

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE SÃO PAULO
PUC-SP**

Wilson Vendramel

Adoção de uma abordagem neopiagetiana e de padrões de abstração para o ensino de engenharia de software: proposição de um modelo pedagógico de correlação entre o alinhamento construtivo e a técnica OC2-RD2

Doutorado em Tecnologias da Inteligência e Design Digital

São Paulo

2023

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE SÃO PAULO
PUC-SP**

Wilson Vendramel

Adoção de uma abordagem neopiagetiana e de padrões de abstração para o ensino de engenharia de software: proposição de um modelo pedagógico de correlação entre o alinhamento construtivo e a técnica OC2-RD2

Tese apresentada à Banca Examinadora da Pontifícia Universidade Católica de São Paulo, como exigência parcial para obtenção do título de Doutor em Tecnologias da Inteligência e Design Digital, na área de concentração de Processos Cognitivos e Ambientes Digitais, sob a orientação do Prof. Dr. Ítalo Santiago Vega.

São Paulo

2023

Banca Examinadora

Dedico este trabalho à minha mãe Clotilde Sita Vendramel, ao meu pai Élcio Vendramel (*in memoriam*), e ao meu amigo Anderson dos Santos Siqueira (*in memoriam*).

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001 – 88887.369458/2019-00.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001 – 88887.369458/2019-00.

AGRADECIMENTOS

Aos meus (ex) professores/mestres/orientadores.

Aos meus (ex) alunos/aprendizes/orientandos.

Aos participantes da pesquisa.

Ao Programa de Tecnologias da Inteligência e Design Digital.

Aos que de alguma forma, direta ou indireta, contribuíram com este trabalho.

“Se você só lê o mesmo que todo mundo lê, acaba pensando o mesmo que todo mundo pensa”.

(Haruki Murakami)

RESUMO

Tendo em vista de que sistemas de software são essenciais para vários campos de conhecimento, o ensino de engenharia de software se torna crucial para a construção de produtos de programação de qualidade para a sociedade. Isso implica na demanda por estudantes qualificados ao longo da vida, dado que a qualidade no ensino tem impacto na melhoria do processo de desenvolvimento de software. Os estudantes de computação e informática precisam frequentemente usar abstrações, construir modelos mentais de análise e projeto, além de representá-los adequadamente nos muitos e diversos artefatos produzidos durante o ciclo de vida de software, porém, por outro lado, a diversidade de propósitos, formas e níveis de abstração não são facilmente compreendidos. O repertório teórico de engenharia de software é vasto, complexo, e repleto de conceitos concretos e abstratos, assim sendo, é significativo existir uma conjuntura pedagógica apropriada para o ensino dessa área da computação. Embora grande parte da teoria piagetiana lide com a aprendizagem de crianças e adolescentes, princípios teóricos piagetianos foram estendidos ao aprendizado de adultos, como é o caso da teoria neopiagetiana que expande a visão piagetiana clássica em diversas áreas, onde uma das perspectivas é a de que os estágios de desenvolvimento cognitivo vão além da infância e continuam na fase adulta, quando aplicados em novos contextos cognitivos. Diante do exposto, este estudo investiga uma lacuna relacionada ao ensino de engenharia de software, com o intuito de permitir ao estudante condições de atingir níveis de complexidade cognitiva crescente na construção de conhecimento concreto e abstrato. Para tal, o objetivo deste trabalho é propor um modelo pedagógico construído a partir da correlação entre o alinhamento construtivo e os planos organizacionais da técnica OC2-RD2, a fim de auxiliar o professor no planejamento de ensino de engenharia de software. O modelo proposto se inspira em uma visão neopiagetiana, sustentado pelos verbos de aprendizagem da taxonomia cognitiva SOLO, como também por uma coleção de padrões de abstração que visam fortalecer o pensamento abstrato, aspirando, desta forma, o desenvolvimento dessa capacidade cognitiva, considerada uma habilidade-chave para os estudantes de cursos da área computacional. Tal estudo é de abordagem qualitativa, propósito exploratório e aplica a entrevista semiestruturada como instrumento de pesquisa. O roteiro da entrevista foi aplicado junto a uma amostra de professores de instituições de ensino distintas. Os dados provenientes do conhecimento e experiência dos docentes foram analisados e interpretados, resultando na construção de um modelo pedagógico (metamodelo) de ensino de engenharia de software, especificamente para a área de modelagem e análise de software. Tal metamodelo é constituído por um conjunto de planos ontológicos temáticos e reutilizáveis de conteúdo e cognição, bem como alinhados construtivamente, a fim de nortear o docente, do ponto de vista de desenvolvimento cognitivo e progressão do aprendizado, no planejamento de cursos e aulas de engenharia de software em diversos tempos e espaços.

Palavras-chave: Ensino de Engenharia de Software, Teoria Neopiagetiana, Padrões de Abstração, Alinhamento Construtivo, Técnica OC2-RD2.

ABSTRACT

Considering that software systems are essential for several fields of knowledge, teaching software engineering becomes crucial for building quality programming products for society. This implies a demand for qualified students throughout life, given that the quality of teaching has an impact on improving the software development process. Computing and informatics students often need to use abstractions, build mental analysis and design models, and adequately represent them in the many and diverse artifacts produced during the software life cycle, however, on the other hand, the diversity of purposes, forms and levels of abstraction are not easily understood. The theoretical repertoire of software engineering is vast, complex, and full of concrete and abstract concepts, therefore, it is significant to have an appropriate pedagogical environment for teaching this area of computing. Although a large part of piagetian theory deals with the learning of children and adolescents, piagetian theoretical principles have been extended to adult learning, as is the case of the neo-piagetian theory that expands the classic piagetian view in several areas, where one of the perspectives is that stages of cognitive development extend beyond childhood and continue into adulthood when applied to new cognitive contexts. Given the above, this study investigates a gap related to the teaching of software engineering, with the aim of allowing students to reach levels of increasing cognitive complexity in the construction of concrete and abstract knowledge. To this end, the objective of this work is to propose a pedagogical model built from the correlation between the constructive alignment and the organizational plans of the OC2-RD2 technique, in order to assist the teacher in planning software engineering teaching. The proposed model is inspired by a neo-piagetian vision, supported by the learning verbs of the SOLO cognitive taxonomy, as well as by a collection of abstraction patterns that aim to strengthen abstract thinking, aspiring, in this way, to develop this cognitive capacity, considered a skill -key for students of courses in the computational area. This study has a qualitative approach, exploratory purpose and applies the semi-structured interview as a research tool. The interview script was applied to a sample of teachers from different educational institutions. Data from the knowledge and experience of the professors were analyzed and interpreted, resulting in the construction of a pedagogical model (metamodel) for teaching software engineering, specifically for the area of software modeling and analysis. This metamodel consists of a set of thematic and reusable ontological plans of content and cognition, as well as constructively aligned, in order to guide the teacher, from the point of view of cognitive development and learning progression, in the planning of engineering courses and classes of software in different times and spaces.

Keywords: Teaching Software Engineering, Neo-Piagetian Theory, Abstraction Patterns, Constructive Alignment, OC2-RD2 Technique.

LISTA DE QUADROS

Quadro 1 – Áreas de Conhecimento do SEEK	36
Quadro 2 – Unidades e tópicos da área de modelagem e análise de software	37
Quadro 3 – Modos e níveis da taxonomia SOLO	40
Quadro 4 – Síntese do referencial teórico utilizado na construção do procedimento de pesquisa	69
Quadro 5 - Caracterização da amostra	73
Quadro 6 – Desenvolvimento cognitivo em uma perspectiva neopiagetiana para MAS	76
Quadro 7 – Tópicos de Modelagem e Análise de Software	78
Quadro 8 – Verbos de aprendizagem para cada tópico por participante	79
Quadro 9 – Verbo de aprendizagem <i>versus</i> tópico por participante	80
Quadro 10 – Verbos de aprendizagem e tópicos de MAS eleitos	81
Quadro 11 – Resultados pretendidos de aprendizagem descobertos	82
Quadro 12 – Mapeamento entre os RPAs da pesquisa e os dos participantes	83
Quadro 13 – Mapeamento entre os RPAs da pesquisa e as AEAs dos participantes	84
Quadro 14 – Mapeamento entre os RPAs da pesquisa e as TAs dos participantes	85
Quadro 15 – Descrição dos padrões de abstração	86
Quadro 16 – Mapeamento entre RPAs da pesquisa e padrões de abstração por participante	88
Quadro 17 – Mapeamento entre RPA e Itens de Conhecimento	91
Quadro 18 – Quatro erros comuns de abstração	92
Quadro 19 – Enunciados das perguntas sobre o modelo proposto	94
Quadro 20– Descrição dos Resultados Pretendidos de Aprendizagem	105
Quadro 21 – Denominações dos planos ontológicos	111
Quadro 22 – Descrição dos desempenhos cognitivos de abstração	116
Quadro 23 – Objetivos de aprendizagem do plano da disciplina “A”	129
Quadro 24 – Aplicação do MPO_MAS na disciplina “A”	130
Quadro 25 – Objetivos de aprendizagem do plano da disciplina “B”	131
Quadro 26 – Aplicação do MPO_MAS na disciplina “B”	131
Quadro 27 – Objetivos de aprendizagem do plano da disciplina “C”	133
Quadro 28 – Aplicação do MPO_MAS na disciplina “C”	133
Quadro 29 – Objetivos de aprendizagem do plano da disciplina “D”	135
Quadro 30 – Aplicação do MPO_MAS na disciplina “D”	135

LISTA DE FIGURAS

Figura 1 – Pentágono conceitual da pesquisa	33
Figura 2 – Uma hierarquia dos verbos da taxonomia SOLO	42
Figura 3 – Modos, ciclos de aprendizagem e formas de conhecimento	43
Figura 4 – Etapas do alinhamento construtivo	45
Figura 5 – Uma imagem abstrata	46
Figura 6 – Propósitos, formas e níveis de abstração na mente do Fubã	49
Figura 7 – The London Underground Map (a) o mapa de 1928 e (b) o mapa de 1933.....	50
Figura 8 – Objeto livro e algumas representações	55
7Figura 9 – Texto narrativo de Bal	56
Figura 10 – Gráfico de Kurt Vonnegut para explicar o drama da Cinderela	56
Figura 11 – Narração no ambiente de aprendizagem	57
Figura 12 – Estrutura da fábula OC2-RD2	58
Figura 13 – Estrutura da história OC2-RD2	59
Figura 14 – Planos organizacionais OC2-RD2	60
Figura 15 – Tipos de dependência entre itens de conhecimento	61
Figura 16 – Percurso metodológico da pesquisa	66
Figura 17 – Estrutura de dependências no plano de conteúdos OC2-RD2	90
Figura 18 – Construção de modelos pedagógicos.....	101
Figura 19 – Correlação entre o alinhamento construtivo e a técnica OC2-RD2.....	103
Figura 20 – Módulos de Alinhamento Construtivo	106
Figura 21 – Módulo de Modelos Ontológicos para os oito RPAs da pesquisa	107
Figura 22 – Módulo Taxonomia SOLO	108
Figura 23 – Módulo de padrões de abstração	109
Figura 24 – Planos ontológicos de modelagem e análise de software por nível SOLO.....	112
Figura 25 – Planos ontológicos por nível SOLO em uma abordagem de facilitação descendente e ascendente de aprendizagem	113
Figura 26 – Ponte cognitiva na mente de Fubã	114
Figura 27 – Planos ontológicos de modelagem e análise de software por nível SOLO e capacidade cognitiva de abstração	115
7Figura 28 – Módulos da arquitetura do MPO_MAS	117
Figura 29 – Módulos e Conceitos da Arquitetura do MPO_MAS.....	118
Figura 30 – Plano ontológico de MAS_PM_U.....	119
Figura 31 – Plano ontológico de MAS_PM_M	119
Figura 32 – Plano ontológico de MAS_MI_M.....	120
Figura 33 – Plano ontológico de MAS_MI_R	121
Figura 34 – Plano ontológico de MAS_MC_R	122
Figura 35 – Plano ontológico de MAS_MA_R.....	123
Figura 36 – Plano ontológico de MAS_MD_R	123
Figura 37 – Plano ontológico de MAS_AC_AE	124
Figura 38 – Etapas para aplicação do MPO_MAS	126

LISTA DE FIGURAS

Figura 39 – Perspectiva de integração entre APIHA e MPO_MAS	142
---	-----

LISTA DE TABELAS

Tabela 1 – Avaliação de adequação dos padrões de abstração por participante	87
Tabela 2 – Avaliação de adequação dos erros de abstração	93
Tabela 3 – Avaliação de adequação do modelo pedagógico proposto	95

LISTA DE ABREVIATURAS E SIGLAS

AC: Alinhamento Construtivo

ACM: Association for Computing Machinery

AEA: Atividade de Ensino e Aprendizagem

APIHA: Ambiente Presencial Interativo e Híbrido de Aprendizagem

APOO: Análise e Projeto Orientado a Objetos

AT: Assessment Task

CA: Constructive Alignment

ESOO: Engenharia de Software Orientada a Objetos

IEEE: Institute of Electrical and Electronics Engineers

ILO: Intended Learning Outcome

MAS: Modelagem e Análise de Software

MPO_MAS: Modelo de Planos Ontológicos de Modelagem e Análise de Software

OC2-RD2: Objetivo-Contratempo-Catástrofe-Reação-Dilema-Decisão

OMG: Object Management Group

RPA: Resultado Pretendido de Aprendizagem

SOLO: Structure of Observed Learning Outcome

TA: Tarefa de Avaliação

TLA: Teaching Learning Activities

UML: Unified Modeling Language

Sumário

1	INTRODUÇÃO	16
1.1	Justificativas e Questão de Pesquisa	28
1.2	Objetivo Geral	29
1.3	Objetivos Específicos	30
1.4	Hipótese	30
1.5	Percurso Metodológico	30
1.6	Estrutura do Trabalho	31
2	FUNDAMENTAÇÃO TEÓRICA	33
2.1	Ensino de Engenharia de Software	34
2.1.1.	Modelagem e Análise de Software	37
2.2	Perspectiva Neopiagetiana e Taxonomia SOLO	39
2.3	Alinhamento Construtivo	44
2.4	Abstração, Modelagem e Representação	46
2.5	Técnica OC2-RD2	55
2.5.1.	Planos Organizacionais OC2-RD2	59
3	METODOLOGIA DE PESQUISA	64
3.1	Características Metodológicas	64
3.2	Construção do Procedimento de Pesquisa	69
3.3	Aplicação da Entrevista e Caracterização da Amostra	72
4	RESULTADOS DA PESQUISA	76
4.1	Análise dos Resultados sobre Desenvolvimento Cognitivo	76
4.2	Análise dos Resultados sobre Alinhamento Construtivo	77
4.3	Análise dos Resultados sobre Padrões de Abstração	85
4.4	Análise dos Resultados sobre Itens de Conhecimento e Erros de Abstração	89
4.5	Análise dos Resultados sobre o Modelo Pedagógico Proposto	93
4.6	Validade e Confiabilidade Qualitativa dos Resultados	95
5	MODELO PEDAGÓGICO PROPOSTO	99
5.1	Caracterização do Modelo Pedagógico	99
5.2	Construção do Modelo	102
5.2.1.	Correlação entre o Alinhamento Construtivo e a Técnica OC2-RD2	103
5.2.2.	Concepção dos Módulos de Alinhamento Construtivo	105
5.2.3.	Concepção do Módulo de Modelos Ontológicos	106
5.2.4.	Concepção do Módulo de Taxonomia SOLO	108
5.2.5.	Concepção do Módulo de Padrões de Abstração	109
5.2.6.	Criação dos Planos Ontológicos	110

5.2.7. Mapeamento dos Planos Ontológicos para Desempenhos Cognitivos de Abstração	115
5.2.8. Design Arquitetural do Modelo.....	116
5.3 Validação do Modelo	124
5.3.1. Procedimento de Aplicação do Modelo	124
5.3.2. Cenários e Exemplos de Aplicação do MPO_MAS	127
6 CONCLUSÕES.....	137
6.1 Considerações Finais	138
6.2 Limitações da Pesquisa.....	139
6.3 Sugestões de Trabalhos Futuros.....	140
REFERÊNCIAS	143
APÊNDICE A – Procedimento de Pesquisa	149
APÊNDICE B – Termo de Consentimento Livre e Esclarecido	162
ANEXO A – Representações de Apoio para o Roteiro de Entrevista.....	164
ANEXO B – Parecer de Mérito Acadêmico	172
ANEXO C – Plano da Disciplina “A”	173
ANEXO D – Plano da Disciplina “B”	174
ANEXO E – Plano da Disciplina “C”	176
ANEXO F – Plano da Disciplina “D”	178

1 INTRODUÇÃO

“Le temps est un grand professeur, mais malheureusement il tue tous ses élèves.”

(Hector Berlioz)

Afirmar que software está cada vez mais profundamente ligado ao cotidiano das pessoas ao redor do mundo é uma narrativa recorrente, mas ao mesmo tempo, verossímil. Aliás, é bastante difícil pensar no presente e imaginar o futuro sem a presença desse objeto artificial.

Os produtos de software estão amplamente presentes em diversas áreas de conhecimento. Pressman e Maxim (2015) listam sete grandes campos de aplicação de software, todos eles com desafios contínuos para a engenharia de software: a) sistema de software; b) software de aplicação; c) software científico e de engenharia; d) software embarcado; e) software para linha de produto; f) aplicações Web e aplicativos móveis e; g) software de inteligência artificial.

Dentre as diversas definições encontradas na literatura para o termo software, uma define que “Software é tanto um produto quanto um veículo que entrega um produto.” (PRESSMAN; MAXIM, 2015, p. 3). Em outras palavras, esses mesmos autores afirmam:

Software é: (1) instruções (programas de computador) que, quando executadas, fornecem os recursos, funções e desempenho desejados; (2) estruturas de dados que permitem que os programas manipulem informações adequadamente e (3) informações descritivas em ambos os formatos impresso e virtual que descrevem a operação e o uso dos programas. (PRESSMAN; MAXIM, 2015, p. 4).

Outra definição similar encontrada é: “O termo software se refere a um programa e a todas as informações e materiais associados necessários para dar suporte à sua instalação, operação, correção e aprimoramento.” (HUMPHREY, 1988, p. 82). Esse significado está em consonância à visão de Brooks Jr. (1995), a qual estabelece que software é um produto de programação, em outros termos, enquanto um programa é um conjunto de instruções que pode ser executado em um computador, um produto de programação é uma coleção de programas com

todos os itens necessários para torná-lo inteligível, utilizável e extensível.

Dado que software é um produto de programação, não se pode ignorar o fato de que esse objeto é uma entidade abstrata a qual possui características próprias que o diferencia de outros produtos. Com base na distinção feita por Aristóteles¹ entre propriedades essenciais e acidentais de uma coisa, na qual um acidente não afeta sua essência, Brooks Jr. (1987; 1995), no contexto de desenvolvimento de software, define que a essência se refere às dificuldades inerentes da natureza do software, como complexidade, conformidade, mutabilidade e invisibilidade, e que os acidentes estão relacionados às atividades de representação do software.

