Multimodal Approach
Negative Examples
PBL + LBL Double Track Teaching Strategy
Positive Examples
Problem-based Learning Adapted
Researchers' Contact

Teaching Strategies based on Active Learning for teaching of UML Models

Negative Examples ^(A)

Strategy Description: This strategy provides problems that are associated with negative modeling descriptions and examples so that they are incorporated into the modeling teaching environment. ^(B)

UML model that strategy can be used: The strategy was used to teach the class model, but it can be adapted to teach any UML model, as long as the trainer identifies the main types of defects for the model one wants to teach. ^(C)

Characteristics of the strategy: ^(D)

- It allows students to understand the main difficulties when performing the modeling.
- It allows students to check for defects committed by other students in the models.
- It allows the students to perceive the defect and not to realize it again.

Strategy type: Strategies for problem-solving ^(E)

Time (Minimum) required for application of the strategy: At least two classes are required, each with 50 minutes. ^(F)

The number of students: The modeling activity of this strategy is preferably performed individually. ^(G)

Steps of using this strategy: ^(H)

- Step 01:** The instructor presents the concepts about the model during the lesson and then provides a scenario for the students to do the modeling.
- Step 02:** Students construct a model based on the scenario delivered by the instructor.
- Step 03:** The instructor performs training on the difficulties that are most common during the modeling of the requested artifact.
- Step 04:** The instructor gives the students a set of models with the main defects marked.
- Step 05:** The instructor asks the students to model the new diagram, based on training and a set of defective models.
- Step 06:** Students perform the correction of the elaborated models.

Tool Support: Does not have tool support. However, Bolloju et al. (2011) present a set of Negative Examples related to the class diagram (see in the annex below). ^(I)

Papers with examples of use this strategy ^(J)

- Bolloju, N., Schneider, C., Vogel, D. Asymmetrical effects of using positive and negative examples on object modeling. In Information Systems Development 2011 (pp. 85-96). Springer, New York, NY.

References: ^(K)
Bolloju N, Schneider C, Vogel D. Asymmetrical effects of using positive and negative examples on object modeling. In Information Systems Development 2011 (pp. 85-96). Springer, New York, NY.

Figura 5. Portal da Web com informações sobre estratégias.

No estudo piloto, uma das principais queixas do instrutor estava relacionada ao questionário para escolha das estratégias apropriadas. Segundo o instrutor I01, as perguntas eram muito gerais e não estavam de acordo com o contexto de ensino. Diante disso, revisitamos o modelo conceitual e analisamos novamente as informações coletadas para cada estratégia. Assim, uma nova versão da árvore de decisão foi proposta. Na Tabela 5, apresentamos todas as perguntas e respostas possíveis da nova árvore de decisão para a escolha de estratégias de aprendizado ativas.

Tabela 5. Grupos de perguntas da versão final da árvore de decisão do questionário.

<p>Questão: Em que momento você deseja aplicar a estratégia?</p>	<p>Resposta: Início da aprendizagem (os estudantes não aprenderam o conceito sobre o diagrama)</p>	<p>Questão: Em quantas aulas você gostaria de utilizar a estratégia?</p>	<p>Durante toda a disciplina</p>	<p>Questão: Com relação ao processo de ensino dos conceitos sobre os modelos, você gostaria de atuar como?</p>	<p>Resposta: Atuaria apenas como um mediador para tirar dúvidas dos estudantes</p>	<p>Questão: Qual o contexto do problema que você gostaria de utilizar com a estratégia?</p>	<p>Resposta: Um problema desenvolvido por mim mesmo</p>		<p>Jigsaw</p>					
							<p>Resposta: Gostaria de utilizar problemas reais da indústria</p>	<p>Você gostaria que cada estudante aprenda um diagrama específico, depois compartilhe o conhecimento adquirido com os demais colegas?</p>	<p>Resposta: Sim</p>	<p>Jigsaw</p>				
									<p>Resposta: Não</p>	<p>Problem-Based Learning Adapted</p>				
									<p>Resposta: Não importa</p>	<p>Jigsaw and Problem-Based Learning Adapted</p>				
							<p>Resposta: Não sei bem qual tipo de problema utilizar</p>	<p>Você gostaria que cada estudante aprenda um diagrama específico, depois compartilhe o conhecimento adquirido com os demais colegas?</p>	<p>Resposta: Sim</p>	<p>Jigsaw</p>				
									<p>Resposta: Não</p>	<p>Problem-Based Learning Adapted</p>				
									<p>Resposta: Não importa</p>	<p>Jigsaw and Problem-Based Learning Adapted</p>				
							<p>Realizaria uma apresentação para os estudantes sobre os conceitos básicos dos diagramas</p>	<p>Questão: Durante as apresentações sobre os conceitos, você deseja envolver os estudantes em outras atividades?</p>	<p>Resposta: Sim</p>	<p>A multimodel Approach</p>				
									<p>Resposta: Não</p>	<p>PBL+LBL double track teaching method</p>				
									<p>Resposta: Não sei ainda</p>	<p>A multimodel Approach and/or PBL+LBL double track teaching method</p>				
							<p>Em algumas aulas (máximo três aulas)</p>	<p>Questão: Com relação aos diagramas que deseja ensinar?</p>	<p>Você deseja ensinar vários diagramas de uma vez</p>	<p>Questão: Com relação ao processo de ensino dos conceitos sobre o diagrama</p>	<p>Resposta: Atuaria apenas como um mediador para tirar dúvidas dos estudantes</p>		<p>Jigsaw</p>	
											<p>Resposta: Realizaria uma apresentação para os estudantes sobre os conceitos básicos dos diagramas</p>		<p>A multimodel Approach</p>	
<p>Resposta: Não sei ainda</p>		<p>Jigsaw and a multimodel Approach</p>												
<p>Não me decidi ainda</p>	<p>Questão: Com relação ao</p>	<p>Resposta: Atuaria apenas como um mediador para tirar</p>	<p>Questão: Qual o contexto do problema que você gostaria de</p>	<p>Resposta: Um problema desenvolvido por mim mesmo</p>		<p>Jigsaw</p>								
				<p>Você gostaria que cada estudante aprenda um diagrama específico,</p>	<p>Sim</p>	<p>Jigsaw</p>								
					<p>Não</p>	<p>Problem-Based Learning Adapted</p>								