A complexidade é uma propriedade intrínseca de software frequentemente evidenciada na literatura. Para Bruegge e Dutoit (2009), os sistemas de software são criações complexas, dado que executam muitas funções e são construídos para atingir muitos objetivos diferentes e várias vezes conflitantes. Na ótica de Brooks Jr. (1987; 1995), a construção de software envolve tanto tarefas essenciais as quais constituem a modelagem de estruturas conceituais complexas que compõem a entidade abstrata de software, quanto tarefas acidentais as quais buscam representar tal entidade abstrata em linguagens de programação ou em outras tecnologias de software. De acordo com Parnas (2012), há uma busca incessante por uma “bala de prata” (*silver bullet*) para desenvolvedores de software, entretanto, quem busca um caminho fácil para a qualidade do software certamente ficará desapontado, dado que software moderno é extremamente complexo e mesmo o menor erro pode ter efeitos consideráveis. Ainda para este último autor, a única forma de garantir que o software seja confiável é com trabalho árduo e disciplinado.

Diante dos obstáculos a serem superados no que tange ao desenvolvimento de aplicações de software, a engenharia de software é considerada uma disciplina-chave tanto para o desenvolvimento de sistemas complexos de qualidade quanto para a evolução contínua desses sistemas ao longo do tempo.

Muitas são as definições sobre engenharia de software encontradas na literatura, das quais duas foram escolhidas para este trabalho, a de Humphrey

1 Acidente (filosofia). Disponível em [https://en.wikipedia.org/wiki/Accident_\(philosophy\)](https://en.wikipedia.org/wiki/Accident_(philosophy)).

(1988) e a de Pressman e Maxim (2015).

“Engenharia de software refere-se à aplicação disciplinada de princípios e métodos de engenharia, científicos e matemáticos para a produção econômica de software de qualidade.” (HUMPHREY, 1988, p. 82).

“A engenharia de software engloba um processo, uma coleção de métodos (prática) e uma série de ferramentas que permitem aos profissionais construir software de computador de alta qualidade.” (PRESSMAN; MAXIM, 2015, p. 14).

Sobre o papel da engenharia de software, Bruegge e Dutoit (2009) complementam que a engenharia de software é formada por um conjunto de quatro atividades amplas: a) atividade de modelagem; b) atividade de resolução de problemas; c) atividade de aquisição de conhecimento e; d) atividade orientada pela lógica.

Como seres desejantes, as pessoas buscam usar produtos de software em qualquer tempo e espaço. Isso significa que elas desejam um produto cada vez mais onipresente, porém essa ubiquidade só vai existir se a solução de software reunir atributos de qualidade necessários para atender as muitas e variadas exigências do mundo real.

Pressman e Maxim (2015) consideram que construir software para atender as demandas do século XXI é um grande desafio a ser enfrentado. Nos últimos anos, o software se transformou em um objeto altamente integrado em quase todos os aspectos de nossas vidas e, conseqüentemente, o número de pessoas interessadas nos recursos e funções fornecidos por um software aplicativo específico cresceu consideravelmente.

As propriedades específicas dos produtos de software assim como a demanda crescente por estes são realçadas nas Diretrizes Curriculares Nacionais para os cursos de graduação em Computação (MEC, 2016).

[...] O software desempenha um papel central em quase todos os aspectos da vida cotidiana, no governo, bancos e finanças, educação, transporte, entretenimento, medicina, agricultura, indústria e direito, entre outros. Softwares, inclusive, mantêm funcionando os vários serviços eletrônicos e programas sociais de larga escala dos governos, o fornecimento de energia elétrica, as redes de telecomunicações, os serviços de transporte aéreo, os caixas eletrônicos dos bancos, os cartões de crédito, as bolsas de valores e mercadorias, e muito mais. Os produtos de software têm ajudado a sociedade

quanto à eficiência e à produtividade. Eles permitem solucionar problemas de forma mais eficaz e fornecem um ambiente muitas vezes, mais seguro, mais flexível e mais aberto. Os produtos de software estão entre os mais complexos dos sistemas artificiais, e software, por sua própria natureza, tem ainda propriedades essenciais intrínsecas (por exemplo, a complexidade, a invisibilidade e a mutabilidade), que não são fáceis de serem dominadas. (MEC, 2016, p. 3-4).

Conforme Wirth (2008), o rápido crescimento do poder computacional tornou possível aplicar a computação a tarefas cada vez mais complicadas, aumentando drasticamente as demandas dos engenheiros de software; em contrapartida, os programas e sistemas se tornaram complexos e quase impossíveis de serem entendidos em sua completude. Esse autor ainda ressalta que “Nossas limitações não são mais dadas por hardware lento, mas por nossa própria capacidade intelectual.” (WIRTH, 2008, p. 1).

Ainda segundo Wirth (2008), o crescimento da complexidade não é algo para ser admirado, mas sim combatido sempre que possível, desse modo, os programadores devem ter tempo e respeito pelo trabalho de alta qualidade. Este autor ainda faz um chamado: “Vamos embarcar em um esforço global para evitar que o software se torne conhecido como *softwaste!*”² (WIRTH, 2008, p. 8).

A formação acadêmica em cursos da área computacional requer conhecimentos, que vão desde fundamentos de computação, tais como algoritmos e lógica de programação, matemática discreta e estruturas de dados, passando por técnicas de computação, incluindo engenharia de software, até alcançar as tão desejadas tecnologias de software.

De antemão, é importante deixar claro que o foco deste trabalho está no ensino de engenharia de software, porém vale frisar que essa disciplina é considerada interdisciplinar, integrando assim tanto conteúdos de outros componentes curriculares de computação, como também de outras áreas de conhecimento.

A engenharia de software emprega tanto conceitos que são exclusivos da natureza do software e de seu desenvolvimento, como também extrai outros de uma variedade de disciplinas de referência. Os alunos devem estar cientes desses

² Termo utilizado para significar “software inútil”.

conceitos e de suas limitações, quer sejam inerentes ou decorrentes da sua adaptação à engenharia de software. Os alunos precisam ser capazes de avaliar e refletir sobre os processos seguidos, bem como no que concerne as soluções produzidas. No que tange ao conhecimento técnico, um estudante de engenharia de software deve demonstrar compreensão e aplicar teorias, modelos e técnicas apropriadas que fornecem uma base para identificação e análise de problema, projeto de software, implementação, verificação e documentação (ACM; IEEE, 2014).

O ensino de engenharia de software está presente na maioria dos cursos de computação mundo afora. Em âmbito nacional, a engenharia de software está presente, como disciplina, em praticamente todos os currículos dos cursos de computação. A formação de um profissional em engenharia de software requer conhecimentos sólidos nas áreas de computação, matemática e processos de produção. A formação em computação demanda conhecimentos teóricos e de práticas de técnicas como análise, modelagem, programação e verificação, além de conhecimentos das tecnologias utilizadas para implementação e implantação de software (MEC, 2016; ZORZO *et al.*, 2017).

Já se tem ciência de que o componente curricular de engenharia de software oferece numerosos e diversos conteúdos para suportar o ciclo de vida de desenvolvimento de software. Não obstante, é meritório também se preocupar com as formas de como esses tópicos devem ser ensinados de maneira efetiva e duradoura.

A busca pelo aprendizado de temas como os listados em ACM e IEEE (2014), MEC (2016) e ZORZO *et al.* (2017) naturalmente vai revelar dificuldades cognitivas do aprendiz durante sua trajetória acadêmica, muitas vezes resultando em reprovação, desistência e cancelamento de disciplinas e, até mesmo, na evasão do curso. Giraffa e Mora (2013) relatam que a evasão nas disciplinas dos níveis iniciais dos cursos de computação têm sido uma situação de impasse e preocupação tanto para professores como também para a sociedade em geral.

Na verdade, as dificuldades de aprendizagem em cursos superiores de computação têm sido uma questão de muita atenção para as Instituições de Ensino Superior (IES), aliás bastante debatida por pesquisadores, professores e demais interessados, em eventos propícios realizados frequentemente.

Em território nacional, as discussões sobre o ensino de computação em

nível superior acontecem em diversos eventos, muitos deles organizados pela Sociedade Brasileira de Computação (SBC)³, tais como: o Workshop sobre Educação em Computação (WEI), o qual ocorre junto ao Congresso da Sociedade Brasileira da Computação (CSBC); o Fórum de Ensino de Engenharia de Software (FEES), evento adjacente ao Congresso Brasileiro de Software (CBSOFT); o Fórum de Educação em Sistemas de Informação (FESI), que acontece junto ao Simpósio Brasileiro de Sistemas de Informação (SBSI); como também em debates sobre ensino de computação na Educação Básica ou nas Licenciaturas em Computação cujas discussões acontecem no Congresso Brasileiro de Informática na Educação (CBIE) (ZORZO *et al.*, 2017).

Vale destacar que esses eventos possuem trilhas dedicadas à educação em computação, nos quais pesquisadores compartilham suas experiências em relação ao ensino de engenharia de software vivenciadas em ambientes de aprendizagem. Muitos professores têm aplicado em suas aulas numerosos e variados métodos e ferramentas de ensino dentre outros recursos pedagógicos que busquem enfatizar uma pedagogia ativa de aprendizagem. Aliás, pesquisas recentes mostram que o rol utilizado é relativamente extenso, por exemplo:

- a) O mapeamento sistemático da literatura de Santos, Rocha e Perkusich (2021) exhibe resultados globais a respeito da utilização de ferramentas e metodologias com o propósito de melhorar o processo de ensino e aprendizagem em engenharia de software;
- b) O mapeamento sistemático da literatura de Cunha *et al.* (2018) compartilha artigos com propostas de jogos, ferramentas, metodologias de ensino e integração entre disciplinas para subsidiar o ensino de engenharia de software no Brasil;
- c) A revisão sistemática da literatura de Lima *et al.* (2019) revela trabalhos sobre a utilização de metodologias ativas no ensino de engenharia de software em território nacional;
- d) O *survey* de Lima *et al.* (2020) denota informações referentes à adoção de metodologias ativas como forma de reduzir os desafios do ensino de engenharia de software em instituições brasileiras.

³ <https://www.sbc.org.br/>

O estudo supracitado de Lima *et al.* (2020) ainda identifica que os maiores problemas da educação em engenharia de software são: a) dificuldade no aprendizado; b) desmotivação e desinteresse; c) ausência das habilidades interpessoais; d) baixa aplicação da teoria em projetos reais; e) alta complexidade dos conteúdos abordados; f) ausência de atividades discursivas e; g) baixa relação entre o professor e o aluno. Esse estudo também apresenta uma lista de metodologias ativas mais adotadas nas aulas, realçando que aprendizagem baseada em projetos, aprendizagem baseada em problemas e as discussões em classe são as mais utilizadas. A pesquisa também identifica que as motivações que mais justificam a adoção de metodologias ativas no ensino de engenharia de software são: a) o processo de ensino e aprendizagem deve ser baseado na resolução de problemas e desafios; b) os conteúdos podem ser aplicados através de projetos práticos; c) aumento do engajamento dos discentes; d) maior orientação e comprometimento com os discentes e; e) mudança de paradigma e didática a ser adotada. Os autores ainda ressaltam que metodologias dessa natureza têm sido utilizadas no processo do ensino como forma de promover uma formação inovadora, humanística, crítica e reflexiva.

Diante do relatado, os professores têm buscado caminhos diversos para encontrar métodos de ensino direcionados para o ensino de computação com o propósito de instituir uma pedagogia ativa de aprendizagem. Nesse percurso, surge a técnica de narrativas OC2-RD2 para auxiliar no planejamento e condução de disciplinas e aulas da área computacional.

A técnica OC2-RD2 (Objetivo-Contratempo-Catástrofe-Reação-Dilema-Decisão) se baseia em estruturas narrativas particulares por meio de fábulas cujo propósito é minimizar os efeitos negativos de aprendizagem em cursos de computação, muitas vezes causados por conceitos matemáticos e lógicos (VEGA, 2016; VEGA, 2018; BUTTIGNON, 2020).

O repertório de engenharia de software é vasto, complexo e repleto de nuances, carecendo de uma conjuntura para um melhor entendimento dos conceitos concretos e abstratos envolvidos. Nesse sentido, essa disciplina pode ser contextualizada como um processo abstrato de desenvolvimento de software, em outros termos, como um conjunto de atividades de Análise, *Design*, Implementação e Teste (ADIT) de um produto de programação. No decurso deste processo, abstrações e refinamentos são realizados sucessivamente. Isso

significa que os estudantes precisam usar abstrações, construir modelos mentais, e saber representá-los adequadamente nos muitos e diversos artefatos, porém, a variedade de propósitos, formas e níveis de representação de modelos abstratos do domínio do problema (perspectiva de análise), assim como da solução de software (perspectiva de *design*), não são facilmente compreendidos pela maioria dos aprendizes.

Santos, Rocha e Perkusich (2021) consideram que ensinar engenharia de software é um desafio, visto que se trata de uma disciplina composta essencialmente por conteúdos fortemente teóricos e de difícil demonstração e, ressaltam que os estudantes possuem dificuldades em assimilar os conhecimentos apresentados em sala de aula por várias razões, sobretudo, pela própria natureza dos temas abordados, grande carga teórica, por serem mais abstratos e pouco demonstráveis na prática.

Segundo Hazzan (2002, 2007), as características intrínsecas à natureza do software afetam o processo de ensino e aprendizagem de engenharia de software, posto que ao lidar com uma entidade intangível, o processo de abstração se torna ainda mais relevante. Além do mais, a autora ressalta que o engenheiro de software, diferentemente dos profissionais de outras engenharias, vai estar envolvido na própria construção do produto, portanto, a noção de abstração deve ser ainda mais valorizada em seu aprendizado, contudo, a abstração não é uma capacidade cognitiva fácil de ser abordada, ainda mais diante da construção de um sistema artificial e complexo, como é o caso do software.

Vega (2016) destaca que há uma relação direta entre a dificuldade de se pensar abstratamente e o afastamento de estudantes em cursos superiores de computação, e salienta que o processo (mental) de abstração é uma maneira de poder lidar com as dificuldades inerentes no que tange à natureza do software, apontadas por Brooks Jr. (1987;1995).

Liskov e Guttag (2000) consideram que o processo de abstração pode ser visto como uma aplicação de um processo muitos-para-um, dado que tal processo permite esquecer a informação e, por consequência, tratar coisas diferentes como se fossem iguais, com a esperança de simplificar a análise, separando atributos relevantes daqueles que não são, entretanto, é fundamental lembrar que essa relevância vai depender do contexto.

Para Wirfs-Brock, Wilkerson e Wiener (1990), o processo de abstração é

psicologicamente natural e necessário, dado que a abstração é crucial para as pessoas entenderem o mundo, portanto, a abstração é essencial para o funcionamento de uma mente humana normal, como também uma ferramenta imensamente poderosa para lidar com a complexidade.

É fato que o conceito de abstração é recorrente em cursos de computação, sendo considerado uma importante ferramenta no desenvolvimento de software (HABERMAN; MULLER, 2008; ROBERTS, 2009). Em poucas palavras, o software é construído sobre abstrações (JACKSON, 2012). A abstração é a chave para se projetar um bom software (WIRFS-BROCK; WILKERSON; WIENER, 1990).

A engenharia de software, por sua vez, utiliza propósitos, formas e níveis de abstração distintos para explicar, modelar, especificar, raciocinar ou resolver problemas. Posto isto, o aprendizado dessa disciplina requer a aplicação constante de abstração, modelagem e representação.

Kramer (2003; 2007) considera a abstração uma habilidade-chave para engenheiros de software, melhor dizendo, uma capacidade essencial durante a fase de engenharia de requisitos porque permite capturar os aspectos críticos do ambiente de negócio e do sistema requerido e, ao mesmo tempo, ignorar o que não é importante. Na fase de *design*, a abstração é necessária para articular a arquitetura de software e as funcionalidades dos componentes que satisfaçam os requisitos funcionais e não funcionais. Mesmo na fase de implementação, as abstrações de dados e classes são utilizadas para generalizar soluções. Por outro lado, o autor considera que o processo de abstração é extremamente difícil de ensinar e aprender.

Durante a revisão de literatura, foi constatado que outros trabalhos também têm se preocupado com as dificuldades encontradas pelos estudantes nas aulas de engenharia de software em relação ao processo de abstração, modelagem e representação, os quais são:

- a) O trabalho de Ferreira *et al.* (2018) identifica dificuldades no processo de ensino e aprendizagem de engenharia de software e destaca que a criação de abstrações específicas é uma das atribuições. Esse estudo ainda ressalta que é necessário criar abstrações em todas as áreas da computação, contudo, em engenharia de software, essas abstrações se tornam mais imprescindíveis, visto que os futuros profissionais precisam

especificar e modelar necessidades de negócio em soluções computacionais;

- b) O estudo de Chourio *et al.* (2019) evidencia que os alunos têm problemas com a abstração de conceitos a serem representados por meio da modelagem. Esse trabalho ainda recomenda aos professores e profissionais de engenharia de software que procurem dar foco na idealização, criação ou pesquisa de técnicas dedicadas especificamente aos erros de abstração;
- c) O trabalho de Oliveira Jr. *et al.* (2021) aponta que a falta de maturidade para se compreender o processo de abstração é um dos desafios para o ensino de diagramas UML;
- d) O estudo de Cooper *et al.* (2005) constata que os estudantes têm dificuldades em progredir de um diagrama UML para outro, de compreender as relações entre os diferentes tipos de diagramas e, principalmente, o relacionamento entre informações do mesmo tipo de diagrama em níveis distintos de abstração;
- e) A pesquisa de Hazzan e Kramer (2016) conduziu um estudo com especialistas em pesquisa e ensino para saber se as habilidades de abstração poderiam ser avaliadas. O estudo entende que o uso da abstração precisa ser claro e que esse tema deve ser considerado e discutido explicitamente por instrutores, alunos e profissionais da área. Os autores ainda veem essa perspectiva como um ponto de partida para a discussão das habilidades de abstração e convidam a comunidade da computação a se juntar e contribuir com tal discussão.

Mesmo que o tema da abstração seja complexo de ser ensinado e aprendido, é importante para os acadêmicos de cursos da área computacional desenvolverem essa habilidade cognitiva.

Roberts (2009) destaca em seu estudo a diferença entre alunos que demonstram pensamento abstrato daqueles que não manifestam essa maneira de pensar e ressalta o interesse de pesquisas que buscam reunir evidências de relação entre a capacidade de pensamento abstrato e o desempenho dos estudantes de cursos da área computacional, inclusive a autora presume haver uma ligação mais direta entre conceitos abstratos de Jean Piaget e o uso de

abstração em computação.

Kramer (2007) acredita que a principal diferença de compreensão e desempenho entre estudantes de engenharia de software esteja na noção de abstração. “Acredito que a chave está na abstração: a capacidade de realizar o pensamento abstrato e exibir habilidades de abstração.” (KRAMER, 2007, p. 38). Em seu trabalho, o autor manifesta certas indagações sobre as habilidades de abstração, sendo:

- a) Os poderes de abstração de nossos alunos dependem de seu desenvolvimento cognitivo?
- b) Podemos melhorar suas habilidades e, em caso afirmativo, como?
- c) É possível ensinar pensamento abstrato e habilidades de abstração?

Ao apresentar tais questionamentos, Kramer (2007) também traz à tona a teoria piagetiana, a qual fornece as bases para uma compreensão do desenvolvimento cognitivo, desde bebês até a idade adulta, na tentativa de se encontrar respostas a respeito da capacidade de abstração.

Embora haja críticas sobre a maneira como Piaget tenha conduzido sua pesquisa e derivou sua teoria, há um suporte geral para suas ideias subjacentes. Nesse sentido, há estudos que sustentam a hipótese de que as crianças progridem nos três primeiros estágios de desenvolvimento, contudo, parece que nem todos os adolescentes progridem para o estágio de operações formais à medida que amadurecem biologicamente, apenas por meio da interação normal com o ambiente. Para o estágio operacional formal, parece que a maturidade determina a base, porém é preciso um ambiente especial para que a maioria dos adolescentes e adultos atinjam esse estágio cognitivo (HUIT; HUMMEL, 2003; LUTZ; HUITT, 2004; KRAMER, 2007). O aprendizado, além de biológico, também é psicológico e social (PIAGET, 2000).

Kramer (2007) presume que o desempenho cognitivo para o estágio de operações formais pode ser melhorado mediante a criação de um ambiente educacional correto e recomenda para tal, o uso de técnicas de ensino que possibilitem aos estudantes a oportunidade de explorar questões hipotéticas, encorajando-os a explicar como eles resolveriam problemas, além do enfoque no ensino de conceitos mais amplos, ao invés de se concentrar em apenas fatos isolados.

Embora a maior parte dos trabalhos de Jean Piaget lide com a aprendizagem de crianças e adolescentes, os princípios piagetianos podem ser estendidos à aprendizagem de adultos, como é o caso da teoria neopiagetiana que expande a visão da teoria piagetiana clássica em diversas áreas, onde uma das perspectivas é a que os estágios de desenvolvimento cognitivo vão além da infância e continuam na fase adulta, quando aplicados em novos contextos cognitivos. (BIGGS; COLLIS, 1982; BIGGS, 1992; MORRA *et al.*, 2012).

Outros trabalhos relacionados ao presente estudo também têm explorado abordagens neopiagetianas no ensino de computação, como o de Lister (2011), Corney *et al.* (2012), Gluga *et al.* (2012), os quais dão enfoque no ensino de programação, e o estudo de Falkner, Vivian e Falkner (2013) que explora a perspectiva neopiagetiana no ensino de desenvolvimento de software. Cabe destacar que todos esses trabalhos ressaltam a capacidade de abstração e a maneira como os estudantes são classificados nos estágios de desenvolvimento neopiagetianos conforme o desempenho alcançado.

Um caminho para se explorar as implicações da teoria neopiagetiana decorre do uso de uma taxonomia cognitiva denominada Structure of the Observed Learning Outcome (SOLO), uma taxonomia cognitiva caracterizada por um sistema de categorias cujo propósito é analisar a estrutura de respostas dos alunos, buscando identificar o tipo de pensamento demonstrado em uma situação específica (BIGGS; COLLIS, 1982; BIGGS, 1992).

Para Mol e Matos (2019), os conhecimentos sobre a taxonomia SOLO podem orientar o professor em diversas dimensões da sua prática pedagógica: o que ensinar, como ensinar e como avaliar, dado que havendo um referencial desse tipo à sua disposição, pode ajudá-lo a preparar suas aulas de forma mais consciente.

A busca por um planejamento de ensino de engenharia de software com base em resultados a serem atingidos tanto por professores quanto pelos alunos é fundamental para se propor atividades de ensino e aprendizagem e tarefas de avaliação. Nessa direção, surge a possibilidade da aplicação do alinhamento construtivo.

O alinhamento construtivo tem dois aspectos: construtivo no sentido que se refere a ideia de que o aluno constrói seu conhecimento com atividades de aprendizagem relevantes e alinhamento que se refere ao que o professor faz,

criando um ambiente de aprendizagem para apoiar as atividades, fazendo com que o aluno alcance os resultados desejados (BIGGS; TANG, 2011).

Em suma, diante do exposto até aqui, a investigação de abordagens pedagógicas reunindo conteúdo e cognição, a preocupação com a melhoria do processo de abstração, modelagem e representação em ambientes de aprendizagem e a oportunidade de contribuir com a comunidade acadêmica da área computacional por meio da proposição de um modelo pedagógico foram fatores que motivaram a pesquisa sobre o ensino de engenharia de software neste trabalho.

Este capítulo prossegue com as justificativas e a questão de pesquisa, o objetivo geral do trabalho, os objetivos específicos, a hipótese, o percurso metodológico seguido e a estrutura desta tese.

1.1 Justificativas e Questão de Pesquisa

Considerando o objeto de estudo contextualizado no preâmbulo deste capítulo, as justificativas para a realização desta pesquisa podem ser resumidas desta maneira:

- 1- Visto que a sociedade está cada vez mais exigente no que se refere ao uso de sistemas de software de qualidade em diversas áreas de conhecimento, a qualificação das pessoas responsáveis por construir e manter esses sistemas ao longo do tempo vem à tona. Diante dessa realidade, a preocupação com a educação dos estudantes em cursos de computação acarreta a busca por recursos pedagógicos que possam favorecer o processo educacional de engenharia de software em cursos de computação;
- 2- Uma vez que o processo de abstração é uma habilidade cognitiva importante para os estudantes de computação, este estudo considera que o tema em foco, juntamente com modelagem e representação, precisam ser evidenciados e discutidos pelos professores para fazer parte do planejamento de ensino de engenharia de software;
- 3- Dado que os estágios de desenvolvimento cognitivo podem continuar na fase adulta quando aplicados em novos contextos cognitivos, esta pesquisa procura em uma perspectiva neopiagetiana a possibilidade de o estudante atingir níveis de complexidade cognitiva crescente, em

outros termos, de eles conseguirem obter progressão dos modos de funcionamento no contexto do ensino e aprendizagem em engenharia de software;

- 4- Levando em conta que o ensino com base em resultados a serem atingidos pelos professores e alunos é fundamental para se propor atividades de ensino e aprendizagem e tarefas de avaliação, este trabalho recorre a uma abordagem neopiagetiana mais uma vez. O ensino alinhado construtivamente, apoiado pelos verbos de aprendizagem da taxonomia cognitiva SOLO, incentivado por um processo de reflexão transformadora e auxiliado por outros professores da mesma área de conhecimento, podem resultar na implementação do alinhamento construtivo no ensino de engenharia de software;
- 5- Em virtude de a técnica de narrativas OC2-RD2 ser um recurso pedagógico para cursos de computação, essa técnica pode ser adotada para apoiar o planejamento de ensino de engenharia de software. Para tal, este estudo busca suporte nos planos organizacionais da técnica OC2-RD2, especialmente na elaboração do Plano de Conteúdos.

A descoberta de uma lacuna interpretativa no contexto apresentado, quiçá aplicativa, mediante a revisão de literatura e as justificativas preditas, evidenciam a pertinência e oportunidade deste estudo, o qual busca investigar a seguinte questão de pesquisa.

Como apoiar o professor no planejamento de ensino de engenharia de software a fim de permitir ao estudante condições de atingir níveis de complexidade cognitiva crescente na construção de conhecimento concreto e abstrato?

1.2 Objetivo Geral

A fim de responder à questão de pesquisa, formulada e contextualizada na seção anterior, o objetivo geral deste trabalho é:

Propor um modelo pedagógico de ensino de engenharia de software construído a partir da correlação entre o alinhamento construtivo e os planos organizacionais da técnica OC2-RD2, para auxiliar o professor no planejamento de disciplinas e aulas de engenharia de software.

1.3 Objetivos Específicos

Para atingir o objetivo geral descrito na seção anterior, cinco objetivos específicos são estabelecidos, os quais são:

- 1- Correlacionar as etapas substanciais do alinhamento construtivo e os planos organizacionais da técnica OC2-RD2, sobretudo o plano de conteúdos, levando em consideração os níveis e verbos de aprendizagem da taxonomia SOLO, a qual é vista como uma abordagem neopiagetiana;
- 2- Conceber uma ontologia de domínio para uma das áreas de conhecimento de engenharia de software, melhor dizendo, para a área de modelagem e análise de software, constituída por um conjunto de planos ontológicos temáticos e reutilizáveis;
- 3- Construir uma arquitetura pedagógica para o modelo proposto, buscando representar um metamodelo ontológico de modelagem e análise de software;
- 4- Incorporar um conjunto de padrões de abstração no modelo pedagógico, buscando mapear os planos ontológicos para desempenhos cognitivos, tentando contextualizar o ensino de abstração e, por consequência, fortalecer o pensamento abstrato;
- 5- Validar, ainda que de maneira preliminar, o modelo proposto em situações-exemplo, ou seja, em determinados planos de disciplinas selecionados.

1.4 Hipótese

Tendo como base à questão de pesquisa e os objetivos descritos nas seções anteriores, a hipótese considerada para esta pesquisa é:

A correlação entre as etapas do alinhamento construtivo e os planos organizacionais da técnica OC2-RD2 oportuniza a construção de um modelo pedagógico de ensino de engenharia de software, com o intuito de auxiliar o professor no planejamento de ensino de engenharia de software.

1.5 Percurso Metodológico

Em linhas gerais, este trabalho é de abordagem qualitativa e propósito de estudo exploratório, aplicando a entrevista semiestruturada como instrumento de

coleta de dados e a análise indutiva como estratégia de investigação. Maiores detalhes sobre a metodologia de pesquisa empregada nesta pesquisa são encontrados no Capítulo 3.

A fim de responder à questão de pesquisa e atingir os objetivos estabelecidos, este estudo se apoia resumidamente no seguinte percurso metodológico:

- a) Procedimento de estudo a partir de revisão da literatura;
- b) Elaboração de um roteiro de entrevista semiestruturada como instrumento de coleta de dados;
- c) Aplicação do instrumento de pesquisa junto à amostra de participantes;
- d) Análise e interpretação qualitativa dos resultados a partir dos dados provenientes das entrevistas;
- e) Adoção de estratégias para assegurar a validade e confiabilidade dos resultados;
- f) Utilização dos resultados para construir a proposição do modelo pedagógico do presente trabalho.

1.6 Estrutura do Trabalho

O Capítulo 1 apresenta de forma introdutória o contexto desta tese, redige as justificativas, formula a questão de pesquisa, define o objetivo geral e os específicos do trabalho, identifica a hipótese, descreve o percurso metodológico empregado e organiza os capítulos do trabalho na presente estrutura.

O Capítulo 2 descreve a fundamentação teórica de sustentação para este estudo, resultante da revisão de literatura.

O Capítulo 3 caracteriza a metodologia de pesquisa empregada nesta tese, explana a construção do procedimento de coleta de dados, e descreve a aplicação de tal instrumento e a caracterização da amostra da pesquisa.

O Capítulo 4 analisa, interpreta e valida qualitativamente os resultados provenientes dos dados coletados via roteiro de entrevista semiestruturada.

O Capítulo 5 caracteriza o modelo pedagógico, explica o processo de construção da proposição, ilustra os módulos da arquitetura pedagógica e demonstra uma validação preliminar da proposta em determinados planos de disciplina.

Finalmente, o Capítulo 6 conclui o estudo em foco, faz as considerações finais, aponta as limitações da pesquisa e sugere trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

“Software is a place where dreams are planted and nightmares harvested, an abstract, mystical swamp where terrible demons compete with magical panaceas, a world of werewolves and silver bullets.”

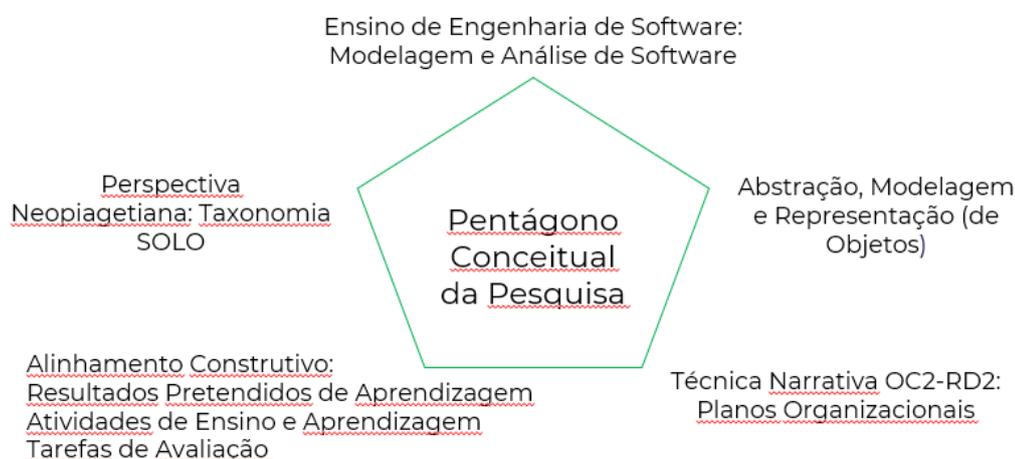
(Brad J. Cox)

Este capítulo descreve os principais conceitos dos assuntos adotados neste trabalho, sendo:

- a) Ensino de Engenharia de Software, especialmente a área de Modelagem e Análise de Software;
- b) Perspectiva Neopiagetiana, com ênfase na Taxonomia Solo;
- c) Alinhamentos Construtivo e suas etapas substanciais, isto é, Resultados Pretendidos de Aprendizagem, Atividades de Ensino e Aprendizagem e Tarefas de Avaliação;
- d) Abstração, Modelagem e Representação de Software, com enfoque em objetos;
- e) Técnica Narrativa OC2-RD2 e seus planos organizacionais.

A Figura 1 apresenta o recorte temático que delimita a fundamentação teórica desta pesquisa.

Figura 1 – Pentágono conceitual da pesquisa



Fonte: Elaborado pelo autor.

2.1 Ensino de Engenharia de Software

A computação não é apenas uma única área de estudo, mas sim uma família de áreas de estudo. Ao longo da década de 90, mudanças importantes na computação, tecnologia de comunicação e seus efeitos sociais levaram a mudanças consideráveis nessa família de disciplinas (ACM; IEEE, 2020), sendo:

- a) A engenharia da computação emergindo da engenharia elétrica;
- b) A ciência da computação evoluindo para uma disciplina acadêmica mais madura;
- c) A expansão dos sistemas de informação à medida que os computadores se tornaram a base dos processos organizacionais e dos ambientes de trabalho;
- d) A tecnologia da informação surge como uma nova disciplina que fomenta a construção e manutenção de infraestruturas informáticas;
- e) A engenharia de software emerge como uma disciplina baseada em ciência da computação e engenharia da computação.

Devido aos fundamentos teóricos, a Ciência da Computação (CC) é frequentemente vista como uma disciplina fundamental. No entanto, às vezes é erroneamente equiparada a toda a computação. Esse equívoco é compreensível, dado que as raízes teóricas da CC surgiram separadamente da tradição da engenharia dos primeiros dias da computação. Embora as ciências físicas sejam fundamentais e ofereçam bases teóricas para os campos da engenharia, nenhuma inclui a outra e cada uma tem uma identidade distinta bem compreendida. Por um lado, os cursos de CC enfatizam recursos computacionais abstratos, como abstração, complexidade e mudança evolutiva, tratados como temas recorrentes, mas por outro, também esses cursos compartilham assuntos comuns fortemente vinculados às atividades de programação e desenvolvimento de software. Chama a atenção de que cerca de 40% das horas de um curso de CC são voltadas para o ensino de algoritmos e complexidade, linguagens de programação, fundamentos de desenvolvimento de software e engenharia de software (ACM; IEEE, 2020). Nota-se a importância dada aos conteúdos de engenharia de software, mesmo quando se trata de um componente curricular de outros cursos da área computacional.

Com o panorama da educação em computação mudando significativamente no início dos anos 2000, a engenharia de software surgiu, inclusive, como um curso de graduação próprio.

Parnas (1999) já salientava que a crescente importância do software, combinada com o conhecimento sobre como construí-lo exigiria graduados que, como outros engenheiros, devem receber uma educação que se concentra em como projetar e fabricar produtos confiáveis, mas que também precisam saber projetar, construir, testar e manter produtos de software. Em suma, não poderia mais espremer o que já se sabia em alguns cursos de engenharia tradicional ou de ciência da computação. De forma sucinta, o autor salienta que os cientistas aprendem ciência e os métodos científicos necessários para estendê-la, e os engenheiros aprendem ciência e os métodos necessários para aplicá-la.

A Engenharia de Software (ES) comumente faz parte de outros cursos de graduação, como os de Engenharia da Computação (EC) e Ciência da Computação (CC). De qualquer forma, embora a ES esteja concentrada na criação de soluções baseadas em software, isso é muito mais do que programação, já que enfatiza o uso de práticas de desenvolvimento de software adequadas e a integração do rigor da engenharia com a capacidade de aplicar algoritmos avançados e estruturas de dados desenvolvidas em CC. O foco da ES está no projeto de sistemas de software confiáveis, seguros e utilizáveis. (ACM; IEEE, 2020).

No que concerne os cursos de ES, é importante frisar que existem Diretrizes Curriculares para Programas de Graduação em Engenharia de Software⁴, um volume da série de currículos de computação e resultado de uma força tarefa conjunta sobre currículos de computação envolvendo a IEEE Computer Society (IEEE-CS) e a Association for Computing Machinery (ACM). O objetivo fundamental desse volume é oferecer às instituições acadêmicas e agências de credenciamento sobre o que deve constituir um ensino de graduação em engenharia de software. As recomendações do currículo de engenharia de software são importantes porque o número de novos programas de graduação em engenharia de software continua a crescer de forma constante e os processos de credenciamento para tais programas foram estabelecidos em vários países (ACM;

⁴ Tradução de Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering.

IEEE, 2014).

Ao mencionar tal currículo, é relevante saber que as duas principais contribuições desse documento são:

- a) Conhecimento de Educação em Engenharia de Software⁵ (SEEK): o que todo graduado em engenharia de software deve saber;
- b) Currículo: formas de ensinar esse conhecimento e as habilidades fundamentais à engenharia de software em diversos contextos.

Se tratando de SEEK, consta o conhecimento que os graduados em ES devem aprender, entretanto, é importante frisar que a forma como esse conhecimento deve ser ensinado pode ser tão importante quanto o que é ensinado. O termo conhecimento é usado para descrever todo o espectro de conteúdo para a disciplina, como informação, terminologia, artefatos, dados, papéis, métodos, modelos, procedimentos, técnicas, práticas, processos e literatura (ACM; IEEE, 2014).