<p>Resposta: Após a exposição inicial dos conceitos sobre o diagrama (os estudantes já realizam alguns exercícios sobre a modelagem)</p> <p>Com relação aos estudantes matriculados em sua disciplina, você considera que:</p> <p>Resposta: Não tenho uma opinião sobre esse ponto</p> <p>Questão: Sobre a aplicação da estratégia, você espera que:</p> <p>Resposta: Os estudantes ainda necessitam de alguma ajuda para elaborar os diagramas</p> <p>Questão: Durante o processo de aprendizagem, você gostaria que:</p> <p>Resposta: Disponibilizasse guidelines com exemplos positivos que direcionassem os estudantes durante a elaboração dos diagramas</p> <p>Resposta: Disponibilizasse guidelines com exemplos negativos que direcionassem os estudantes durante a elaboração dos diagramas</p> <p>Resposta: Não me decidi</p> <p>Resposta: Os estudantes já sabem elaborar os diagramas sem necessitar de alguma ajuda</p> <p>Questão: Com relação aos diagramas que serão elaborados pelos estudantes, você gostaria que:</p> <p>Os estudantes realizassem uma avaliação nos diagramas que serão elaborados</p> <p>Os estudantes elaborassem os modelos a partir de um determinado problema</p> <p>Resposta: Os estudantes sejam guiados durante o processo de aprendizagem</p> <p>Resposta: Os estudantes sejam capazes de realizar a modelagem de forma autonoma</p> <p>Resposta: Os estudantes realizem uma avaliação</p> <p>Questão: Quanto à avaliação,</p> <p>Resposta: Os estudantes realizem a modelagem a partir de um problema toy, ou seja, retirado de um livro.</p>	<p>dúvidas dos estudantes</p> <p>utilizar com a estratégia?</p> <p>Resposta: Realizaria uma apresentação para os estudantes sobre os conceitos básicos dos diagramas</p> <p>Questão: Durante as apresentações sobre os conceitos, você deseja envolver os estudantes em outras atividades?</p> <p>Resposta: Sim</p> <p>Resposta: Não</p> <p>Resposta: Não sei ainda</p>	<p>depois compartilhe o conhecimento adquirido com os demais colegas?</p> <p>Você gostaria que cada estudante aprenda um diagrama específico, depois compartilhe o conhecimento adquirido com os demais colegas?</p> <p>A multimodel Approach</p> <p>PBL+LBL double track teaching method</p> <p>A multimodel Approach and/or PBL+LBL double track teaching method</p>	<p>Não importa</p> <p>Sim</p> <p>Não</p> <p>Não importa</p>	<p>Jigsaw and Problem-Based Learning Adapated</p> <p>Jigsaw</p> <p>Problem-Based Learning Adapated</p> <p>Jigsaw and Problem-Based Learning Adapated</p>						
					<p>Resposta: Disponibilizasse guidelines com exemplos positivos que direcionassem os estudantes durante a elaboração dos diagramas</p> <p>Positive Examples</p>	<p>Resposta: Disponibilizasse guidelines com exemplos negativos que direcionassem os estudantes durante a elaboração dos diagramas</p> <p>Positive Examples</p>	<p>Resposta: Não me decidi</p> <p>Positive Examples and/or Negative Examples</p>			
								<p>Os diagramas elaborados fossem avaliados por outros estudantes</p> <p>Método de Aprendizagem utilizando dados de inspeções</p>	<p>Os diagramas elaborados fossem elaborados pelos próprios estudantes</p> <p>Learning form Erroneous Examples</p>	
										<p>Não sei ainda</p> <p>Método de Aprendizagem utilizando dados de inspeções and Learning form Erroneous Examples</p>
					<p>Os estudantes realizassem a modelagem a partir de um problema toy, ou seja, retirado de um livro.</p> <p>Think-Pair-Square</p>	<p>Os estudantes realizassem a modelagem a partir de um sistema já existente no mercado</p> <p>Aprendizagem baseada em Sistemas Similares</p>	<p>Não sei ainda</p> <p>Think-Pair-Square and Aprendizagem baseada em Sistemas Similares</p>			
								<p>Resposta: Exemplos positivos, para que analisem e melhorem o diagrama seguindo as melhores práticas de modelagem</p> <p>Positive Examples</p>	<p>Resposta: Negative Examples, para que os estudantes percebam os principais problemas que ocorrem durante a modelagem</p> <p>Negative Examples</p>	<p>Resposta: Não me decidi ainda</p> <p>Positive Examples and/or Negative Examples</p>
					<p>Resposta: Os estudantes realizem a modelagem a partir de um problema toy, ou seja, retirado de um livro.</p> <p>Think-Pair-Square</p>	<p>Os diagramas fossem avaliados por outros estudantes</p> <p>Método de Aprendizagem utilizando dados de inspeções and Learning form Erroneous Examples</p>				

									Os estudantes realizassem a modelagem a partir de um sistema já existente no mercado	Aprendizagem baseada em Sistemas Similares
									Não sei ainda	Think-Pair-Square and Aprendizagem baseada em Sistemas Similares

Nesta nova versão do questionário, reagrupamos as perguntas de acordo com o momento que os instrutores esperam usar a estratégia: no início do aprendizado e depois de ensinar os conceitos sobre os modelos. No primeiro momento, os instrutores são direcionados para estratégias que podem ser usadas desde o início do aprendizado dos alunos, para que os alunos sejam responsáveis por seu próprio aprendizado. Nesse momento, os instrutores agem apenas como mediadores, ou seja, orientam os alunos enquanto usam essas estratégias. No segundo momento, os instrutores podem escolher estratégias que podem ser aplicadas após apresentar os conceitos básicos sobre o modelo para os alunos. Observamos também que as informações disponíveis no portal são suficientes para o instrutor aplicar as estratégias. Com base na sugestão do professor, aplicamos questionários de pós-modelagem com o objetivo de avaliar a experiência de aprendizado dos alunos. Em seguida, adicionamos no repositório (v2) as percepções dos alunos sobre estratégias (Figura 6). Isso pode ajudar os próximos instrutores a decidir uma das estratégias. Por exemplo, o instrutor respondeu ao questionário e a estratégia sugerida foi Exemplos negativos. No entanto, ao visualizar a experiência de aprendizado de outros alunos sobre essa estratégia, o instrutor pode decidir usar ou não essa estratégia.