O SEEK é organizado por áreas de conhecimento, cada qual representando uma subdisciplina particular, geralmente reconhecida como uma parte significativa do conhecimento de ES que um estudante de graduação deve conhecer. As áreas de conhecimento são elementos estruturais de alto nível usados para organizar, classificar e descrever o conhecimento de engenharia de software (ACM; IEEE, 2014). O Quadro 1 relaciona as 10 áreas de conhecimento que compõem o SEEK.

Quadro 1 – Áreas de Conhecimento do SEEK

Fundamentos de Computação	Design de Software
Fundamentos de Matemática e Engenharia	Verificação e Validação de Software
Prática Profissional	Processo de Software
Modelagem e Análise de Software	Qualidade de Software
Análise e Especificação de Requisitos	Segurança

Fonte: Adaptado de ACM e IEEE (2014, p. 28).

No que tange ao currículo, consta uma relação de 20 diretrizes a ser considerada tanto por aqueles que desenvolvem um currículo para um curso de

⁵ Tradução de software Engineering Education Knowledge (SEEK).

graduação de ES, como também por aqueles que ministram cursos individuais, isto é, componentes curriculares. Uma dessas diretrizes, especificamente a quarta (Curriculum Guideline 4) chamou a atenção deste estudo, pois estabelece que muitos conceitos, princípios e questões de engenharia de software devem ser ensinados como temas recorrentes ao longo do curso para ajudar os alunos a desenvolver uma mentalidade de engenharia de software, elencando explicitamente a temática de abstração, modelagem e representação. Tal diretriz ainda estabelece que esses temas devem ser ensinados de maneira distribuída em muitas disciplinas da grade curricular, onde as primeiras devem apresentar o conteúdo a ser ensinado, e os componentes curriculares subsequentes devem reforçar e expandir esses assuntos (ACM; IEEE, 2014).

No caso deste trabalho, o foco está na engenharia de software como curso individual, isto é, disciplina, uma vez que tal unidade geralmente faz parte da estrutura curricular da maioria dos cursos de graduação da área computacional, mesmo que às vezes, tal componente curricular tenha nomenclaturas distintas para determinados cursos superiores.

2.1.1. Modelagem e Análise de Software

Por conta do interesse deste estudo, se escolheu a área de conhecimento de Modelagem e Análise de Software (MAS), visto que se nota maior atenção sobre a importância da abstração em suas unidades e tópicos, conforme mostra o Quadro 2. Além disso, conteúdos de MAS são importantes para qualquer disciplina de ES, visto que “modelagem e análise podem ser consideradas conceitos centrais em qualquer disciplina de engenharia porque são essenciais para documentar e avaliar decisões e alternativas de projeto.” (ACM; IEEE, 2014, p. 31).

Quadro 2 – Unidades e tópicos da área de modelagem e análise de software

Unidades	Tópicos
Fundamentos de Modelagem (FM)	FM1: Princípios de modelagem (por exemplo, decomposição, abstração, generalização, projeção/visões, e uso de abordagens formais)
	FM2: Pré-condições, pós-condições, invariantes e design por contrato
	FM3: Introdução aos modelos matemáticos e notação formal
Tipos de Modelos	TM1: Modelagem de informações (por exemplo, modelo

(TM)	entidade-relacionamento e diagramas de classe)
	TM2: Modelagem comportamental (por exemplo, diagramas de estado, análise de casos de uso, diagramas de interação, modos de falha e análise de efeitos, e análise de árvore de falha)
	TM3: Modelagem arquitetural (por exemplo, padrões de arquitetura e diagramas de componente)
	TM4: Modelagem de domínio (por exemplo, abordagens de engenharia de domínio)
	TM5: Modelagem empresarial (por exemplo, processos de negócios, organizações, metas e fluxo de trabalho)
	TM6: Modelagem de sistemas embarcados (por exemplo, análise de escalonamento em tempo real e protocolos de interface)
Fundamentos de Análise (FA)	FA1: Análise da construção (por exemplo, completude, consistência, e robustez)
	FA2: Análise da corretude (por exemplo, análise estática, simulação, e verificação de modelo)
	FA3: Análise da confiabilidade (por exemplo, análise de modo de falha e árvores de falha)
	FA4: Análise formal (por exemplo, prova de teoremas)

Fonte: Adaptado de ACM e IEEE (2014, p. 31).

Outro motivo pela seleção de MAS é que essa área elenca tópicos que enfatizam a importância da abstração, como também da modelagem e representação. Isso não significa que as demais áreas de conhecimento não tratam esse tema como importante, aliás foi dito anteriormente que esses temas devem ser ensinados de maneira recorrente no decorrer do curso.

As ementas e objetivos de unidades curriculares de engenharia de software que englobam tópicos de MAS podem ser planejados para contemplar os temas de abstração, modelagem e representação com maior profundidade durante o ensino dos conteúdos previstos no plano da disciplina. Vale destacar que no decurso desta pesquisa, esses três temas serão frequentemente tratados de forma conjunta, conotando uma tríade de conhecimento fundamental no ensino de ES.

Outro argumento, mas não menos importante, que ampara a escolha de MAS neste trabalho, é a presença marcante do paradigma orientado a objetos nas ementas de componentes curriculares de ES, muitas vezes instanciando a

disciplina como uma Engenharia de Software Orientada a Objetos (ESOO), na qual se direciona conteúdos para serem ensinados com o suporte da modelagem e análise orientada a objetos, e conseqüentemente, refinamentos sucessivos de abstração, modelagem e representação.

A junção de atividades de ES com o paradigma orientado a objetos é bastante citada na literatura. Apenas como exemplo, Bruegge e Dutoit (2009), apresentam processos e artefatos de ESOO, a fim de lidar com a complexidade, construindo e validando modelos do domínio da aplicação ou do sistema, e destacam o papel relevante da modelagem orientada a objetos na construção de software.

2.2 Perspectiva Neopiagetiana e Taxonomia SOLO

A taxonomia cognitiva SOLO é uma perspectiva neopiagetiana (BIGGS; COLLIS 1982; BIGGS, 1992; MOL; MATOS, 2019).

A teoria piagetiana clássica compreende o desenvolvimento cognitivo como um processo em estágios discretos, uma estrutura lógica (*structure d'ensemble*) definindo cada estágio e regendo todas as performances realizadas dentro dele, porém, existem exceções ao desempenho típico de cada estágio, as quais são chamadas de decalagens (*décalages*). No contexto escolar, a decalagem é extremamente comum, já que um aluno pode parecer estar dando respostas “formais precoces” em matemática e “concretas precoces” em história, ou formal em matemática um dia e concretas no dia seguinte. Tais observações indicam claramente mudanças no aprendizado, desempenho ou motivação do aprendiz, e não exatamente no estágio cognitivo (BIGGS, 1992).

Bem como Piaget, os pesquisadores Biggs e Collis defendem a existência de estágios de desenvolvimento que admitem supostas estruturas cognitivas específicas. É verdade que todos eles concordam que, do nascimento à fase adulta, os sujeitos aprendem de formas que são típicas para suas idades, todavia, Biggs e Collis percorreram caminhos distintos de Piaget. A intenção de Biggs e Collis era analisar respostas de alunos com o objetivo de medir o desenvolvimento, semelhante ao que foi feito por Piaget, contudo, eles descobriram que era difícil identificar o estágio de desenvolvimento de um estudante por meio de respostas dadas para tarefas, uma vez que o aprendiz pode dar respostas que se encaixam em diferentes estágios, a depender da

disciplina, do conteúdo estudado, ou até mesmo do tempo em que tais respostas foram geradas. Por conta disso, Biggs e Collis alteraram o enfoque de análise das estruturas cognitivas do aprendiz para a resposta que ele fornecia em tarefas específicas. Desse modo, o ponto de partida da taxonomia SOLO é a qualidade da aprendizagem demonstrada em um certo momento (MOL; MATOS, 2019).

Sobre o aprendizado em si, é importante salientar o crescimento da complexidade da resposta dentro de um estágio. Biggs e Collis (1982) identificaram uma sequência consistente e hierárquica denominada ciclo de aprendizagem. Esses autores verificaram que o mesmo padrão de ciclo de aprendizagem é aplicável a uma variedade de tarefas, assim como em vários modos, em outras palavras, saber em que ponto um sujeito pode estar no ciclo de aprendizagem indica o quão longe a aprendizagem progrediu dentro de um determinado modo. Essas informações podem ser usadas para avaliar os resultados de aprendizagem, eis que surge a taxonomia cognitiva SOLO (Structure of the Observed Learning Outcome) ou, em português, "Estrutura dos Resultados de Aprendizagem Observados". Essa taxonomia pode ser usada para avaliar a qualidade da aprendizagem ou para estabelecer objetivos curriculares (BIGGS, 1992).

De acordo com Biggs e Collis (1982) e Biggs (1992), são cinco níveis básicos distintos no ciclo de aprendizagem, isto é, pré-estrutural, uniestrutural, multiestrutural, relacional e abstrato estendido. As respostas uniestruturais, multiestruturais e relacionais se enquadram em um determinado modo. As respostas pré-estruturais pertencem ao modo anterior, indicando que a aprendizagem está em um nível de abstração muito baixo para a tarefa em questão, enquanto as respostas abstratas estendidas estão em um nível de abstração que se estende para o próximo modo e se torna o nível uniestrutural desse próximo modo. O Quadro 3 descreve esses níveis em relação a um determinado modo, aqui chamado de modo alvo, salientando que o foco da aprendizagem está dentro do modo alvo, e seu progresso é indicado pelos níveis 2, 3 ou 4, descrevendo assim o ponto do ciclo de aprendizagem já atingido pelo aprendiz.

Quadro 3 – Modos e níveis da taxonomia SOLO

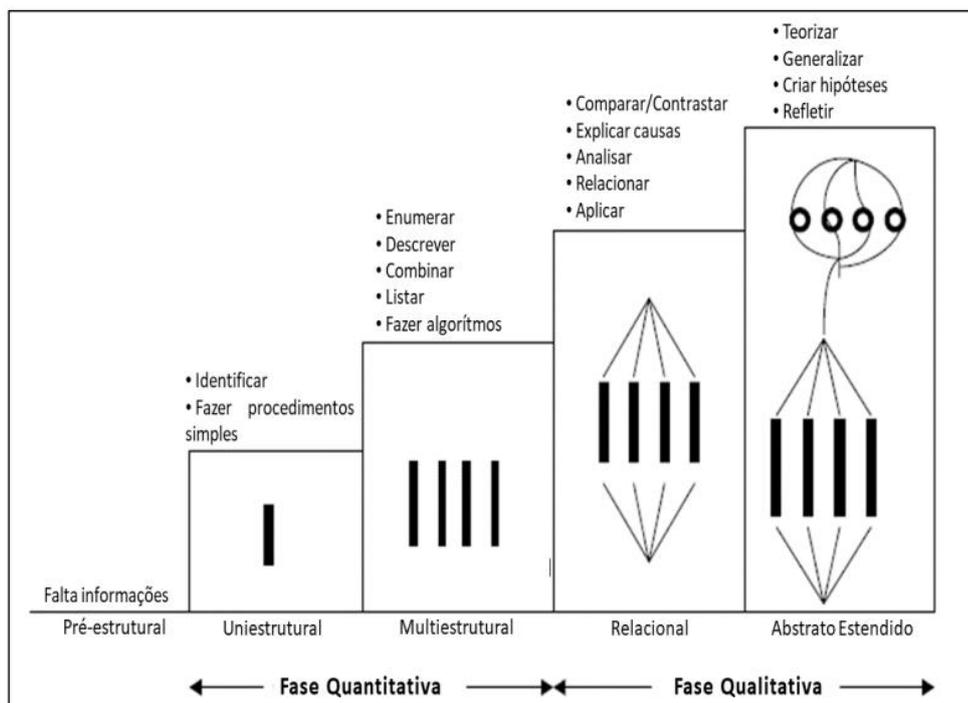
Modo	Nível Estrutural
Próximo	5. Abstrato Estendido. O estudante agora generaliza a estrutura para

	incorporar recursos novos e mais abstratos, representando um modo de operação novo e superior.
Alvo	4. Relacional. O estudante agora integra as partes umas às outras, de modo que o todo tenha uma estrutura e um significado coerentes. 3. Multiestrutural. O estudante aprende recursos cada vez mais relevantes ou corretos, mas não os integra. 2. Uniestrutural. O estudante se concentra no domínio relevante e pega um aspecto com o qual trabalhar.
Anterior	1. Pré-estrutural. Diante de uma tarefa, o estudante está distraído ou enganado por um aspecto irrelevante pertencente a um estágio ou modo anterior.

Fonte: Adaptado de Biggs (1992, p. 38).

Além dos modos e níveis SOLO, a presente taxonomia possui um conjunto de verbos de aprendizagem, os quais permitem um entendimento dos avanços e desenvolvimentos relacionados às tarefas. A Figura 2 exhibe uma hierarquia de verbos, agrupados em cinco níveis: Pré-estrutural, Uniestrutural, Multiestrutural, Relacional e Abstrato Estendido, onde cada nível (do Uniestrutural ao Abstrato Estendido) é composto por verbos que ajudam na elaboração de tarefas com foco nos resultados pretendidos de aprendizagem. Além disso, a taxonomia SOLO é organizada em duas fases, sendo uma quantitativa e outra qualitativa. A fase quantitativa se localiza num estágio de aprendizagem superficial e a qualitativa tem início quando o aluno começa a fazer as relações, em um estágio de aprendizagem efetiva. O estágio quantitativo de aprendizagem ocorre primeiro e, depois, a aprendizagem efetiva, na fase qualitativa. Enquanto a fase quantitativa engloba os níveis Pré-estrutural, Uniestrutural e Multiestrutural, a fase qualitativa contempla os níveis Relacional e Abstrato Estendido. Vale ressaltar que o nível Abstrato Estendido contém o nível Relacional, que contém o nível Multiestrutural, e assim sucessivamente (RIZZO e POLETTI, 2021).

Figura 2 – Uma hierarquia dos verbos da taxonomia SOLO



Fonte: Biggs (2003) *apud* Rizzo e Poletti (2021, p. 1094).

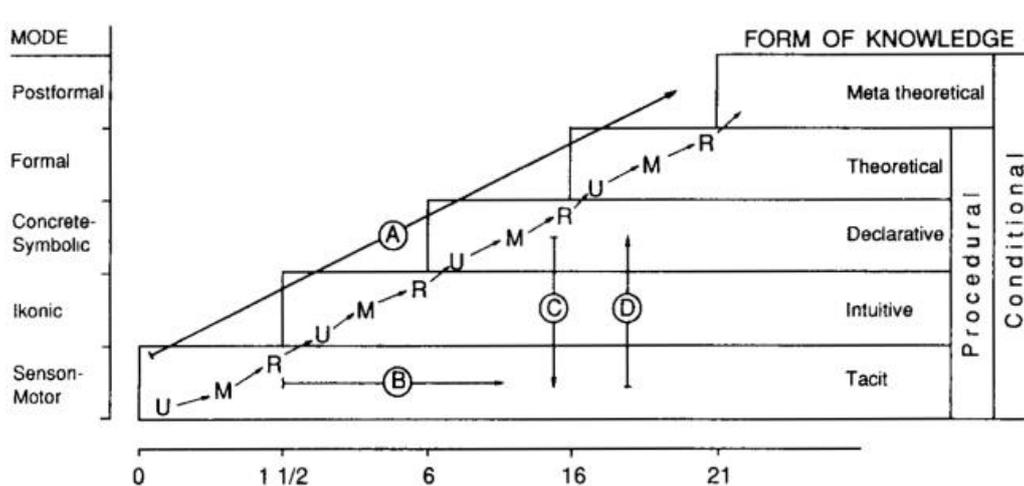
Visualizando a taxonomia SOLO em uma estrutura mais ampla da teoria cognitiva, Biggs (1992) apresenta um modelo, conforme exhibe a Figura 3, que fornece uma macroestrutura para visualizar a aprendizagem e o desenvolvimento, assim como denota que a experiência prévia obtida em sala de aula, é particularmente relevante para a aprendizagem no contexto escolar. Os modos fornecem a mídia representacional para os conteúdos de aprendizagem; esses conteúdos passam a ser progressivamente organizados dentro do ciclo de aprendizagem. No entanto, a aprendizagem pode não ser necessariamente permanecer intramodal. Muito do aprendido é modal cruzado, como é o caso das próprias respostas abstratas estendidas. Existem outras tarefas, educacionalmente significativas, que envolvem aprendizagem multimodal ou cruzada.

Na macroestrutura apresentada por Biggs (1992), os modos geralmente aparecem nas idades indicadas na abscissa e se acumulam conforme indicado na ordenada, permanecendo como meios potenciais de aprendizado ao longo da vida. O ciclo de aprendizagem progride de uniestrutural (U), passando por multiestrutural (M), para relacional (R) dentro de cada modo, a extensão de

relacional para abstrato estendido envolvendo uma transferência modal cruzada para uniestructural no próximo modo. Cada uma das quatro linhas, (a), (b), (c) e (d), representa um tipo qualitativamente diferente de desempenho (performance) cognitivo, assim sendo:

- Curso de desenvolvimento ótimo;
- Curso de aprendizagem dentro de uma modalidade (intramodal);
- Facilitação de cima para baixo da aprendizagem de ordem inferior (multimodal);
- Facilitação de baixo para cima da aprendizagem de ordem superior (multimodal).

Figura 3 – Modos, ciclos de aprendizagem e formas de conhecimento



Fonte: Biggs (1992, p. 39).

No que tange aos modos *versus* níveis exibidos na Figura 3, Biggs (1992) afirma que os tipos qualitativos de desempenho cognitivo (a) e (b) já foram estudados de maneira extensiva, diferentemente dos tipos (c) e (d), isto é, os multimodais. Esse autor ainda destaca que cinco formas de conhecimento emergem diretamente de cada um dos cinco modos (tácito, intuitivo, declarativo, teórico e metateórico), havendo um sexto modo, denominado procedimental, operando dentro dos modos, e um sétimo, chamado de condicional, o qual opera entre modos. Cada uma dessas formas de conhecimento, exceto o metateórico, pode ser alvo na aprendizagem escolar.

Biggs (1992) ressalta que sua teoria é um esquema que deve muito a outros modelos neo-piagetianos, mas que a difere em duas particularidades, ou seja, no

uso da experiência anterior conseguida durante a aprendizagem escolar e na suposição de que novos estágios não ofuscam os anteriores. Cabe frisar que uma característica do seu modelo é a de que um ciclo dentro do estágio de estruturação do conteúdo da aprendizagem se repete entre os estágios, melhor dizendo, entre os modos.

2.3 Alinhamento Construtivo

Biggs e Tang (2011) distingue três teorias comuns de ensino, dependendo do que é visto como o principal determinante da aprendizagem: (1) o que os alunos são; (2) o que os professores fazem e; (3) o que os alunos fazem. Essa categorização define “níveis” de pensamento sobre o ensino.