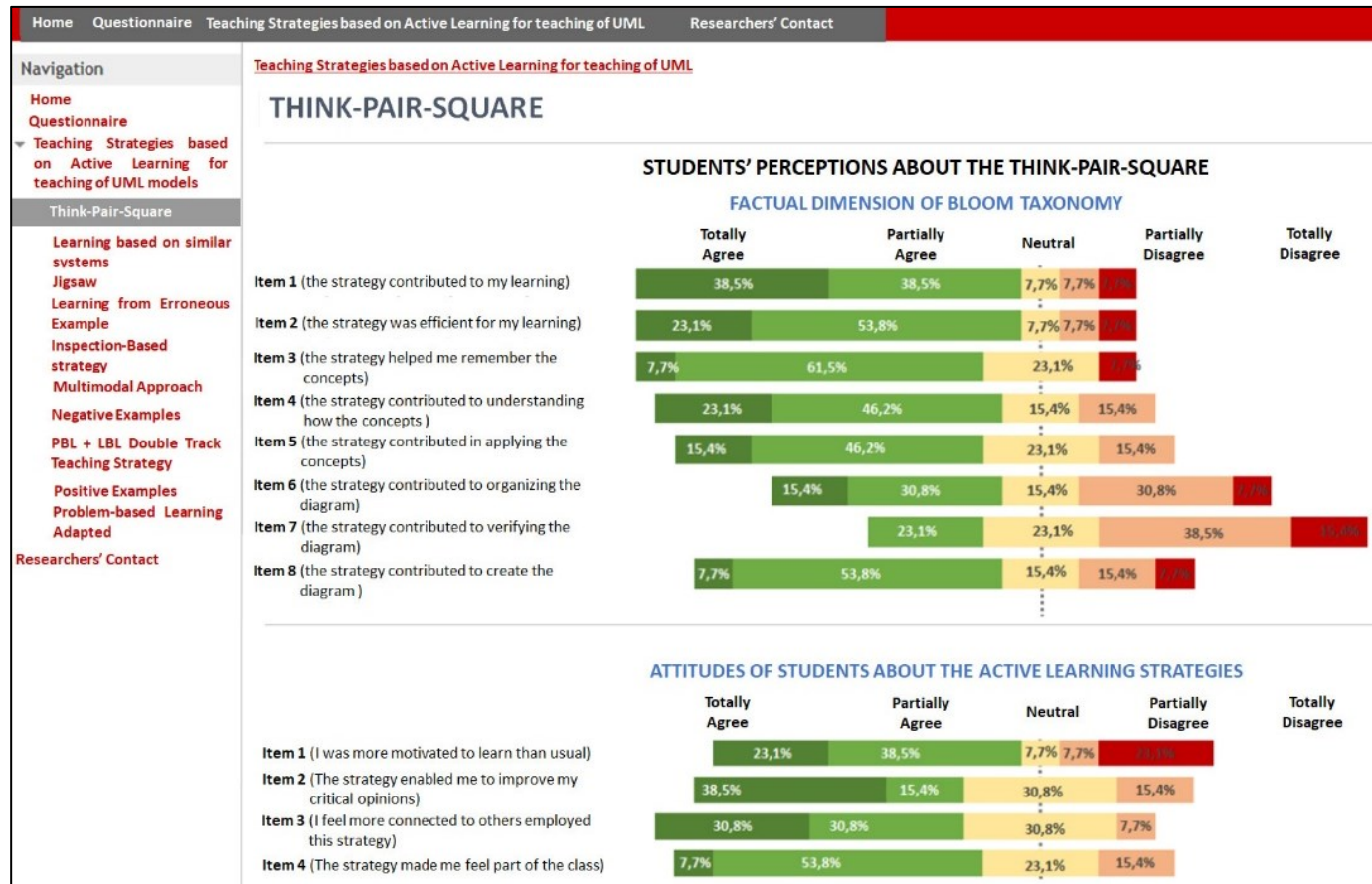


Figura 6. Exemplo de percepção dos alunos sobre a estratégia (Think-pair-square) na segunda versão do repositório.

O primeiro estudo empírico (ciclo um da avaliação do DSR) nos forneceu evidências para fazer melhorias que abordassem as dificuldades identificadas no uso do repositório. Nesse momento, desenvolvemos o portal do repositório (Figura 7) e decidimos transformar o repositório em um repositório aberto, no qual todos os artefatos necessários para implementar as estratégias são disponibilizados aos instrutores.

Navigation
Home
Strategies classified by moments of learning
Choose the most appropriate strategy here
Choose your scenario here
Artifacts to help evaluate strategies
Teaching Methods Based on Active Learning Strategies
Learning based on similar systems
Jigsaw
Learning from Erroneous Examples
Método de Aprendizagem utilizando Dados de Inspeções
Multimodal Approach
Negative Examples
PBL + LBL Double Track Teaching Method
Positive Examples
Problem Based Learning (PBL) Adapted
Think-Pair-Square

[Teaching Methods Based on Active Learning Strategies >](#)

Negative Examples

What is this strategy?



Negative Examples is a teaching method based on an active learning strategy that provides problems that are associated with negative modeling descriptions and examples so that they are incorporated into the modeling teaching environment.

What are the main gains of the strategy for students?



This method allows students to:

- Understand the main difficulties other students had when modeling a particular model.
- Check from the examples worked in class, which are the primary defects that the other students committed to developing a model.
- It allows students to realize the defect and not realize it again.

What are the prerequisites for using this strategy?



- Students need to know the model notation and have previously solved modeling exercises.
- If the instructor wants to use real scenarios, the time for modeling can vary from one to two classes, each class with 1h40min.

Strategy Classification: Strategy for problem-solving.

References:

Bolloju N, Schneider C, Vogel D. Asymmetrical effects of using positive and negative examples on object modeling. In Information Systems Development 2011 (pp. 85-96). Springer, New York, NY.

What are the steps to applying the Negative Examples Strategy?



The strategy has a set of steps that can be followed:

- **Step 01:** The instructor introduces the concepts about the model during class and then provides a scenario for students to perform the modeling.
- **Step 02:** Students build a model based on the scenario delivered by the instructor.
- **Step 03:** The instructor conducts training on the difficulties that are most common during modeling of the requested artifact (below are some defects about some models to assist instructors).
- **Step 04:** The instructor gives the students a set of models with the primary defects marked.
- **Step 05:** The instructor asks students to model a new diagram based on training and a

What is the time for applying the strategy?



To apply the strategy, a class of 1h40min is required.

Come check out some assessments made with this strategy?



[Click here](#) to see the evaluations performed with this strategy.

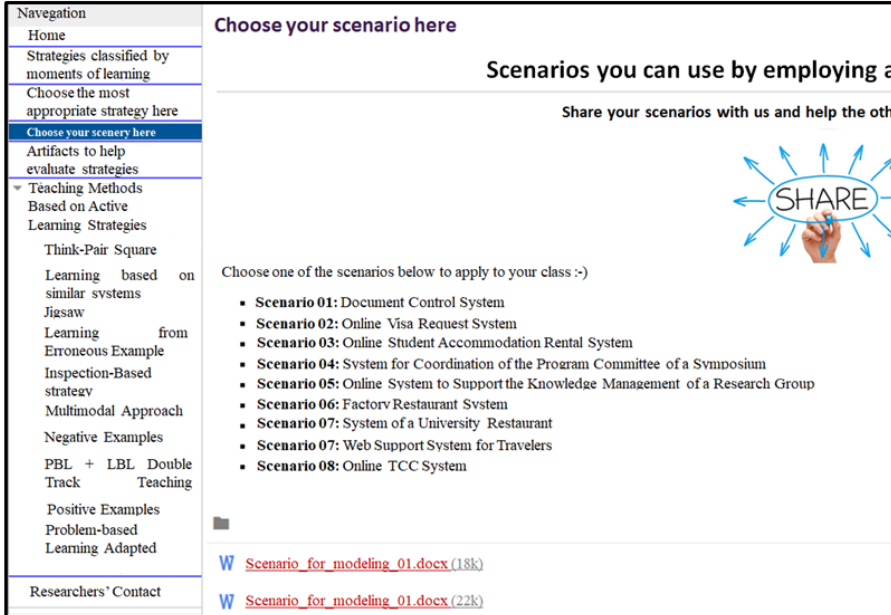
Have you used this method? How about sharing with us what was the students' perception of the method? Just [Click Here](#).

Figura 7. Uma nova versão do portal da web (v3).

Os artefatos fornecidos aos instrutores são:

- **Cenários de modelagem:** se os instrutores não tiverem cenários disponíveis para aplicar estratégias, eles podem usar alguns dos cenários fornecidos pelo portal (Figura 8.a);
- **Questionário de avaliação e uma planilha de coleta de dados:** se os instrutores desejam avaliar a experiência de aprendizado dos alunos e desejam apresentar esses resultados aos alunos para refletir sobre seu aprendizado (Figura 8.a).
- **Quadro do Grupo Focal:** se os instrutores desejarem reunir dados qualitativos, fornecemos um quadro de grupo de foco e instruções sobre como conduzi-lo com os estudantes (Figura 8.b).

(a) modeling scenarios



Choose your scenario here

Scenarios you can use by employing a

Share your scenarios with us and help the oth

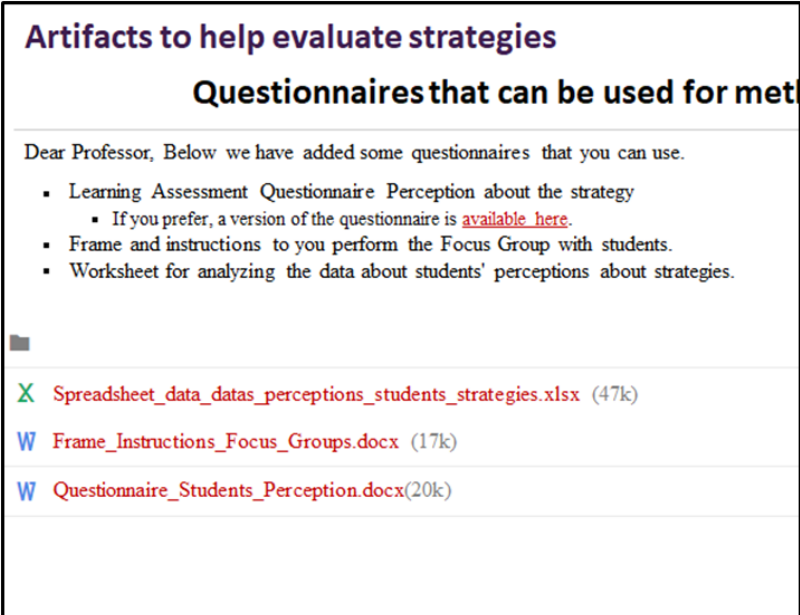
Choose one of the scenarios below to apply to your class :-)

- **Scenario 01:** Document Control System
- **Scenario 02:** Online Visa Request System
- **Scenario 03:** Online Student Accommodation Rental System
- **Scenario 04:** System for Coordination of the Program Committee of a Symposium
- **Scenario 05:** Online System to Support the Knowledge Management of a Research Group
- **Scenario 06:** Factory Restaurant System
- **Scenario 07:** System of a University Restaurant
- **Scenario 07:** Web Support System for Travelers
- **Scenario 08:** Online TCC System

[Scenario_for_modeling_01.docx \(18k\)](#)

[Scenario_for_modeling_01.docx \(22k\)](#)

(b) artifacts to help evaluate strategies



Artifacts to help evaluate strategies

Questionnaires that can be used for met

Dear Professor, Below we have added some questionnaires that you can use.

- Learning Assessment Questionnaire Perception about the strategy
 - If you prefer, a version of the questionnaire is [available here](#).
- Frame and instructions to you perform the Focus Group with students.
- Worksheet for analyzing the data about students' perceptions about strategies.

[Spreadsheet_data_datas_perceptions_students_strategies.xlsx \(47k\)](#)

[Frame_Instructions_Focus_Groups.docx \(17k\)](#)

[Questionnaire_Students_Perception.docx\(20k\)](#)

Figura 8. Páginas da Web com artefatos que os instrutores podem empregar em seus cursos.

No segundo estudo empírico, os instrutores relataram que o repositório não é explícito no esforço que o instrutor pode ter para aplicar essas estratégias. Dessa forma, conversamos com cada um dos instrutores que aplicaram as estratégias e perguntamos quanto esforço, medido em horas, eles

gastaram para preparar todo o material para a aula (Figura 9). Em seguida, para cada uma das estratégias utilizadas, adicionamos a seção "Quanto tempo para implementar a estratégia", adicionamos informações com o tempo necessário para preparar os materiais.

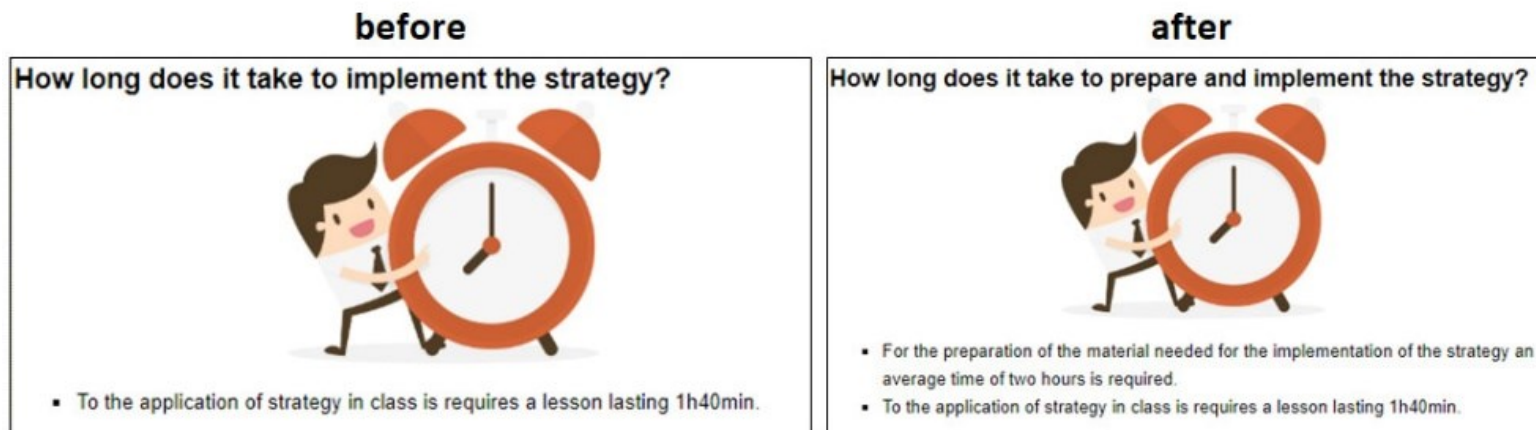


Figura 9. Alterações feitas para adicionar tempo e esforço para preparar as estratégias.

ESTRATÉGIAS DE ENSINO BASEADAS EM PRÍNCIPIOS DE APRENDIZAGEM ATIVA VISANDO O ENSINO DE DIAGRAMAS DA UML

A seguir será descrita a características das tecnologias que foram utilizadas para compor o OpenSMALS

TECNOLOGIA E ID	CARACTERÍSTICAS DAS TECNOLOGIAS	
TEACHING METHODS FOR CONCEPTUAL MODELING BASED ON ERROR ANALYSES OF CLASS DIAGRAMS - S15	Descrição	O método visa auxiliar os estudantes novatos no ensino de diagrama de classes a partir dos erros típicos que outros estudantes também novatos realizaram.
	Modelo a ser ensinado	Diagrama de classes, pois são definidos critérios para o ensino destes diagramas
	Tipo de Método	Resolução de problemas
	Papel dos estudantes	Não há papeis para os estudantes
	Tempo Máximo	90 minutos
	Tempo Mínimo	Não informado.
	Formação de Equipes	Individualmente
	Apoio Ferramental	Não
	Atividades	<ol style="list-style-type: none"> 1. Professor categorizar os erros em sintáticos, semânticos e pragmáticos. 2. Entregar os tipos de erros para os estudantes. 3. Estudantes realizam a modelagem
A METHOD BASED ON THINK-PAIR-SQUARE – S16	Descrição	O método tem como objetivo partir do princípio do trabalho individual, para o em pares e, por fim, em grupo
	Modelo a ser ensinado	Diagrama de casos de uso, mas pode ser adaptado para outros
	Tipo de Método	Discussão
	Papel dos estudantes	Não há papeis para os estudantes
	Tempo Máximo	348 minutos
	Tempo Mínimo	239 minutos
	Formação de Equipes	No mínimo 4 integrantes
	Apoio Ferramental	Há a necessidade de um ambiente integrado para realizar a comunicação e obter uma familiaridade entre os membros do time.
	Atividades	<ol style="list-style-type: none"> 1. Uso de um questionário para balancear os grupos