- a) No nível 1, o papel do professor é expor a informação, enquanto o dos alunos é absorvê-la; se os alunos não têm capacidade ou motivação para fazer isso corretamente, o problema é deles;
- b) No nível 2, o papel do professor é explicar conceitos e princípios, bem como apresentar informações. Para tal, os docentes precisam de várias habilidades, técnicas e competências. Aqui o foco está no que o professor faz, e não no que o estudante é, e nessa proporção, é mais reflexivo e sofisticado;
- c) No nível 3, o foco está no que os alunos fazem, dados que eles estão engajados nas atividades de aprendizagem com maior probabilidade de levar aos resultados pretendidos.

O alinhamento construtivo (Constructive Alignment – CA) tem como base dois princípios (BIGGS; TANG, 2011):

- a) “Construtivo”: princípio oriundo da teoria construtivista, o qual significa que os alunos usam sua própria atividade para construir seu conhecimento conforme interpretado por meio de seus próprios esquemas existentes;
- b) “Alinhamento”: princípio da teoria do currículo o qual define que as tarefas de avaliação devem ser alinhadas com o que se pretende aprender.

“O alinhamento construtivo estende de maneira prática a afirmação de

Shuel de que 'o que o aluno faz é realmente mais importante para determinar o que é aprendido do que o que o professor faz' (SHUEL; 1986 *apud* BIGGS; TANG; 2011, p. 97).

Ainda segundo Biggs e Tang (2011), se concentrar no que e como os estudantes devem aprender, em vez de quais tópicos o professor deve ensinar, requer que um resultado de aprendizagem pretendido, pois estabelece não apenas o que deve ser aprendido (tópico de conteúdo), mas como deve ser aprender e em que nível. Nesse sentido, a declaração do resultado define um verbo que informa ao aluno como ele deve se envolver para atingir o resultado de aprendizagem do tópico em questão, por exemplo, 'refletir sobre X' ou 'aplicar a teoria Y'. Vale ressaltar que os verbos devem ser considerados tanto nas atividades de ensino e aprendizagem, como também nas tarefas de avaliação.

É importante informar que a aplicação do alinhamento construtivo requer o entendimento da taxonomia cognitiva SOLO, para que o indivíduo se engaje na construção crescente de seu conhecimento, conforme o resultado de aprendizagem pretendido (RIZZO; POLETTI, 2019). Em uma visão simplificada, o alinhamento construtivo pode ser representado de acordo com a Figura 4, onde o professor elabora o resultado pretendido da aprendizagem (Intended Learning Outcome - ILO), planeja as atividades de ensino e aprendizagem (Teaching Learning Activities - TLAs) e define as tarefas de avaliação (Assessment Task - ATs). Cabe frisar que o planejamento das atividades de ensino e aprendizagem e das tarefas de avaliação são em função do resultado pretendido de aprendizagem.

Figura 4 – Etapas do alinhamento construtivo



Fonte: Rizzo e Poletti (2019, p. 187).

Biggs e Tang (2011) ressaltam que a aplicação do alinhamento construtivo exige uma reflexão transformadora. Na maioria das vezes, tal reflexão não é interessante de ser feita de maneira isolada. Nessa ótica, a participação de colegas de trabalho da mesma área de atuação é bastante significativa, dado que o colega conhece o contexto e pode contribuir com sugestões para haver essa

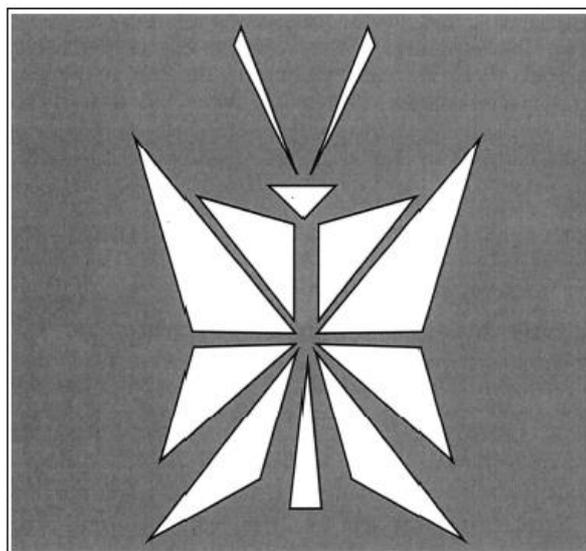
reflexão. Esses autores consideram que a diversidade de ideias é excelente, pois mostra que não existe uma única forma de implementar o alinhamento construtivo.

2.4 Abstração, Modelagem e Representação

Os humanos usam abstrações, uma ferramenta mental natural, para lidar com a complexidade. Ao filtrar detalhes insignificantes, a abstração acentua os elementos de importância central em qualquer nível de detalhe. Uma vez que a mente só pode lidar com uma quantidade finita de complexidade, é necessário usar a abstração para compreender sistemas complexos. A abstração evita a sobrecarga de complexidade por meio da ocultação de detalhes irrelevantes (HALLADAY; WIEBEL, 1993).

Dando prosseguimento, a Figura 5 exibe uma imagem abstrata, extraída de Halladay e Wiebel (1993). A fim de fazer um exercício rápido de abstração. O que você vê? O que você registrou em sua mente?

Figura 5 – Uma imagem abstrata



Fonte: Halladay e Wiebel (1993, p. 33).

Halladay e Wiebel (1993) explicam que a maioria dos observadores conclui imediatamente que a imagem da Figura 5 representa uma borboleta. Olhe para a imagem mais uma vez. Agora, parece haver algo além de uma borboleta. Ao analisar novamente essa imagem, você vai observar que a provável borboleta é composta de vários triângulos. Se você olhar a imagem uma terceira vez, poderá até vê-la como um conjunto de segmentos de linha organizados. Embora todas as

três explicações descrevam corretamente a referida figura, a imagem da borboleta salta à primeira vista, a qual se torna inconfundível porque a mente aplica um conceito chamado abstração aos segmentos de linha e triângulos.

Para os referidos autores, as diferentes quantidades de detalhes são níveis de abstração. Por exemplo, a imagem da borboleta é um nível de abstração. Os triângulos são um segundo nível. Os segmentos de linha são um terceiro. Conceitos mais detalhados, como os conceitos de linha, são níveis mais baixos de abstração. Por outro lado, conceitos mais gerais, como a imagem da borboleta, são níveis mais altos de abstração. A mente se move conscientemente de um nível de abstração para outro, decidindo quais detalhes considerar.

Os níveis de abstração naturalmente têm quantidades consistentes de detalhes. É difícil imaginar a Figura 5 como uma imagem de uma parte de uma borboleta misturada com alguns triângulos e alguns segmentos de linha. A mente vê a imagem como uma borboleta completa ou como um conjunto de linhas. Vale frisar que misturar níveis de abstração cria confusão mental (HALLADAY; WIEBEL, 1993).

Para esses autores, embora a abstração seja um processo mental que ocorre naturalmente, pode-se fortalecer os resultados exercendo disciplina consciente sobre o processo. Ao se deparar com um sistema complexo, é importante procurar vários níveis de abstração e analisar cada nível separadamente, tendo o cuidado de manter a consistência da abstração, como também entender o que deve ser evitado ao usar abstração. Diante disso, os quatro erros mais comuns são:

- a) Particionar elementos illogicamente para um nível de abstração;
- b) Reduzir ou pular níveis de abstração;
- c) Não identificar todos os elementos para um nível de abstração;
- d) Identificar elementos inexistentes para um nível de abstração.

É de se convir que uma simples imagem abstrata rendeu uma explicação considerável a respeito do processo de abstração, porém essa elucidação vai ser importante para o entendimento de determinados pontos tratados adiante.

Indo além da imagem abstrata da borboleta e adentrando na área da computação, é significativo saber por que esse processo mental é de extrema importância para o desenvolvimento de software. Na sequência, algumas

explicações são apresentadas.

Modelar significa construir uma abstração de um sistema que foca em aspectos interessantes e ignora detalhes irrelevantes de acordo com o contexto. A modelagem é uma maneira de lidar com a complexidade ignorando detalhes irrelevantes. O termo modelo se refere a qualquer abstração do sistema. Um modelo é uma representação abstrata de um sistema que nos permite responder a perguntas a respeito. Os modelos são úteis ao lidar com sistemas muito grandes, muito pequenos, muito complicados ou muito caros para serem experimentados em primeira mão. Os modelos também nos permitem visualizar e entender sistemas que não existem mais ou que apenas alegam existir (BRUEGGE; DUTOIT, 2009).

O paradigma da orientação a objetos faz uso intensivo de abstrações para se construir modelos consistentes e lógicos. A modelagem de software orientada a objetos pode ser representada visualmente por meio do suporte de diagramas UML cuja linguagem de modelagem é bastante empregada para representar modelos de software.

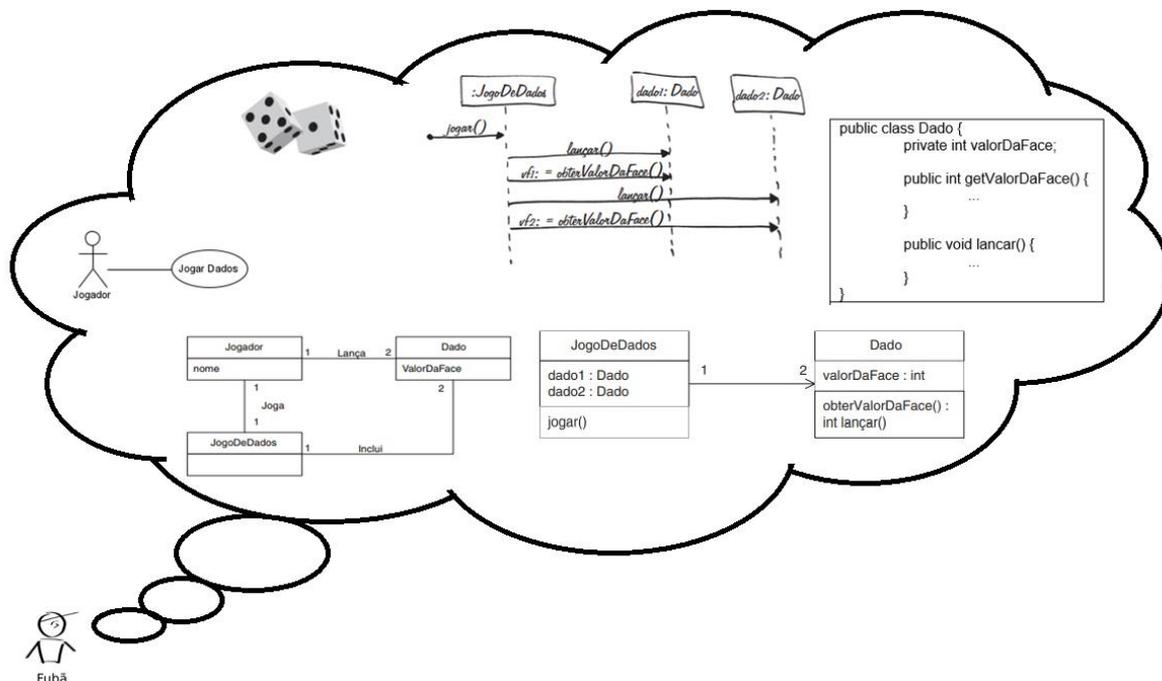
Para Engels *et al.* (2005), o ensino de UML implica no ensino de engenharia de software, como também acarreta o ensino da abstração, em outras palavras, ensinar UML permite uma ilustração de muitos conceitos cruciais de engenharia de software, que vão muito além de notações gráficas e textuais, sendo a abstração o mais importante dentre esses conceitos.

Sabe-se que o desenvolvimento de software inclui abstração e refinamentos sucessivos, objetivando cada vez mais um melhor entendimento do problema (análise), da solução (*design*) e da materialização da solução (implementação). Posto isso, os estudantes precisam aprender a construir modelos lógicos e representá-los de formas distintas. Apenas para exemplificar uma possível situação, observe a Figura 6, a qual busca representar a “mente” do personagem Fubã⁶ durante as atividades de análise, *design* e implementação de software. O exemplo em foco, isto é, um “Jogo de Dados”, apesar de simples, já permite visualizar a possível movimentação natural entre os níveis de abstração. Além disto, é possível imaginar uma dúvida assombrosa no pensamento do Fubã,

⁶ Um dos personagens da técnica de narrativas OC2-RD2 que representa a faceta da curiosidade investigativa.

a qual pode ser resumida na seguinte indagação “onde termina a análise e começa o *design*?”.

Figura 6 – Propósitos, formas e níveis de abstração na mente do Fubã

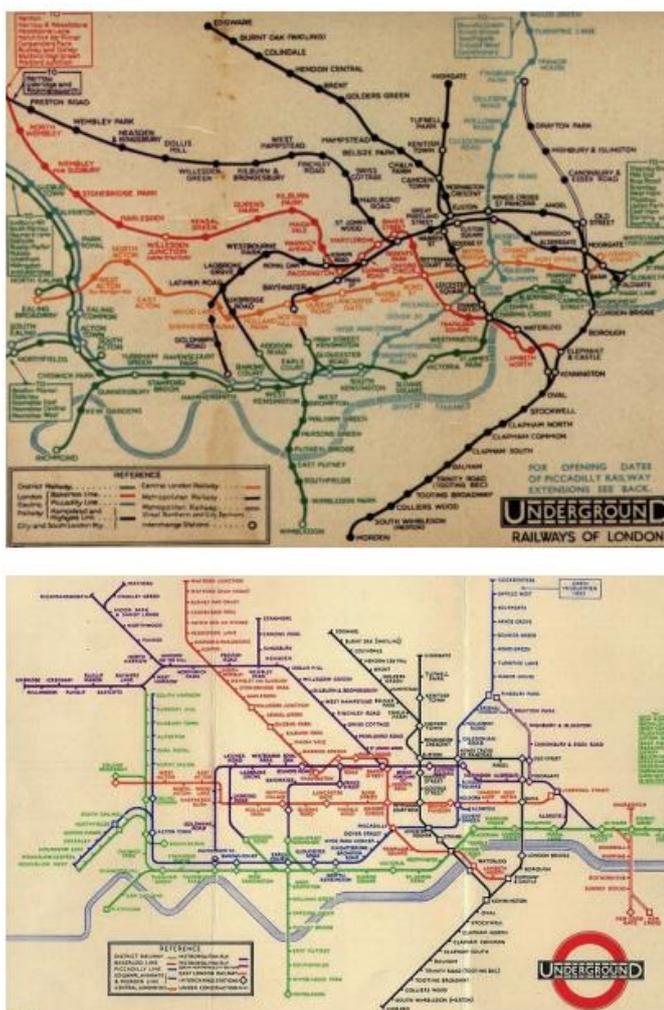


Fonte: Elaboração do autor com adaptações de Larman (2007, p. 36-38).

Essa movimentação, tratada aqui como manobra cognitiva entre os níveis de abstração e, conseqüentemente, a representação disso tudo em um conjunto de diagramas relacionados é, na maioria dos casos, uma atividade difícil de ser compreendida pela maior parte dos aprendizes.

Dando continuidade, Kramer (2007) apresenta um exemplo da utilidade da abstração e sua representação, o qual teve a contribuição de Harry Beck para o famoso mapa do metrô de Londres (Figura 7). Em 1928, o mapa era essencialmente uma sobreposição do sistema subterrâneo em um mapa geográfico convencional de Londres, mostrava as curvas das linhas de trem e do rio Tamisa, e as distâncias relativas entre as estações. Em 1931, Beck produziu a primeira representação esquemática abstrata, simplificando as curvas para linhas apenas horizontais, verticais e diagonais, na qual as distâncias entre as estações não eram mais proporcionais às distâncias geográficas.

Figura 7 – The London Underground Map (a) o mapa de 1928 e (b) o mapa de 1933



Fonte: Harry Beck (© Transport for London) *apud* Kramer (2007 p. 40).

A representação (abstração simplificada) da Figura 7 é apropriada para se navegar no metrô de Londres. Essa abstração de trânsito não é apenas usada hoje, mas também foi adotada por agências de transporte em vários países. Na verdade, o nível de abstração teve de ser cuidadosamente selecionado de modo a incluir apenas os detalhes necessários, mas ao mesmo tempo, negligenciar o desnecessário; em outras palavras, o mapa muito abstrato forneceria informações suficientes para o propósito; o mapa muito detalhado se tornaria confuso e menos compreensível. Como qualquer abstração, os programas de computador também podem significar a capacidade de lidar com abstrações de maneira precisa (KRAMER, 2007).

Kramer (2007) ainda considera que habilidades de abstração são essenciais na construção de modelos de análise e *design* (ou projeto), e

implementações adequados para o propósito específico em questão. O pensamento abstrato é essencial para manipular e raciocinar sobre abstrações, sejam elas modelos formais de análise ou programas em uma dada linguagem de programação.

Por sua vez, Blaha e Rumbaugh (2005) explicam que a abstração é o exame seletivo de certos aspectos de um problema. O objetivo da abstração é separar os aspectos importantes para alguma finalidade e omitir os que não o são. A abstração deve sempre ter um propósito, pois isso determina o que é e o que não é importante. Os autores ainda salientam que muitas abstrações da mesma coisa são possíveis, a depender do propósito para qual são feitas.

A abstração é apresentada em modelos e estes são usados para raciocinar sobre nossas abstrações (BOOCH, 2009). Sobre abstrações e representações, Parnas (1995; 2012) ressalta que:

- a) Um modelo de um produto é uma representação simplificada desse produto; um modelo pode ser físico (geralmente reduzido em tamanho e detalhes) ou abstrato;
- b) Um modelo terá algumas das propriedades importantes do original, mas não todas;
- c) Nem todas as propriedades do modelo são propriedades do sistema real.

Nesse sentido, os modelos podem ser muito úteis para os desenvolvedores porque permitem ao desenvolvedor analisar e entender as propriedades que são compartilhadas pelo sistema e pelo modelo. No entanto, um modelo deve ser usado com muito cuidado; o usuário deve estar sempre ciente de que as informações obtidas pela análise do modelo podem não se aplicar ao produto real (PARNAS, 2012).

Booch (2009) enfatiza que construir abstrações úteis é difícil, uma vez que elas devem ter graus de liberdade e, ao mesmo tempo, serem inequívocas. Além disso, esse autor realça que uma visão nunca é suficiente, ou seja, se tentar reunir todo o significado de uma coisa em uma única visão, isso vai suprimir o próprio código, e essa complexidade vai obscurecer os tópicos mais importantes que moldam o sistema.

A modelagem é parte do processo de abstração usado para representar

alguns aspectos de um sistema. Um conjunto de modelos internamente consistentes em diferentes níveis de abstração facilita a comunicação entre os usuários de software/partes interessadas e engenheiros de software (SWEBOK, 2014).