TECNOLOGIA E ID	CARACTERÍSTICAS DAS TECNOLOGIAS	
		2. Think - É realizado individualmente. Cada membro da equipe trabalha para produzir um ou mais diagramas de casos de uso. 3. Pair - É executado em par. Os membros da equipe trabalham em pares nos modelos que identificaram individualmente. 4. Square- Todos os membros da equipe juntos exploram o modelo de cada par para produzir um modelo comum e mais abrangente. As equipes podem ser compostas de qualquer número par maior ou igual a quatro.
A METHOD TO DETECT LEARNERS’DIFFICULT POINTS IN OBJECT MODELING EXERCISES – S17	Descrição	O método analisa o processo de criação do diagrama da classe dos alunos e detecta os padrões comportamentais dos alunos que aparecem quando os alunos enfrentam pontos difíceis.
	Modelo a ser ensinado	Diagrama de classes, pois tem critérios apenas para este tipo de diagrama
	Tipo de Método	Resolução de problemas
	Papel dos estudantes	Não há papeis para os estudantes
	Tempo Máximo	Não informado
	Tempo Mínimo	Não informado
	Formação de Equipes	Individualmente
	Apoio Ferramental	Se o professor quiser, ele pode utilizar a ferramenta para gerar a identificação dos defeitos de forma automática
Atividades	1. Um professor explica a técnica de modelagem de objetos e dá uma tarefa. 2. Os alunos trabalham nela usando nossa ferramenta e enviam o artefato que eles fizeram. 3. O professor procura uma lista de artefatos submetidos. Neste momento, a ferramenta detecta os aprendizes cujos artefatos têm métricas outras e os enfatiza visualmente, como mudar suas cores. 4. O professor remete o processo de criação do artefato dos alunos que foi enfatizado por ferramenta e reconhece os pontos difíceis dos alunos. A ferramenta ajuda isso, fornecendo um visualizador de processos integrado com um visualizador de métricas. 5. O professor analisa todos os artefatos submetidos. A ferramenta mostra o resultado da análise, como os pontos difíceis dos alunos, as métricas calculadas e assim por diante. 6. O professor fornece aos alunos um feedback detalhado sobre o processo de criação com base nas informações descritas na etapa anterior, além do feedback normal baseado no artefato. Isso pode ser efetivo para ativar a reflexão dos aprendizes. 7. Os alunos reproduzem seu processo e refletem sua aprendizagem com base no feedback.	
MÉTODOS DE APRENDIZAGEM	Descrição	O método tem por objetivo auxiliar na leitura e criação de diagramas UML, baseado nos comentários realizados nos diagramas da UML
	Modelo a ser ensinado	Diagrama de classes, mas pode ser adaptado

TECNOLOGIA E ID	CARACTERÍSTICAS DAS TECNOLOGIAS	
UTILIZANDO DADOS DE INSPEÇÕES – S81	Tipo de Método	Discussão
	Papel dos estudantes	Há dois papéis: o inspetor e o modelador.
	Tempo Máximo	Não informado. Mas pode ser empregado em uma aula de 90min.
	Tempo Mínimo	Não informado
	Formação de Equipes	3 pessoas por equipe
	Apoio Ferramental	Sim, este processo pode ser automatizado utilizando a ferramenta proposta por outro professor
	Atividades	<ol style="list-style-type: none"> 1. Realização da leitura dos diagramas, através de uma inspeção. 2. Baseado nos comentários dos inspetores, os estudantes realizam a melhoria no diagrama.
PBL + LBL DOUBLE TRACK TEACHING METHOD – S116	Descrição	O método de ensino de trilha dupla baseia-se na aprendizagem baseada em leitura (LBL) e a aprendizagem baseada em problemas (PBL). Neste método de ensino o professor ensina a base teórica relacionada ao problema real
	Modelo a ser ensinado	Qualquer diagrama
	Tipo de Método	Resolução de problemas
	Papel dos estudantes	Não há papéis
	Tempo Máximo	Durante toda a disciplina. Porém pode ser adaptado para um conteúdo específico.
	Tempo Mínimo	Não informado
	Formação de Equipes	Mínimo 3 pessoas
	Apoio Ferramental	Não.
	Atividades	<ol style="list-style-type: none"> 1. O professor ensina a teoria dos conteúdos. 2. Prover dados de ensino de exemplo real de desenvolvimento de software) para todos os alunos, solicitar que o aluno divida-se em grupo e escolha um tópico, procure material de referência por vários recursos (biblioteca e redeetc.) no horário de pós-aula. O professor continua simples de guia antes da lição. 3. Os alunos continuam a conversar. 4. O professor guia e incentiva o aluno no tempo. O aluno fala e explica em detalhes, discute e traz uma nova pergunta para formar professores e a interação efetiva dos alunos.
MULTIMODAL APPROACH COMBINING PROJECT-BASED LEARNING WITH WEEKLY QUIZZES,	Descrição	Esta abordagem multimodal visa fomentar habilidades de design relevantes para a indústria, combinando uma abordagem orientada por projeto com mecanismos de feedback automatizados e questionários.
	Modelo a ser ensinado	Diagrama de classes e sequência, mas pode ser adaptado
	Tipo de Método	Discussão

TECNOLOGIA E ID	CARACTERÍSTICAS DAS TECNOLOGIAS	
TESTS AND ACTIVE LEARNING TASKS – S137	Papel dos estudantes	Não tem papeis
	Tempo Máximo	2 semanas o ciclo completo
	Tempo Mínimo	Não informado. Uma aula para cada ponto
	Formação de Equipes	Combinação de individual e grupo
	Apoio Ferramental	Para os testes online, os autores disponibilizam uma ferramenta
	Atividades	Primeira semana ou etapa 1 (entrega): <ol style="list-style-type: none"> 1. Ministrara uma aula com a classe inteira onde os tópicos gerais do curso são delineados 2. Ministre Tutoriais menores (aproximadamente 25) 3. Realize tarefas de aprendizagem ativa e passiva na web (individualmente) Segunda semana ou fase 2 (avaliação): <ol style="list-style-type: none"> 4. Uma rápida revisão da aula da semana anterior, seguida por um teste on-line ou um exercício de design, desenhando um diagrama UML para um cenário particular.
POSITIVE EXAMPLES ON OBJECT MODELING – S148	Descrição	Neste método são fornecidas diretrizes associadas a exemplos positivos e descrições que são incorporados a um ambiente de treinamento
	Modelo a ser ensinado	Diagrama de classes, mas pode ser adaptado desde que se tenha os defeitos.
	Tipo de Método	Resolução de problemas
	Papel dos estudantes	Não há papeis
	Tempo Máximo	180 minutos
	Tempo Mínimo	95 minutos
	Formação de Equipes	Individual
	Apoio Ferramental	Sim, possui para direcionar na marcação dos erros nos diagramas.
	Atividades	<ol style="list-style-type: none"> 1. O instrutor forneceu um folheto de instruções gerais sobre o experimento e explicou os procedimentos (10 min). 2. Cada assunto desenvolveu um modelo de objeto (diagrama de classes da versão 1) para o sistema (25 min). 3. Ruptura (10 min). 4. Cada assunto analisou problemas de qualidade comum de modelos de objetos ilustrados com exemplos positivos ou negativos no contexto de um estudo de bookstorecase universitário (30 min). 5. Cada assunto então revisou sua versão inicial do modelo de objeto e enviou diagrama de classe da versão 2 (25 min).