Segundo Blaha e Rumbaugh (2005), todas as abstrações são incompletas e imprecisas e que a realidade é uma teia sem emendas, e qualquer descrição dela é uma abreviação. Apesar das palavras e da linguagem humana serem abstrações, isto é, descrições incompletas do mundo real, isso não desfigura sua utilidade. O propósito de uma abstração é limitar o universo de modo que seja possível compreendê-lo. Durante a atividade de modelagem, não é necessário buscar a verdade absoluta, mas a adequação para alguma finalidade. Não existe um único modelo entendido como correto de uma situação, apenas modelos adequados e inadequados.

Blaha e Rumbaugh (2005) ainda consideram útil modelar um sistema de software partir de três perspectivas diferentes, todavia relacionadas, cada qual capturando pontos importantes, porém, todos necessários para uma representação completa, assim sendo:

- a) O modelo de classes representando os aspectos estáticos, estruturais, de “dados” do sistema, ou seja, descreve a estrutura dos objetos em um sistema;
- b) O modelo de estados representando os aspectos temporais, comportamentais, de “controle” de um sistema e, isto é, descreve os aspectos dos objetos que cuidam do tempo e da sequência de operações;
- c) O modelo de interações representando a colaboração de objetos individuais, os aspectos de “interações” de um sistema, quer dizer, descreve interações entre objetos, como os objetos individuais colaboram entre si para realizar o comportamento do sistema como um todo.

Dado que as visões dos três modelos são relevantes para o desenvolvimento de software, é importante representar esses modelos visualmente, por meio de diagramas. Nessa direção, uma linguagem de representação ganha seu espaço, tendo como exemplo, a UML.

O objetivo da UML é fornecer aos arquitetos de sistema, engenheiros de software e desenvolvedores de software ferramentas para análise, *design* e implementação de sistemas baseados em software, bem como para modelagem de negócios e processos semelhantes. Essa linguagem de modelagem originou-se de três métodos orientados a objetos (Booch, OMT e OOSE) e incorporaram várias práticas recomendadas de *design* de linguagem de modelagem, programação orientada a objetos e linguagens de descrição arquitetural (OMG, 2017).

Ensinar UML é ensinar engenharia de software, é ensinar abstração, em outras palavras, ensinar UML permite uma ilustração de muitos conceitos cruciais de engenharia de software, muito além de sua notação concreta. O mais importante entre esses conceitos é o da abstração (ENGELS *et al.*, 2005). Esses autores consideram que um problema central do ensino de modelagem (e, portanto, abstração) é que o processo de abstração não pode ser formalizado, já que depende da criatividade e da intuição. Conseqüentemente, nenhum modelo único pode ser designado na prática como a abstração ótima para um determinado problema, entretanto, há um espaço de boas soluções. Portanto, a construção de abstrações não pode ser aprendida seguindo uma receita determinística. Em vez disso, tal processo mental requer certa habilidade e intuição para detectar informações relevantes, estabelecer o nível correto de abstração e escolher representações adequadas para as informações. Essas habilidades não podem ser ensinadas com eficácia meramente em aulas expositivas (ou palestras). Tais habilidades devem ser baseadas na experiência, desse modo, os exercícios são absolutamente cruciais para o ensino de modelagem abstrata e representação desta em artefatos.

Recorrendo aos diagramas UML, o modelo de classes pode ser representado pelos diagramas de classe, o modelo de estados pelos diagramas de transição de estados, o modelo de interações pelos diagramas de sequência e diagramas de atividades. Note que um modelo de uma visão do sistema pode ser representado por um conjunto de diagramas, ou seja, não há uma relação unívoca entre modelo e diagrama.

Os diagramas são refinados sucessivamente no decorrer do desenvolvimento de software. Na análise, o modelo de domínio é construído sem considerar a eventual solução. Depois, o modelo de *design* acrescenta detalhes

de solução ao modelo. Então, a implementação codifica o modelo anterior, materializando a solução como um software.

Os diagramas UML são representações visuais que possibilitam retratar três perspectivas (LARMAN, 2002):

- a) Perspectiva conceitual: os diagramas são interpretados como descrevendo coisas em uma situação do mundo real ou domínio de interesse;
- b) Perspectiva de especificação (software): os diagramas (usando a mesma notação que na perspectiva conceitual) descrevem abstrações ou componentes de software com especificações e interfaces, mas nenhum compromisso com uma implementação particular (por exemplo, não especificamente uma classe em C# ou Java);
- c) Perspectiva de implementação (software): os diagramas descrevem as implementações de software em uma tecnologia específica (como em C# ou Java).

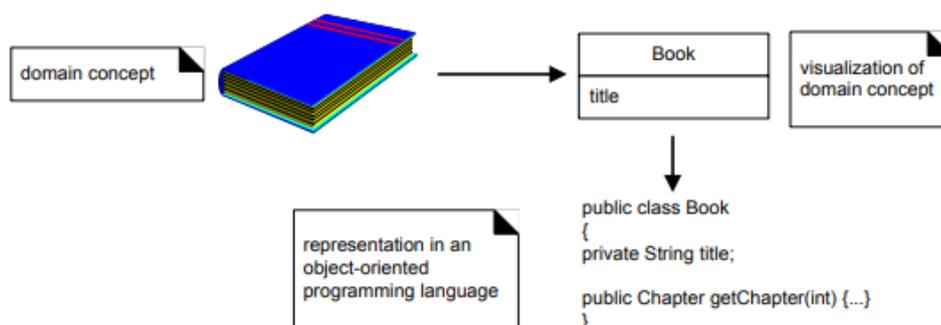
Visto que esta seção mencionou assiduamente os termos análise, *design*, implementação e, em muitos momentos, a palavra objeto, é relevante saber como cada um desses processos (estágios de desenvolvimento) lida com o conceito de objeto. Larman (2002) ressalta a significância do pensamento em objetos (modelos abstratos de software) e explana que:

- a) Durante a análise orientada a objetos, há uma ênfase em encontrar e descrever os objetos (ou conceitos), do domínio do problema;
- b) Durante o *design* orientado a objetos (ou simplesmente projeto de objetos), há uma ênfase na definição dos objetos de software e como eles colaboram para atender aos requisitos;
- c) Finalmente, durante a implementação orientada a objetos, os objetos de software são implementados.

E quanto a representação desse objeto? Apenas como exemplo, observe a Figura 8. As três representações se referem ao mesmo objeto, no caso um livro. Isso significa que, em um sistema de software de livraria, o livro é considerado um conceito do domínio de negócio, mas também uma classe conceitual na análise e uma classe em Java na implementação; ainda é possível abstrair que há uma

classe de especificação (de software) no *design*, ao refinar a classe conceitual em e transitar em direção à implementação desta.

Figura 8 – Objeto livro e algumas representações



Fonte: Larman (2002, p. 7).

A Figura 8 ilustra nitidamente as diferentes representações para o objeto livro, porém, é importante frisar que nenhuma dessas representa diretamente o objeto físico do domínio de negócio (livraria), ou seja, não há uma correspondência direta entre o mundo real e as representações de análise, *design* e implementação.

Por fim, ainda é cabível mencionar que a UML é uma linguagem semiformal de representação visual. Ao interpretar os conceitos de essência e acidente, essa linguagem pode ser vista como uma tarefa acidental, traduzindo aqui que a sintaxe do diagrama pode até estar correta, mas, sem semântica, ou seja, a representação não produz significado para quem observa o diagrama.

2.5 Técnica OC2-RD2

As estruturas narrativas OC2-RD2 foram adaptadas da teoria de narrativa denominada narratologia (Bal, 2009), teoria esta que estuda narrativas, textos narrativos, imagens, espetáculos, eventos, artefatos culturais que contam uma história, ajudando no entendimento, análise e avaliação de narrativas. O texto narrativo é considerado o elemento central da referida técnica e determina os três pilares para sua aplicação: fábula, história e texto, conforme mostra a Figura 9. (BUTTIGNON, 2020).

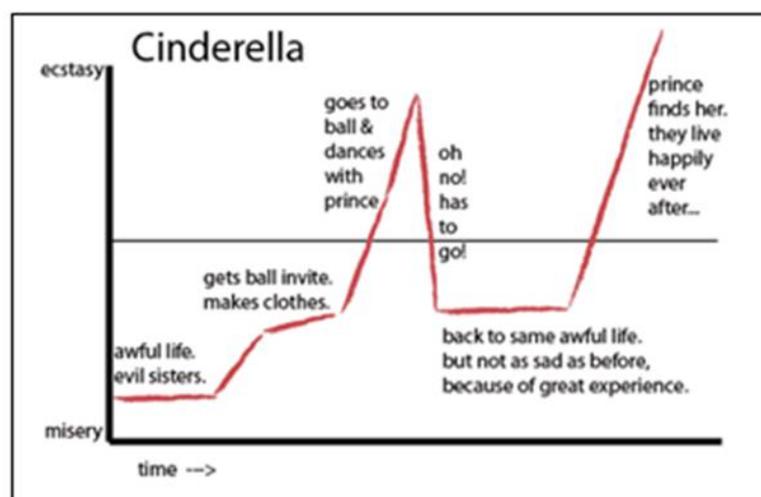
7Figura 9 – Texto narrativo de Bal



Fonte: Buttignon (2020, p. 88).

Essa técnica também se inspirou na palestra de Kurt Vonnegut em New York que relacionou os tipos de emoções das pessoas conforme a evolução de uma história, onde o tempo se move da esquerda para direita e a felicidade de baixo para cima, conforme mostra a Figura 10 que representa a oscilação das emoções para a história da Cinderela. Mesmo que esse drama já tenha sido escrito tantas vezes e contado de diversas formas distintas, as pessoas amam essa história. As pessoas pensam que suas vidas deveriam ser assim, ou seja, elas precisam de drama em suas vidas (SIVERS, 2009).

Figura 10 – Gráfico de Kurt Vonnegut para explicar o drama da Cinderela



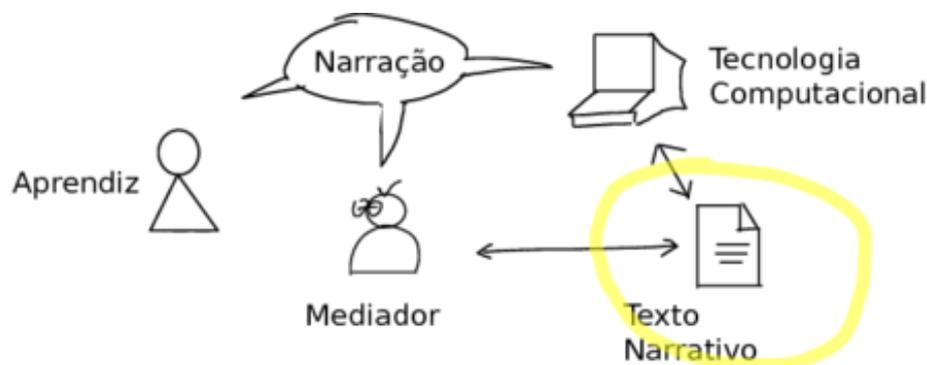
Fonte: Sivers, 2019.

No caso das narrativas OC2-RD2, ao invés das emoções das pessoas serem causadas por conta do drama gerado ao longo da história, as emoções surgem mediante o grau de aprendizagem dos estudantes durante a narração da

história na aula, podendo oscilar entre a presença e a ausência de conhecimento sobre o tema abordado. Outrossim, vale ressaltar que em ambas as situações, mesmo havendo oscilações de emoções, as pessoas buscam um final feliz para a história (VEGA, 2016; VEGA, 2018; BUTTIGNON, 2020).

A técnica OC2-RD2 se caracteriza pelo uso de narrativas como uma técnica de ensino nas aulas de cursos de computação, buscando construir um ambiente de aprendizagem onde a narração tem um papel importante nas interações entre três tipos de atores: aprendiz, mediador (mestre) e tecnologia computacional, conforme exibe a Figura 11.

Figura 11 – Narração no ambiente de aprendizagem



Fonte: Vega (2018, p. 7).

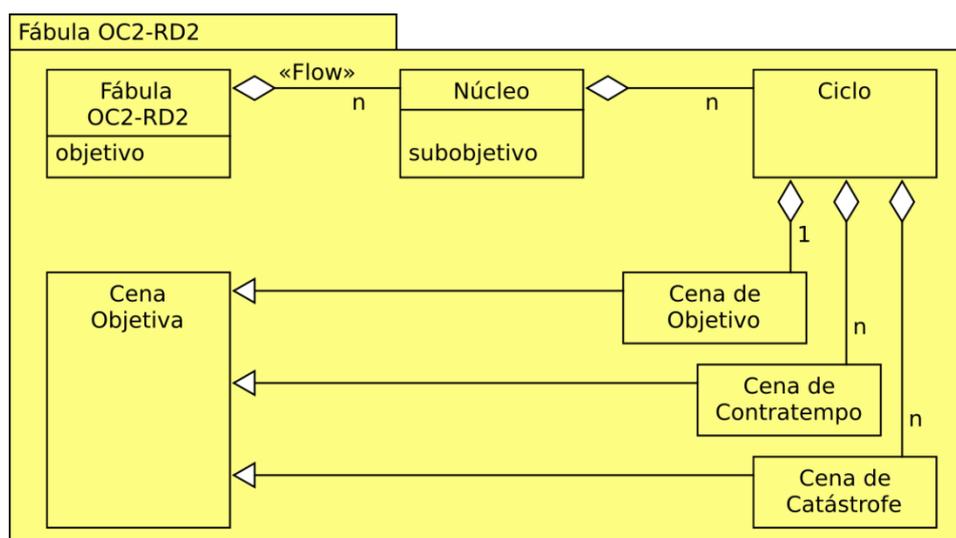
A técnica OC2-RD2 apresenta núcleos narrativos com base na fábula, primeiro pilar, propiciando condições para que o aprendiz interprete o problema a ser solucionado e reflita sobre a questão e sua solução. O texto narrativo OC2-RD2 se apoia no conceito de fábula abordado pela teoria de narrativa de Bal (2009; 2017), a qual define fábula como uma série de acontecimentos lógicos e temporais, embora apenas os primeiros acontecimentos sejam considerados pela técnica. Uma fábula OC2-RD2 contém as relações de causa e efeito, em outros termos, uma sequência de eventos e acontecimentos que estabelecem segmentos de fluxos lógicos que são posteriormente encadeados na elaboração de uma história (BUTTIGNON, 2020).

A fábula é a sequência de eventos de uma história, sendo conhecida também por intriga, ação, trama e enredo. As estruturas narrativas OC2-RD2 assumem que a fábula é um elemento central de interatividade numa narração. Na fábula OC2-RD2, os eventos são organizados por núcleos narrativos que enfatizam os conteúdos de conhecimentos que devem ser assimilados pelo

aprendiz (VEGA, 2018; BUTTIGNON, 2020).

A fábula tem um objetivo, sendo organizada por núcleos narrativos, cada qual com o seu subjetivo. Para atingir cada subobjetivo, são previstos ciclos com cenas objetivas, sendo estas de Objetivo, Contratempo e Catástrofe. A Figura 12 mostra a estrutura de uma fábula OC2-RD2.

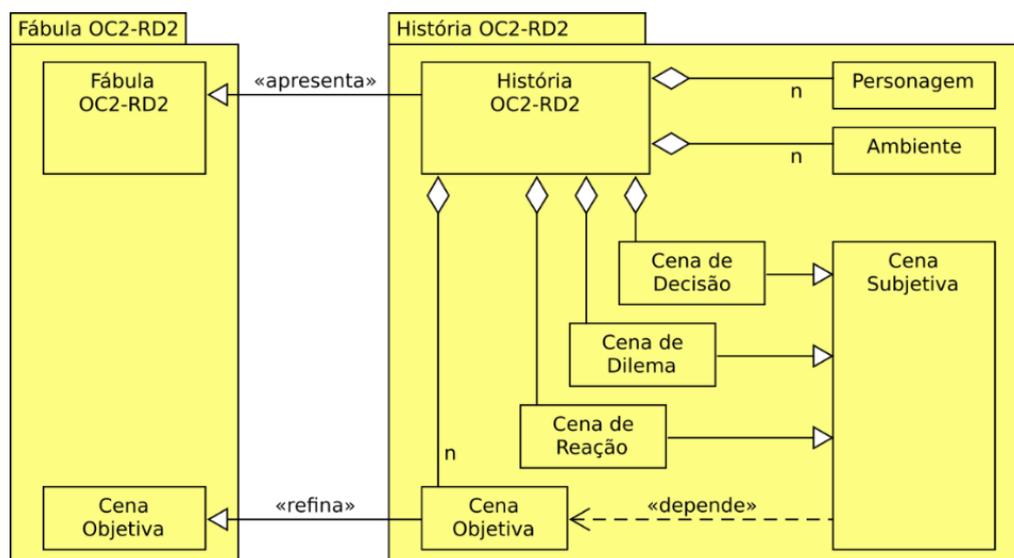
Figura 12 – Estrutura da fábula OC2-RD2



Fonte: Vendramel, Guirelli e Vega (2020, p. 110).

Com base nas sequências de eventos definidas pela fábula, a história, segundo pilar, deve ser trabalhada para narrar a trama aos aprendizes. As cenas objetivas da fábula são refinadas por cenas objetivas de história, contemplando ainda outros tipos de elementos narrativos: ambientes, personagens e cenas subjetivas. As cenas objetivas devem respeitar a ordem dos eventos especificada pela fábula. As cenas subjetivas (Reação, Dilema e Decisão) também são elaboradas a partir das cenas objetivas, porém devem considerar os aspectos emocionais do aprendiz que levam aos momentos de interação de aprendizagem (VEGA, 2018; BUTTIGNON, 2020). A Figura 13 exibe a estrutura de uma história OC2-RD2. Os ambientes são os espaços onde a história é contada, isto é, onde ela se desenrola, podendo ser uma sala de aula, uma sala de projeto, um laboratório ou outro espaço de aprendizagem propício (VEGA, 2018; VENDRAMEL; GUIRELLI; VEGA, 2020).

Figura 13 – Estrutura da história OC2-RD2



Fonte: Vendramel, Guirelli e Vega (2020, p. 111).

Nas cenas objetivas, a narração é feita em terceira pessoa, no caso pelo mestre, buscando sempre apresentar pontos a serem atingidos por personagens após a superação de obstáculos que necessitam de um conhecimento específico sobre o tema contemplado pela fábula. As cenas subjetivas, por sua vez, desviam o ponto de vista narrativo para a primeira pessoa, revelando a perspectiva do personagem em relação aos acontecimentos da história, conferindo interatividade às narrativas e incorporando um modelo para se avaliar a aquisição de conhecimento, importante para o mestre identificar oportunidades de intervenção no processo de aprendizagem (VEGA, 2018; BUTTIGNON, 2020).

O texto, terceiro e último pilar, foi adaptado na técnica para oferecer os recursos tecnológicos para a narração da fábula (ou história) OC2-RD2 (BUTTIGNON, 2020).