TECNOLOGIA E ID	CARACTERÍSTICAS DAS TECNOLOGIAS	
NEGATIVE EXAMPLES ON OBJECT MODELING – S148	Descrição	Neste método são fornecidos diretrizes associadas a exemplos negativos e descrições que são incorporados a um ambiente de treinamento
	Modelo a ser ensinado	Diagrama de classes, mas pode ser adaptado desde que se tenha os defeitos.
	Tipo de Método	Resolução de problemas
	Papel dos estudantes	Não há papeis
	Tempo Máximo	180 minutos
	Tempo Mínimo	95 minutos
	Formação de Equipes	Individual
	Apoio Ferramental	Sim, possui para direcionar na marcação dos erros nos diagramas.
	Atividades	<ol style="list-style-type: none"> 1. O instrutor forneceu um folheto de instruções gerais sobre o experimento e explicou os procedimentos (10 min). 2. Cada assunto desenvolveu um modelo de objeto (diagrama de classes da versão 1) para o sistema (25 min). 3. Ruptura (10 min). 4. Cada assunto analisou problemas de qualidade comum de modelos de objetos ilustrados com exemplos positivos ou negativos no contexto de um estudo caso de uma livraria universitária (30 min). 5. Cada assunto então revisou sua versão inicial do modelo de objeto e enviou diagrama de classe da versão 2 (25 min).
APRENDIZAGEM BASEADA EM SISTEMAS FAMILIARES – S149	Descrição	Nesta abordagem, os alunos investigam um sistema familiar bastante complexo, como um temporizador, sistema de controle de cruzeiro, caixas eletrônicos, sistema de elevador e sistema de micro-ondas. As vantagens desta abordagem é proporcionar aos alunos um conhecimento da metodologia de desenvolvimento, dar aos alunos um conhecimento útil de uma técnica de especificação, mostrar aos alunos a importância da especificação correta, e permitir aos alunos testar a especificação usando um sistema implementado.
	Modelo a ser ensinado	Diagrama de classes e estados, mas pode ser adaptado
	Tipo de Método	Resolução de problemas
	Papel dos estudantes	Não há papéis
	Tempo Máximo	Não informado
	Tempo Mínimo	Não informado
	Formação de Equipes	Mínimo 3 pessoas
	Apoio Ferramental	Não.

TECNOLOGIA E ID	CARACTERÍSTICAS DAS TECNOLOGIAS	
	Atividades	Primeiramente, os grupos de estudantes tentam identificar as funcionalidades básicas do sistema e a partir disto, o grupo tenta modelar um diagrama de objetos e tentando relacionar os objetos identificados. E cada nível de abstração que os estudantes entrarem representam um ciclo do processo de ensino. Em seguida, os estudantes tentam modelar o comportamento do sistema através de um diagrama de estados
JIGSAW TECHNIQUE – S153	Descrição	Neste método, cada aluno em uma equipe de serra terá que realizar uma dessas tarefas parciais, que acabará por ser integrada por todos os membros da equipe.
	Modelo a ser ensinado	Diagrama de classes, mas pode ser adaptado
	Tipo de Método	Ensino recíproco
	Papel dos estudantes	Não
	Tempo Máximo	120 minutos
	Tempo Mínimo	9 minutos
	Formação de Equipes	3 a 6 pessoas
	Apoio Ferramental	Não
	Atividades	<ol style="list-style-type: none"> 1. Teste inicial 2. Entrega de material, explicação e formação de grupos (2 alunos) 3. Desempenho de uma tarefa atribuída pelos alunos 4. Formação de grupos de especialistas (máximo de 6 alunos) e desempenho de uma atividade 5. Formação de grupos de estudantes que integrarão diagramas (máximo de 6 alunos) 6. Encerramento do trabalho cooperativo
PROBLEM BASED LEARNING - Estudo – Estudo Experimental	Descrição	PBL é um método pedagógico que enfatiza o papel de um problema e os alunos são responsáveis pela sua aprendizagem. Este método tem uma dupla ênfase em ajudar os alunos a desenvolver estratégias e a construir conhecimentos
	Modelo a ser ensinado	Atividades e Sequência, mas pode ser adaptado a qualquer modelo
	Tipo de Método	Resolução de problemas
	Papel dos estudantes	Não
	Tempo Máximo	240 minutos
	Tempo Mínimo	120 minutos
	Formação de Equipes	3 a 5 estudantes

TECNOLOGIA E ID	CARACTERÍSTICAS DAS TECNOLOGIAS	
	Apoio Ferramental	Não
	Atividades	
LEARNING FROM ERRONEOUS EXAMPLES – Estudo Experimental	Descrição	Aprendizagem Baseada Exemplos Errouneuos (ErrEx) é uma abordagem baseada em AL, pois engaja os alunos em um exercício de aprendizagem ativa, oferecendo a oportunidade para uma learning experience construtiva e colaborativa através dos exemplos errados.
	Modelo a ser ensinado	Atividades e Sequência, mas pode ser adaptado a qualquer modelo
	Tipo de Método	Resolução de problemas
	Papel dos estudantes	Não
	Tempo Máximo	240 minutos
	Tempo Mínimo	120 minutos
	Formação de Equipes	3 a 5 estudantes
	Apoio Ferramental	Não
	Atividades	

Tecnologia	Descrição	Modelo a ser ensinado	Tipo de Método	Papel	Tempo Máximo	Tempo Mínimo	Formação de Equipes?	Atividade	Tarefa e descrição	Apoio Ferramental
Teaching Methods for Conceptual Modeling Based On Error Analyses Of Class Diagrams - S1	O método visa auxiliar os estudantes novatos a partir dos erros típicos que estudantes novatos realizam nos diagramas.	Diagrama de classes, pois são definidos critérios para o ensino destes diagramas	Resolução de problemas	Não	90 minutos	Não informado	Individualmente. Não podem discutir e resolver os problemas em grupos	4. Professor categorizar os erros em sintáticos, semânticos e pragmáticos. 5. Entregar os tipos de erros para os estudantes. 6. Estudantes realizam a modelagem		
a method based on think-pair-square – S16	O método tem como objetivo partir do princípio do trabalho individual, para o em pares e, por fim, em grupo	Diagrama de casos de uso, mas pode ser adaptado para outros	Discussão	Não	239	348	Cooperativa. No mínimo 4 integrantes	5. Uso de um questionário para balancear os grupos 6. Think - É realizado individualmente. Cada membro da equipe trabalha para produzir um ou mais diagramas de casos de uso. 7. Pair - É executado em par. Os membros da equipe trabalham em pares nos modelos que identificaram individualmente. 8. Square- Todos os membros da equipe juntos exploram o modelo de cada par para produzir um modelo comum e mais		Há a necessidade de um ambiente integrado para realizar a comunicação e obter uma familiaridade entre os membros do time.

Tecnologia	Descrição	Modelo a ser ensinado	Tipo de Método	Papel	Tempo Máximo	Tempo Mínimo	Formação de Equipes?	Atividade	Tarefa e descrição	Apoio Ferramental
								abrangente. As equipes podem ser compostas de qualquer número par maior ou igual a quatro.		
a method to detect learners' difficult points in object modeling exercises – S17	O método analisa o processo de criação do diagrama da classe dos alunos e detecta os padrões comportamentais dos alunos que aparecem quando os alunos enfrentam pontos difíceis.	Diagrama de classes, pois tem critérios apenas para este tipo de diagrama	Resolução de problemas		Não informado	Não informado	individualmente	<ol style="list-style-type: none"> 1. Um professor explica a técnica de modelagem de objetos e dá uma tarefa. 2. Os alunos trabalham nela usando nossa ferramenta e enviam o artefato que eles fizeram. 3. O professor procura uma lista de artefatos submetidos. Neste momento, a ferramenta detecta os aprendizes cujos artefatos têm métricas outras e os enfatiza visualmente, como mudar suas cores. 4. O professor remete o processo de criação do artefato dos alunos que foi enfatizado por ferramenta e reconhece os pontos difíceis dos alunos. A ferramenta ajuda isso, fornecendo um visualizador de processos integrado com um visualizador 		Se o professor quiser, ele pode utilizar a fermenta para gerar a identificação dos defeitos de forma automática

Tecnologia	Descrição	Modelo a ser ensinado	Tipo de Método	Papel	Tempo Máximo	Tempo Mínimo	Formação de Equipes?	Atividade	Tarefa e descrição	Apoio Ferramental
								de métricas e processando replayer. 5. O professor analisa todos os artefatos submetidos. A ferramenta mostra o resultado da análise, como os pontos difíceis dos alunos, as métricas calculadas e assim por diante. 6. O professor fornece aos alunos um feedback detalhado sobre o processo de criação com base nas informações descritas na etapa anterior, além do feedback normal baseado no artefato. Isso pode ser efetivo para ativar a reflexão dos aprendizes. 7. Os alunos reproduzem seu processo e refletem sua aprendizagem com base no feedback.		
Métodos de Aprendizagem utilizando dados de inspeções – S81	O método tem por objetivo auxiliar na leitura e criação de diagramas UML, baseado nos comentários	Diagrama de classes, mas pode ser adaptado	discussão	Há dois papéis: o inspetor e o modelador .	Não informado. Mas pode ser empregado em uma	Não informado	Sim. Porém, não é informada a quantidade. Mínimo de 3 pessoas	3. Realização da leitura dos diagramas, através de uma inspeção. 4. Baseado nos comentários dos inspetores, os		Sim, este processo pode ser automatizado utilizando a ferramenta proposta por

Tecnologia	Descrição	Modelo a ser ensinado	Tipo de Método	Papel	Tempo Máximo	Tempo Mínimo	Formação de Equipes?	Atividade	Tarefa e descrição	Apoio Ferramental
	realizados nos diagramas da UML				aula de 90min.			estudantes realizam a melhoria no diagrama.		outro professor
PBL + LBL Double Track Teaching Method – S116	O método de ensino de trilha dupla baseia-se na aprendizagem baseada em leitura (LBL) e a aprendizagem baseada em problemas (PBL). Neste método de ensino o professor ensina a base teórica relacionada ao problema real	Qualquer diagrama	Resolução de problemas	Não há papéis	Durante toda a disciplina. Porém pode ser adaptado para um conteúdo específico.	Não informado	Sim, porém não informado. Três pessoas	5. O professor ensina a teoria dos conteúdos. 6. Prover dados de ensino de exemplo real de desenvolvimento de software) para todos os alunos, solicitar que o aluno divida-se em grupo e escolha um tópico, procure material de referência por vários recursos (biblioteca e redeetc.) no horário de pós-aula. O professor continua simples de guia antes da lição. Os alunos continuam a conversar. O professor guia e incentiva o aluno no tempo. O aluno fala e explica em detalhes, discute e traz uma nova pergunta para formar professores e a interação efetiva dos alunos.		Não necessita.

Tecnologia	Descrição	Modelo a ser ensinado	Tipo de Método	Papel	Tempo Máximo	Tempo Mínimo	Formação de Equipes?	Atividade	Tarefa e descrição	Apoio Ferramental
Multimodal approach combining project-based learning with weekly quizzes, tests and active learning tasks – S137	Esta abordagem multimodal visa fomentar habilidades de design relevantes para a indústria, combinando uma abordagem orientada por projeto com mecanismos de feedback automatizados e questionários.	Diagrama de classes e sequência, mas pode ser adaptado	Discussão	Não tem papeis	2 semanas o ciclo completo	Não informado. Uma aula para cada ponto	Individual para grupo	Primeira semana ou etapa 1 (entrega): 1. Ministrando uma aula com a classe inteira onde os tópicos gerais do curso são delineados 2. Ministre Tutoriais menores (aproximadamente 25) 3. Realize tarefas de aprendizagem ativa e passiva na web (individualmente) Segunda semana ou fase 2 (avaliação): 4. Uma rápida revisão da aula da semana anterior, seguida por um teste on-line ou um exercício de design, desenhando um diagrama UML para um cenário particular.		Para os testes online, os autores disponibilizam uma ferramenta
Positive Examples on Object Modeling – S148	Neste método são fornecidos diretrizes associadas a exemplos	Diagrama de classes, mas pode ser	Resolução de problemas	Não há papeis	180 minutos	95 minutos	Individual	1. O instrutor forneceu um folheto de instruções gerais sobre o experimento e explicou os		Sim, possui para direcionar na marcação dos

Tecnologia	Descrição	Modelo a ser ensinado	Tipo de Método	Papel	Tempo Máximo	Tempo Mínimo	Formação de Equipes?	Atividade	Tarefa e descrição	Apoio Ferramental
	positivos e descrições que são incorporados a um ambiente de treinamento	adaptado desde que se tenha os defeitos.						procedimentos (10 min). 2. Cada assunto desenvolveu um modelo de objeto (diagrama de classes da versão 1) para o sistema (25 min). 3. Ruptura (10 min). 4. Cada assunto analisou problemas de qualidade comum de modelos de objetos ilustrados com exemplos positivos ou negativos no contexto de um estudo de bookstorecase universitário (30 min). 5. Cada assunto então revisou sua versão inicial do modelo de objeto e enviou diagrama de classe da versão 2 (25 min).		erros nos diagramas.
Negative Examples on Object Modeling – S148	Neste método são fornecidos diretrizes associadas a exemplos negativos e descrições que são incorporados a um ambiente de treinamento	Diagrama de classes, mas pode ser adaptado desde que se tenha os defeitos.	Resolução de problemas	Não há papéis	180 minutos	95 minutos	Individual	1. O instrutor forneceu um folheto de instruções gerais sobre o experimento e explicou os procedimentos (10 min). 2. Cada assunto desenvolveu um modelo de objeto		Sim, possui para direcionar na marcação dos erros nos diagramas.