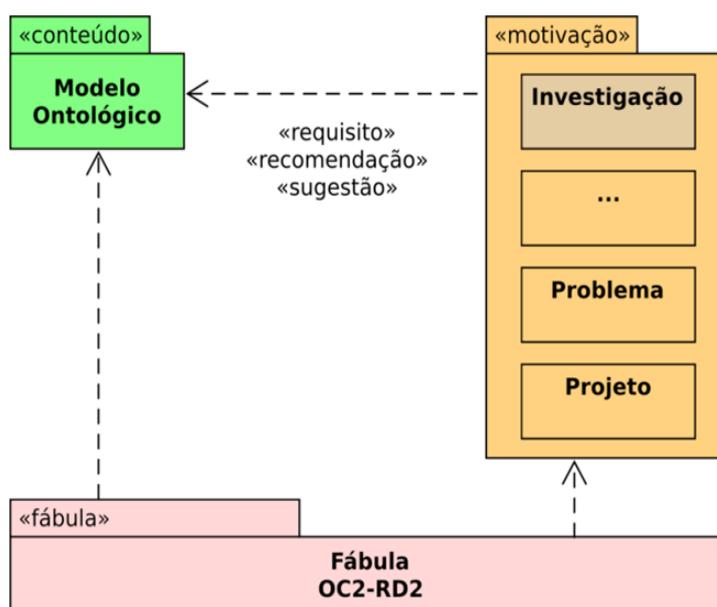
O contexto de uma narração deve possibilitar ao aprendiz, ao assisti-la, tomar consciência da situação enfrentada pelo personagem principal. Neste sentido, a elaboração de uma cena objetiva precisa levar em consideração um plano de conteúdos e um plano de motivações, a partir dos quais são projetadas as cenas da história e, em certas ocasiões, refinados os eventos da fábula.

2.5.1. Planos Organizacionais OC2-RD2

Os requisitos para a aplicação de uma história OC2-RD2 envolvem três planos organizacionais: 1) plano de conteúdos; 2) plano de motivações e; 3) plano

OC2-RD2, sendo este dividido em três etapas: a) elaboração da fábula; b) elaboração da história e; c) implementação da história. Aplica-se a técnica em dois modos: planejamento e condução do encontro. No modo de planejamento, constrói-se uma história cuja estrutura narrativa inclui cenas de objetivo, contratempo e catástrofe (VEGA, 2018). A Figura 14 exhibe os planos organizacionais OC2-RD2.

Figura 14 – Planos organizacionais OC2-RD2



Fonte: Vega (2018, p. 17).

O plano de conteúdos contém os itens de conhecimento a serem adquiridos pelo aprendiz no contexto do ambiente de aprendizagem. Isso requer a modelagem organizacional desses itens por meio de um grafo cujas arestas capturam as intenções de relacionamentos entre os diferentes itens: requisito, recomendação ou sugestão (VEGA, 2018):

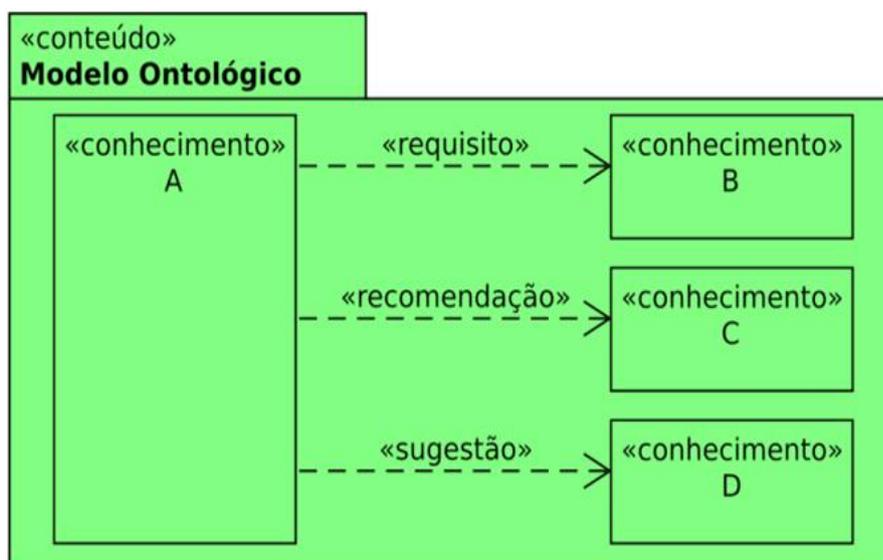
- a) Requisitos: os diferentes itens de conhecimento são relacionados por arestas direcionadas de requisitos. A direção da aresta estabelece uma relação de pré-requisito entre itens, dependendo do autor do plano de conteúdos;
- b) Recomendações: arestas de recomendação também podem ser utilizadas entre os nós do plano conteúdo, de acordo com propósitos didáticos do autor, entretanto, a direção da aresta aponta para a recomendação de um item de conhecimento que auxilie no processo de

aprendizagem sob o ponto de vista de quem prepara este plano;

- c) Sugestões: arestas de sugestão indicam eventuais complementações de estudo, não sendo requeridas ou recomendadas por razões ontológicas ou didáticas na visão do autor do plano. A direção da aresta aponta para a sugestão de um item de conhecimento, muitas vezes estabelecendo uma conexão com itens de outras áreas, até mesmo em uma ótica interdisciplinar.

A Figura 15 mostra os três tipos de dependência entre itens de conhecimento.

Figura 15 – Tipos de dependência entre itens de conhecimento



Fonte: Vega (2018, p. 18).

Para Vega (2018), embora dependa do julgamento (subjetivo) do autor do ambiente de aprendizagem, a relação de requisito entre itens de conhecimento deve preservar, quando existir, aquela da ontologia da área-alvo de conhecimento, em acordo com a literatura da área em questão.

O processo de construção do texto narrativo para ambientes de aprendizagem quais implica a existência de um mediador-narrador deve considerar elementos de alguma teoria narrativa. O plano de aula, visto como um roteiro narrativo, passa a exigir conhecimentos adicionais para a sua concepção (VEGA, 2018). As etapas para a construção do roteiro de um texto narrativo são:

- 1) Elaboração do Plano de Conteúdos: essa etapa define o plano

organizacional de conteúdos que deve conter os itens de conhecimento a serem adquiridos pelo aprendiz no contexto do ambiente de aprendizagem;

- 2) **Elaboração do Plano de Motivações:** essa etapa define o plano organizacional de motivações que deve contemplar os elementos que visam despertar o interesse do aprendiz, contextualizando a aplicação de conteúdos em situações-exemplo;
- 3) **Elaboração do Plano de Fábulas:** Esta etapa define o plano organizacional de fábulas que se apoia nos dois outros planos organizacionais: conteúdos e motivações. A fábula é a sequência de eventos de uma história, sendo conhecida também por intriga, ação, trama e enredo. A relação entre as cenas da fábula irá definir o fluxo narrativo principal das histórias, especificando a ordem dos eventos do fluxo e interconectando as cenas desse plano por relações de transição simples e relações de causa-efeito;
- 4) **Criação da História:** nessa etapa, elabora-se uma história para ser contada a partir da fábula criada. Basicamente, a história refina as cenas objetivas, introduz personagens e ambientes, bem como as cenas subjetivas que são os canais de interação planejados para manter o aprendiz engajado durante a narração. As cenas objetivas da história devem preservar os planos organizacionais de conteúdos e de motivações, mesmo na presença dos diálogos e das narrativas envolvendo os personagens. A escolha do personagem que irá assumir o papel principal depende do objetivo da história;
- 5) **Implementação da História:** Nesta última etapa, a história é implementada. Para tal, escolhem-se as mídias apropriadas para narrar a história em um ambiente de aprendizagem. Abrem-se, aqui, diversas oportunidades para o uso de tecnologias educacionais.

Apesar da menção das cinco etapas para se construir um texto narrativo, é importante salientar, desde já, que o interesse primário deste trabalho está nos itens de conhecimentos e suas dependências, os quais são providos a partir da elaboração do plano organizacional de conteúdos. Apesar de relevante para a técnica de narrativas OC2-RD2, o estudo em questão trata a elaboração dos

planos organizacionais de motivações e de fábulas de maneira secundária. Quanto à criação e implementação da história, essas etapas não são abordadas por esta pesquisa. Outrossim, o enfoque desta pesquisa é aplicar a técnica no modo de planejamento, não na condução do encontro.

3 METODOLOGIA DE PESQUISA

"Educação não transforma o mundo. Educação muda pessoas. Pessoas transformam o mundo."

(Paulo Freire)

Este capítulo tem a finalidade de caracterizar a metodologia de pesquisa empregada nesta tese, além de explicar a construção do procedimento de coleta de dados, e descrever a aplicação de tal instrumento e a caracterização da amostra da pesquisa.

3.1 Características Metodológicas

Em linhas gerais, a caracterização desta pesquisa segue as diretrizes de Gray (2009) e Creswell (2010). Este estudo é de abordagem qualitativa e propósito exploratório, a qual aplica a entrevista semiestruturada como procedimento de pesquisa para se fazer a coleta de dados. Este trabalho teve início por meio de revisão da literatura sobre os principais conceitos que, além de oferecer subsídios conceituais para a fundamentação teórica, revelaram lacunas de estudo e nortearam a presente pesquisa.

Em consonância ao que foi sucintamente descrito na Seção 1.5, e buscando aqui elucidar os detalhes cabíveis para responder à questão de pesquisa e atingir os objetivos estabelecidos presentes neste trabalho, este estudo se apoia no seguinte percurso metodológico:

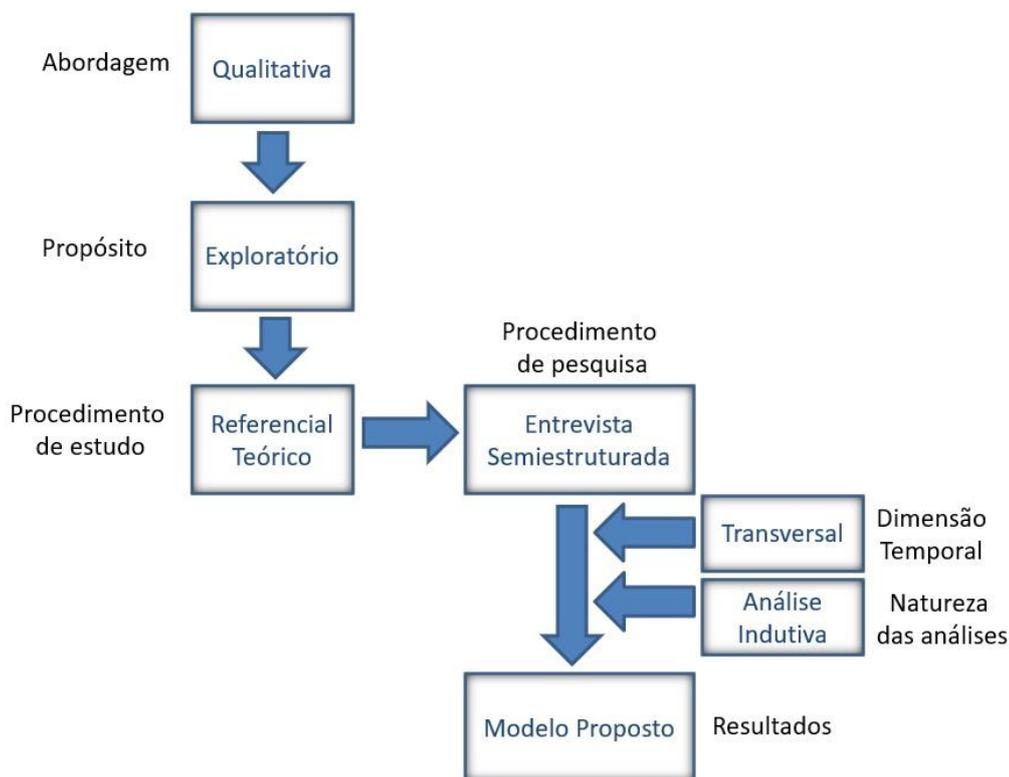
- a) Procedimento de estudo: a partir de referencial teórico, incluindo artigos, dissertações, teses, livros, normas, guias e notas de aula a respeito dos principais temas abordados nesta tese;
- b) Elaboração do procedimento de pesquisa: um roteiro de entrevista semiestruturada foi elaborado com base na literatura estudada e devidamente apreciado pelo Comitê de Ética em Pesquisa;
- c) Aplicação do procedimento de pesquisa: a condução da entrevista semiestruturada foi aplicada para coletar dados junto a uma amostra de professores de disciplinas relacionadas à engenharia de software de diferentes cursos e instituições de ensino. Do ponto de vista do tempo, a coleta caracterizou-se como transversal, em virtude dessas

entrevistas terem sido aplicadas em um determinado período, especificamente em novembro de 2022;

- d) Análise e Interpretação dos resultados: após a etapa de coleta, os dados provenientes do conhecimento e experiência dos professores foram analisados e interpretados qualitativamente via método amplo indutivo, melhor dizendo, das premissas particulares para conclusões mais amplas, produzindo, deste modo, agrupamentos por similaridade;
- e) Validade e Confiabilidade dos resultados: ao longo da aplicação do procedimento de pesquisa, como também após as etapas de coleta, análise e interpretação dos dados, certos procedimentos estratégicos foram aplicados na tentativa de assegurar a validade e confiabilidade dos resultados;
- f) Após análise, interpretação e aplicação de algumas estratégias para garantir a validade e confiabilidade qualitativa, os resultados foram utilizados para construir a proposição do modelo pedagógico abordado neste trabalho.

O percurso metodológico supracitado é ilustrado de forma sintetizada na Figura 16.

Figura 16 – Percurso metodológico da pesquisa



Fonte: Elaboração do autor.

Por se tratar de uma pesquisa de natureza qualitativa, o enfoque não está nos valores quantitativos exatos, mas sim nas descrições qualitativas. Com base em Creswell (2010), certas características desse tipo de pesquisa são reconhecidas neste estudo, as quais são: cabendo realçar que não há aqui uma ordem de relevância:

- a) O pesquisador como um instrumento fundamental: o pesquisador fez a coleta de dados por meio de entrevista semiestruturada. O próprio pesquisador elaborou tal procedimento e coletou os dados junto aos participantes;
- b) Análise indutiva de dados: o pesquisador criou seus próprios padrões, categorias e temas de baixo para cima, organizando os dados em unidades de informação cada vez mais abstratas. Esse processo indutivo organizou o trabalho de um lado para o outro entre os temas de interesse e os dados coletados até o pesquisador estabelecer uma relação abrangente de temas. Cabe destacar que a colaboração interativa com os participantes proporcionou a

oportunidade de dar forma aos temas e abstrações que emergiram das análises;

- c) Significados dos participantes: durante todo o processo deste estudo qualitativo, o pesquisador se manteve concentrado na aprendizagem do significado que os participantes deram para a questão de pesquisa;
- d) Projeto emergente: o projeto deste estudo foi considerado emergente e, por meio da natureza das análises qualitativas, surgiu a oportunidade de aprender mais (construir o conhecimento) sobre a questão a ser investigada junto aos participantes e atuar na pesquisa de modo que fosse possível adquirir informações relevantes;
- e) Lente teórica: o pesquisador frequentemente buscou usar lentes distintas para enxergar o estudo, possibilitando o uso da teoria de diversas perspectivas;
- f) Interpretativo: por um lado, a pesquisa qualitativa é um modo de investigação que permite ao pesquisador interpretar aquilo que se enxerga, ouve e entende. Por outro, é fato que suas interpretações não podem ser separadas de suas origens, história, contextos e entendimentos anteriores. Apesar disso, depois que o estudo é liberado por meio de um relato de pesquisa, os leitores e participantes podem interpretar os resultados de maneira distinta, podendo inclusive, provocar outros entendimentos do estudo, ou seja, as interpretações realizadas pelos leitores, participantes e pesquisadores proporcionam múltiplas visões emergentes do problema em questão;
- g) Relato holístico: o pesquisador procurou desenvolver um quadro complexo da questão em estudo. Isso envolveu o relato de múltiplas perspectivas, a identificação de numerosos fatores envolvidos no contexto e o esboço de um quadro mais amplo que emergiu das análises qualitativas, resultando um modelo pedagógico (ontológico), representado visualmente e constituído por muitas facetas relacionadas ao planejamento de ensino de engenharia de software, tanto do ponto de vista de conteúdos, como também dos níveis de compreensão.

Segundo Gray (2009), as pesquisas com propósito exploratório buscam identificar o que está ocorrendo e fazer perguntas quando não se sabe o suficiente a respeito de um fenômeno. Nesse sentido, este estudo se concentrou em buscas na literatura, como também conduzindo entrevistas com uma amostra de especialistas sobre a temática em foco.

Como procedimento de pesquisa, se elaborou um roteiro de entrevista semiestruturada. Dentre as peculiaridades das entrevistas descritas por Gray (2009), algumas dessas estão presentes neste trabalho, uma vez que tal procedimento de pesquisa possibilita:

- a) Prover informação com potencial para explorar uma questão de pesquisa em maior profundidade;
- b) Explorar histórias e perspectivas dos entrevistados;
- c) Obter riqueza nas respostas;
- d) Permitir ao entrevistador saber a quem entrevistar e;
- e) Trabalhar com uma unidade de amostra menor.

Outras particularidades da entrevista identificadas neste estudo qualitativo, ainda mais por ser semiestruturada, conforme Creswell (2010), são:

- a) A entrevista foi conduzida remotamente, dado que os participantes não podiam ser observados diretamente;
- b) Por se tratar de uma entrevista semiestruturada, o pesquisador pôde alterar a ordem das perguntas, a depender do entrevistado, além de ter havido a alternativa de não utilizar a relação completa de perguntas para todos os participantes;
- c) A entrevista semiestruturada proporcionou um aprofundamento das visões dos participantes.

No que tange aos requisitos éticos, este estudo obteve o consentimento do participante por meio de um Termo de Consentimento Livre e Esclarecido (TCLE), disponível no Apêndice B. Ademais, vale complementar que o projeto de pesquisa foi submetido via Plataforma Brasil, onde foi gerado o Certificado de Apresentação de Apreciação Ética (CAAE) número "62792222.2.0000.5482". O parecer consubstanciado referente a este projeto foi aprovado pelo Comitê de Ética em

Pesquisa (CEP) da Pontifícia Universidade Católica de São Paulo (PUC-SP).

As próximas seções detalham a construção do procedimento de pesquisa utilizado para a coleta de dados, isto é, o roteiro de entrevista semiestruturada, assim como a aplicação de tal instrumento.

3.2 Construção do Procedimento de Pesquisa

Tendo como base a literatura estudada, apresentada nos Capítulos 1 e 2, foi plausível identificar temáticas pertinentes para esta pesquisa, as quais estão divididas em quatro conjuntos de assuntos, da seguinte forma:

- a) Desenvolvimento Cognitivo;
- b) Alinhamento Construtivo;
- c) Padrões de Abstração e;
- d) Itens de Conhecimento e Erros de Abstração.