Tecnologia	Descrição	Modelo a ser ensinado	Tipo de Método	Papel	Tempo Máximo	Tempo Mínimo	Formação de Equipes?	Atividade	Tarefa e descrição	Apoio Ferramental
								(diagrama de classes da versão 1) para o sistema (25 min). 3. Ruptura (10 min). 4. Cada assunto analisou problemas de qualidade comum de modelos de objetos ilustrados com exemplos positivos ou negativos no contexto de um estudo de bookstorecase universitário (30 min). 5. Cada assunto então revisou sua versão inicial do modelo de objeto e enviou diagrama de classe da versão 2 (25 min).		
Aprendizagem Baseada em sistemas familiares – S149	Nesta abordagem, os alunos investigam um sistema familiar bastante complexo, como um temporizador, sistema de controle de cruzeiro, caixas eletrônicos, sistema de elevador e	Diagrama de classes e estados, mas pode ser adaptado	Resolução de problemas	Não há papéis	Não informado	Não informado	Mínimo 3 pessoas	Primeiramente, os grupos de estudantes tentam identificar as funcionalidades básicas do sistema e a partir disto, o grupo tenta modelar um diagrama de objetos e tentando relacionar os objetos identificados. E cada nível de abstração que os estudantes entrarem representam um ciclo do processo de ensino. Em seguida, os		Não

Tecnologia	Descrição	Modelo a ser ensinado	Tipo de Método	Papel	Tempo Máximo	Tempo Mínimo	Formação de Equipes?	Atividade	Tarefa e descrição	Apoio Ferramental
	<p>sistema de microondas. As vantagens desta abordagem incluem</p> <ul style="list-style-type: none"> • proporcionar aos alunos um conhecimento da metodologia de desenvolvimento • dar aos alunos um conhecimento útil de uma técnica de especificação, • mostrar aos alunos a importância da especificação correta, e • permitir aos alunos testar a especificação usando um sistema implementado. 							estudantes tentam modelar o comportamento do sistema através de um diagrama de estados		
Jigsaw Technique – S153	Neste método, cada aluno em uma equipe de serra terá que realizar uma dessas tarefas parciais, que	Diagrama de classes, mas pode ser adaptado	Ensino recíproco	Não	120 minutos	9 minutos	3 a 6 pessoas	<ol style="list-style-type: none"> 1. Teste inicial 2. Entrega de material, explicação e formação de grupos (2 alunos) 3. Desempenho de uma tarefa atribuída pelos alunos 		Não necessita

Tecnologia	Descrição	Modelo a ser ensinado	Tipo de Método	Papel	Tempo Máximo	Tempo Mínimo	Formação de Equipes?	Atividade	Tarefa e descrição	Apoio Ferramental
	acabará por ser integrada por todos os membros da equipe.							<p>4. Formação de grupos de especialistas (máximo de 6 alunos) e desempenho de uma atividade</p> <p>5. Formação de grupos de estudantes que integrarão diagramas (máximo de 6 alunos)</p> <p>6. Encerramento do trabalho cooperativo</p>		

REFERÊNCIAS

- Barkley, E. F., Cross, K. P., and Major, C. H. “Collaborative learning techniques: A handbook for college faculty” John Wiley & Sons, 2014.
- Bera, P. “Analyzing the cognitive difficulties for developing and using uml class diagrams for domain understanding”. In *Journal of Database Management (JDM)*, vol. 23, no. 3, pp. 1–29, 2012.
- Boberic-Krstićev, D., e Tešendić, D. “Experience in teaching OOAD to various students”. In *Informatics in Education*, vol. 12, no. 1, pp. 43–58, 2013.
- Bolloju, N., Schneider, C., and Vogel, D. “Asymmetrical effects of using positive and negative examples on object modeling”. In *Information Systems Development*, pp. 85-96, 2011.
- El Alloui, Y., and El Beqqali, O. “User profile Ontology for the Personalization approach”. In *International Journal of Computer Applications* 41(4) (2011) 31-40.
- Fernandez, A., Insfran, E., e Abrahão, S. “Usability evaluation methods for the web: A systematic mapping study”. In *Information and software Technology*, vol. 53, no. 8, pp. 789–817, 2011.
- Kinjo, T., Ohgame, Y., and Hazeyama, A. “An Object-Oriented Modeling Learning Support System Using Inspection Comments”. In *Journal of Information and Systems in Education* 5(1) (2006) 66–75.
- Kitchenham, B. A., e Charters, S. “Guidelines For Performing Systematic Literature Reviews in Software Engineering”. In *EBSE Technical Report EBSE-2007-01*, 2007.
- Kruus, H., Robal, T., e Jervan, G. “Teaching modeling in SysML/UML and problems encountered”. In *2014 25th EAEEIE Annual Conference (EAEEIE)*, pp. 33-36, 2014.
- Kuhrmann, M., Fernández, D.M., e Daneva, M. “On the pragmatic design of literature studies in software engineering: an experience-based guideline”. In *Empirical software engineering*, vol. 22, no. 6, pp. 2852–2891, 2017.
- Landis, J., e Koch, G. (1977). The measurement of observer agreement. *Biometrics* 1977, 159-174.
- Moisan, S., e Rigault, J. P. “Teaching Object-Oriented Modeling and UML to Various Audiences”. In *International Conference on Model Driven Engineering Languages and Systems Workshops*, pp. 40-54, 2009.
- Petersen, K., Vakkalanka, S., e Kuzniarz, L. “Guidelines for conducting systematic mapping studies in software engineering: An update”. *Information and Software Technology*, vol. 64, pp. 1–18, 2015.
- Portillo, J. A. P. S., and Campos, P. G. “The jigsaw technique: experiences teaching analysis class diagrams”. In *Mexican Int. Conf. on Computer Science*, pp. 289-293, 2009.
- Santos, G., Rocha, A. R., Conte, T., Barcellos, M. P., Prikladnicki, R., 2012. Strategic Alignment between Academy and Industry: A Virtuous Cycle to Promote Innovation in Technology. In *26º Simpósio Brasileiro de Engenharia de Software (SBES 2012)*, v. 1. p. 196-200.

- Scanniello, G., and Erra, U. "Distributed modeling of use case diagrams with a method based on think-pair-square: Results from two controlled experiments". In *Journal of Visual Languages & Computing* 25(4) (2014) 494-517.
- Sien, V. Y. "An investigation of difficulties experienced by students developing unified modelling language (UML) class and sequence diagrams". In *Computer Science Education*, vol. 21, no. 4, pp. 317-342, 2011.
- Silva, W., Steinmacher, I. F. and Conte, T. U. "Is It Better to Learn from Problems or Erroneous Examples?". In *30th Int. Conf. on Software Engineering Education and Training (CSEE&T)*, pp. 222-231, 2017.
- Steinmacher, I.; Silva, M. A. G.; Gerosa, M. A.; e Redmiles, D. F. "A systematic literature review on the barriers faced by newcomers to open source software projects". In *Information and Software Technology*, v. 59, pp. 67-85, 2015.
- Stikkolorum, D.R., Ho-Quang, T., Karasneh, B., Chaudron, M.R. "Uncovering students' common difficulties and strategies during a class diagram design process: an online experiment". In: *EduSymp@ MoDELS*, pp. 29-42, 2015.
- Stoecklin, S., Williams, D. D, and Swain, R. "Understanding object-oriented systems specifications using familiar systems", In *Int. Conf. Software Engineering: Education & Practice*, pp. 10-15, 1998.
- Strauss, A., e Corbin, J. "Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory". 2 ed. SAGE Publications, London, 1998.
- Tanaka, T., Mori, K., Hashiura, H., Hazeyama, A., e Komiya, S. "Proposals of a Method Detecting Learners' Difficult Points in Object Modeling Exercises and a Tool to Support the Method". In *IIAI 3rd International Conference on Advanced Applied Informatics (IIAIAA)*, pp. 650-655, 2014.
- Thevathayan, C., and Hamilton, M. "Imparting software engineering design skills". In *Proc. Nineteenth Australasian Computing Education Conference*, pp. 95-102, 2017.
- Wang, Y., Su, R., and Li, G. "Research on PBL and LBL Double Track Teaching Model in Unified Modeling Language Teaching Based on Outstanding Engineers". In *Informatics and Management Science*, pp. 381-386, 2013.
- Zhang, H., Kitchenham, B., Pfahl, D. "Software process simulation modeling: an extended systematic review". In *International Conference on Software Process*, pp. 309-320, 2010.