O critério de seleção dos tópicos de interesse dentro de cada temática foi definido segundo o julgamento do próprio pesquisador em relação ao interesse da pesquisa, evidentemente apoiado pelo referencial teórico estudado. O Quadro 4 sintetiza as referências para a construção do instrumento de pesquisa; algumas foram utilizadas de forma mais explícita, outras proveram suporte subjacente ao estudo.

Quadro 4 – Síntese do referencial teórico utilizado na construção do procedimento de pesquisa

Temática	Tópicos de interesse relacionado à temática	Referências utilizadas
Desenvolvimento Cognitivo	Processos e artefatos de software	Biggs e Collis (1982) Biggs (1992) Becker (2012) Falkner, Vivian e Falkner (2013) Gluga <i>et al.</i> (2012) Lister (2011) Piaget (1978)
	Tipos de raciocínio	
	Conseguir fazer/construir e compreender ⁷ abstrações	
	Modos de funcionamento, ciclos de aprendizagem e formas de conhecimento	
	Tipos qualitativos de desempenho cognitivo	
	Teoria neopiagetiana	

⁷ Os verbos “conseguir fazer/construir e compreender” foram uma adaptação dos verbos “fazer e compreender” de Piaget (1978), uma tradução dos termos em francês “*réussir et comprendre*” de Piaget (1974). Becker (2012) considera que a tradução “conseguir e compreender” seria mais adequada.

Alinhamento Construtivo	Unidades e tópicos de modelagem e análise de software Taxonomia SOLO Resultados Pretendidos de Aprendizagem Técnicas de Ensino e Aprendizagem Tarefas de Avaliação	ACM e IEEE (2004) ACM e IEEE (2014) Biggs e Collis (1982) Biggs e Tang (2011) Mol e Matos (2019) Pressman (2011) Rizzo e Poletti (2019) Rizzo e Poletti (2021)
Padrões de Abstração	Abstração, modelagem e representação de objetos Propósitos, formas e níveis de abstração em modelagem e análise de software	Halladay e Wiebel (1993) Hazzan e Kramer (2006) Hazzan e Kramer (2016) Larman (2007) Yourdon e Argila (1999)
Itens de Conhecimento e Erros de Abstração	Planos organizacionais OC2-RD2 Plano de conteúdos e tipos de dependência entre itens de conhecimento Erros comuns de abstração	Halladay e Wiebel (1993) Vega (2018) Vega (2019)

Fonte: Elaboração do autor.

Cabe reforçar que todos esses conjuntos de tópicos foram direcionados para o processo de ensino e aprendizagem de Modelagem e Análise de Software (MAS), uma das áreas de conhecimento de Engenharia de Software. Quanto ao interesse deste estudo pela área de conhecimento em questão, vale recordar que uma das razões é a relevância dos tópicos dessa área em qualquer componente curricular de engenharia de software. “Modelagem e análise podem ser consideradas conceitos centrais em qualquer disciplina de engenharia porque são essenciais para documentar e avaliar decisões e alternativas de projeto.” (ACM; IEEE, 2014, p. 31). Outro motivo pela escolha da área de conhecimento em foco é que se observa essa ela elenca tópicos que enfatizam a importância da abstração, que por sinal isso já havia sido apontado por Kramer (2007). Isso não significa que as demais áreas de conhecimento não tratam esse tema com a devida importância, aliás, segundo ACM e IEEE (2014), abstração, modelagem e representação são temas que devem ser ensinados de maneira recorrente em um curso de computação.

Os tópicos de interesse de cada uma das quatro temáticas e a área de conhecimento de MAS, vista como a quinta temática, foram contemplados na

elaboração do instrumento de pesquisa, especificamente um roteiro de entrevista semiestruturada com a finalidade de coletar dados a partir de uma amostra de participantes que representam professores(as) de disciplinas de engenharia de software e/ou componentes curriculares correlatos para construir, sob a perspectiva docente, um modelo pedagógico composto por planos temáticos e reutilizáveis de modelagem e análise de software, com o intuito de orientar o professor, do ponto de vista de desenvolvimento cognitivo e progressão do aprendizado, no planejamento de ensino de engenharia de software.

A partir destas considerações, o roteiro da entrevista semiestruturada elaborada para este estudo pode ser conferido em detalhes no Apêndice A cuja estrutura contém seis seções de perguntas, divididas desta forma:

- a) A seção A é formada por cinco perguntas que identificam o perfil do participante da pesquisa. Vale reforçar que os dados dos respondentes são mantidos em sigilo, conforme TCLE;
- b) A seção B é composta por três perguntas preliminares sobre desenvolvimento cognitivo no ensino e aprendizagem de modelagem e análise de software;
- c) A seção C é constituída por seis questões a respeito do alinhamento construtivo no ensino e aprendizagem de modelagem e análise de software;
- d) A seção D é formada por oito perguntas sobre padrões para contextualizar a abstração no ensino e aprendizagem de modelagem e análise de software;
- e) A seção E é composta por duas questões sobre itens de conhecimento e erros de abstração no ensino e aprendizagem de modelagem e análise de software;

Finalmente, a seção F é constituído por quatro perguntas gerais a respeito do modelo pedagógico proposto, com o intuito de reconhecer indícios a respeito da proposição apresentada pelo presente estudo em relação aos objetivos do trabalho.

Vale informar que as referências elencadas previamente no Quadro 4 também proporcionaram a elaboração de um artefato adicional constituído por

uma coleção de representações (figuras e quadros) cujo propósito foi o de apoiar o participante durante a entrevista. Esse documento, denominado Representações de Apoio para o Roteiro de Entrevista, se encontra disponível no Anexo A.

Ainda é válido comunicar que os participantes receberam orientações a respeito da entrevista antecipadamente por e-mail, inclusive para assistirem ao vídeo "Teaching Teaching & Understanding Understanding"⁸, com o intuito de possibilitar uma visão prévia sobre determinados assuntos abordados pelo instrumento de pesquisa e, por consequência, prover suporte para responder certas indagações.

3.3 Aplicação da Entrevista e Caracterização da Amostra

Primeiro, cabe comunicar que a escolha dos entrevistados neste trabalho foi realizada intencionalmente, uma vez que segundo Creswell (2010), o intuito da pesquisa qualitativa é a seleção intencional dos participantes ou dos locais que melhor ajudam o pesquisador a compreender o problema ou a questão de pesquisa. Além disso, o estudo qualitativo não obriga a pesquisa ter uma amostragem ou seleção aleatória de numerosos participantes e locais, comumente visto na pesquisa quantitativa.

A amostra desta pesquisa é formada por 10 professores(as) que atuam ou atuaram no componente curricular de engenharia de software e/ou disciplinas afins em cursos superiores de computação e informática, os quais foram julgados pelo pesquisador como especialistas na temática abordada. Se trata de uma amostra pequena, típica em estudos qualitativos, relativamente de fácil acesso, dado que o pesquisador teve relações acadêmicas anteriores com esses docentes, refletindo assim maior validade e confiabilidade qualitativa nos resultados. Pode-se, até mesmo, entender que esta amostra configura uma versão de um painel de especialistas⁹.

8 Teaching Teaching & Understanding Understanding, filme premiado disponível em DVD pela Universidade de Aarhus, Dinamarca, escrito e dirigido por Claus Brabrand. Em menos de 20 minutos, Claus leva o espectador pelos fundamentos do alinhamento construtivo com Doina e Rune, versões dinamarquesas de Susan e Robert. Disponível na Internet com legendas em português em <<<https://www.youtube.com/watch?v=rM7KMi14ZHI>>> cuja duração é de 18min44s.

9 Nomenclatura extraída do Método Delphi. Disponível em https://en.wikipedia.org/wiki/Delphi_method.

Após contato prévio com cada participante e antes da aplicação do instrumento de pesquisa, os(as) professores(as) receberam orientações a respeito da entrevista por mensagens via e-mail e/ou WhatsApp¹⁰. Além disto, o TCLE foi encaminhado para o participante manifestar, de forma livre e esclarecida, seu consentimento em participar da pesquisa em estudo, como também foi enviado o artefato “Representações de Apoio para o Roteiro de Entrevista” e o link do vídeo “Teaching Teaching & Understanding Understanding”.

Todas as entrevistas foram conduzidas de forma remota via serviço de comunicação por vídeo Google Meet¹¹. As entrevistas foram realizadas no período compreendido entre 05 de novembro e 22 de novembro de 2022. Apenas como adendo, vale saber que a entrevista mais curta teve a duração aproximada de 1h30min, a mais longa durou cerca de 2h40min e a duração média das 10 entrevistas foi de 2h15min. Dos 10 participantes, sete optaram por participar da entrevista em duas sessões, enquanto outros três responderam as questões em uma única sessão. Em termos geográficos, oito participantes estavam situados no Brasil e os outros dois se localizavam no exterior.

Com base nos dados coletados pela pesquisa, exclusivamente os procedentes da Seção A do Roteiro de Entrevista, a amostra desta pesquisa é caracterizada no Quadro 5, no qual se tem noção do perfil dos participantes, os quais são identificados de P1 a P10.

Quadro 5 - Caracterização da amostra

Perfil versus Participante	Titulação máxima (Leg. A)	Anos de experiência	Organização Acadêmica de atuação (Leg. B)	Programa/Curso de atuação (Leg. C)	Disciplinas ministradas (Leg. D)
P1	D* (1)	3-5 (1)	INST (1) UNIV (1)	SI (1) CC (1) ADS (1) CD (1)	ESW (1) QSW (1) LP (1) BD (1)
P2	M (1)	3-5 (2)	FACUL (1)	SI (2) ADS (2) EC (1) Outro: DSM (1)	ESW (2) LP (2) Outra: ED (1) Outra: ALP (1)
P3	M (2)	Mais de 15 (1)	FACUL (2) CUNI (1)	ADS (3)	ESW (3) POO (1) LP (3)
P4	D (1)	Mais de 15 (2)	INST (2)	ADS (4) GTI (1)	APOO (1) POO (2) LP (4)

10 Disponível em <https://www.whatsapp.com>.

11 Disponível em <https://apps.google.com/meet>.

P5	M (3)	Mais de 15 (3)	FACUL (3) UNIV (2)	SI (3) ADS (5) ES (1) Outro: DSM (2)	ESW (4) QSW (2) APOO (3) POO (3) LP (5)
P6	D* (2)	6-10 (1)	INST (3)	ADS (6) CD (2)	ESW (5) POO (4) LP (6) BD (2)
P7	M (4)	Mais de 15 (4)	UNIV (3)	CC (2) ES (2) EC (2)	ESW (6) QSW (3) ASW (1) APOO (4) BD (3) Outra: ED (2) Outra: ALP (2)
P8	D* (3)	Mais de 15 (5)	FACUL (4) INST (4)	ADS (7)	ESW (7) ASW (2) APOO (5)
P9	D* (4)	Mais de 15 (6)	CUNI (2) INST (5) UNIV (4)	SI (4) CC (3) ADS (8)	ESW (8) APOO (6) POO (5) LP (7) BD (4) Outra: ED (3) Outra: ALP (3)
P10	D* (5)	11-15 (1)	FACUL (5) CUNI (3) UNIV (5)	ADS (9) ES (3)	ESW (9) QSW (4)
<p>(Leg. A): (M) Mestrado; (D) Doutorado; (D*) Estágio de Pós-Doutorado. (Leg. B): (FACUL) Faculdade; (CUNI) Centro Universitário; (INST) Instituto; (UNIV) Universidade. (Leg. C): (SI) Sistemas de Informação; (CC) Ciência da Computação; (ADS) Análise e Desenvolvimento de Sistemas; ES (Engenharia de Software); (EC) Engenharia da Computação; CD (Ciência de Dados); GTI (Gestão da Tecnologia da Informação); (DSM) Desenvolvimento de Software Multiplataforma. (Leg. D): (ESW) Engenharia de Software; (QSW) Qualidade de Software; (ASW) Arquitetura de Software; Análise e Projeto Orientado a Objetos (APOO); (POO) Programação Orientada a Objetos; (LP) Laboratório de Programação; (BD) Banco de Dados; Outra: (ED) Estrutura de Dados; Outra: ALP (Algoritmos e Lógica de Programação).</p>					

Fonte: Dados da pesquisa.

A respeito dos dados coletados, todos os participantes possuem formação *Stricto-Sensu*, sendo a maior parte (6) com titulação máxima de doutorado, incluindo a maioria destes com estágio de Pós-Doutorado (5); a maior parte com mais de 15 anos de experiência em cursos superiores de computação (6); pelo menos metade com histórico diversificado de atuação em Organizações Acadêmicas, especialmente Faculdade (5), Instituto (5) ou Universidade (5); Quase todos atuam ou atuaram no curso de Tecnologia em Análise e Desenvolvimento de Sistemas (9), seguido do curso de Sistemas de Informação (4) e depois dos cursos de Ciência da Computação (3) e Engenharia de Software (3); a maioria tem um histórico variado de disciplinas ministradas, destacando o

componente curricular de Engenharia de Software (9), seguido de Laboratório de Programação (7) e, em terceiro, a disciplina de Análise e Projeto Orientado a Objetos (6).

Os resultados provenientes das outras seções do Roteiro de Entrevista são analisados, interpretados e validados no próximo capítulo.

4 RESULTADOS DA PESQUISA

"l'objectif principal de l'éducation est de créer des gens capables de faire de nouvelles choses et de ne pas simplement répéter ce que d'autres générations ont fait."

(Jean William Fritz Piaget)

Este capítulo tem a incumbência de analisar, interpretar e validar qualitativamente os resultados provenientes dos dados coletados via roteiro de entrevista semiestruturada.

4.1 Análise dos Resultados sobre Desenvolvimento Cognitivo

Nesta seção, são apresentados os resultados das perguntas da Seção B, referentes ao desenvolvimento cognitivo, baseado em uma visão neopiagetiana, no processo de ensino e aprendizagem em Modelagem e Análise de Software (MAS). Ao todo são três questões cujas respostas são reveladas no Quadro 6. Vale lembrar que os detalhes das perguntas se encontram no Roteiro de Entrevista, disponível no Apêndice A.

Quadro 6 – Desenvolvimento cognitivo em uma perspectiva neopiagetiana para MAS

Pergunta <i>versus</i> Participante	B1	B2	B3
P1	C (1)	C (1)	D (1)
P2	C (2)	D (1)	E (1)
P3	C (3)	D (2)	E (2)
P4	C (4)	C (2)	D (2)
P5	C (5)	C (3)	B (1)
P6	A (1)	C (4)	D (3)
P7	C (6)	D (3)	C (1)
P8	C (7)	D (4)	E (3)
P9	C (8)	D (5)	E (4)
P10	C (9)	D (6)	E (5)

Fonte: Dados da pesquisa.

No que concerne os resultados da questão “B1”, quase todos os participantes (9) concordam que, para refinar o processo de ensino e aprendizagem em MAS, os tópicos dessa área de conhecimento devem ser planejados para o estudante compreendê-los como Processos e Artefatos,

havendo assim convergência com o estudo de Falkner, Vivian e Falkner (2013).

Sobre os resultados da questão "B2", a maioria (6) concorda que, para buscar aprimoramento do processo de ensino e aprendizagem em MAS, o professor deve planejar suas aulas para o estudante demonstrar os três grupos de comportamentos, facilitados tanto de baixo para cima quanto de cima para baixo, havendo desta forma correspondência com os estudos de Biggs e Collis (1982) e Biggs (1992) e Lister (2011) seguido pelo grupo de respondentes (4) o qual concorda que o estudante precisa demonstrar raciocínio hipotético-dedutivo; pensar de maneira abstrata; conseguir fazer e compreender abstrações que levam a realizações concretas, havendo consonância com os trabalhos de Lister (2011), Gluga *et al.* (2012) e Falkner, Vivian e Falkner (2013).

Quanto aos resultados da questão "B3", metade dos respondentes concorda que, para melhorar o processo de ensino e aprendizagem em MAS, o professor deve planejar suas aulas para o estudante demonstrar as quatro habilidades de abstração, facilitadas tanto de baixo para cima quanto de cima para baixo, havendo desta maneira, correspondência com os estudos de Biggs e Collis (1982) e Biggs (1992) e Lister (2011), seguido do grupo de participantes (3) o qual concorda que o estudante precisa construir e compreender abstrações que ultrapassem o domínio do problema, se concentrando em várias propriedades abstratas em um determinado momento sobre situações ainda não vivenciadas diretamente, havendo assim correspondência com os trabalhos de Lister (2011), Gluga *et al.* (2012) e Falkner, Vivian e Falkner (2013).

4.2 Análise dos Resultados sobre Alinhamento Construtivo

Nesta seção, são apresentados os resultados das perguntas da Seção C, alusivas ao alinhamento construtivo no processo de ensino e aprendizagem em modelagem e análise de software. Ao todo são seis questões cujas respostas são difundidas adiante. Cabe recordar que os detalhes das perguntas se encontram no Roteiro de Entrevista, disponível no Apêndice A.

A respeito das unidades e tópicos de modelagem e análise de software que podem elevar a noção de abstração no ensino de engenharia de software, o Quadro 7 expõe as respostas da questão "C1". Dos 13 temas disponíveis para essa área de conhecimento, rememorando que estes são oriundos do currículo de Engenharia de Software (ACM; IEEE, 2014), cada um dos respondentes

identificou cinco tópicos. Na visão dos participantes da pesquisa, os tópicos FM1 (9), TM1 (9), TM2 (6), TM4 (5) e FA1 (8) são os que mais podem ajudar no aumento da capacidade de abstração dos estudantes. Isso não significa que os conteúdos menos escolhidos, ou mesmo os não selecionados, não contribuam com a noção dessa capacidade cognitiva.

Quadro 7 – Tópicos de Modelagem e Análise de Software

Tópico <i>versus</i> Participante	FM1	FM2	FM3	TM1	TM2	TM3	TM4	TM5	TM6	FA1	FA2	FA3	FA4
P1	X			X			X			X	X		
P2	X			X	X		X					X	
P3	X	X		X	X					X			
P4	X				X	X				X	X		
P5		X		X		X	X				X		
P6	X			X		X		X		X			
P7	X			X	X			X		X			
P8	X			X	X		X			X			
P9	X			X		X				X	X		
P10	X			X	X		X			X			
Total	9	2	0	9	6	4	5	2	0	8	4	1	0

(FM): Fundamentos de Modelagem; TM (Tipos de Modelos); (FA) Fundamentos de Análise.

Fonte: Dados da pesquisa.

Ainda sobre os resultados das unidades e tópicos de MAS, é importante salientar que, conforme ACM e IEEE (2004; 2014), abstração, assim como modelagem e representação, são temas que devem ser ensinados, ao menos parcialmente, de maneira recorrente em todo o currículo do curso, para assim, ajudar o estudante a desenvolver uma mentalidade de engenharia de software. Hazzan e Kramer (2006; 2016) reforçam que a abstração é um conceito central em cursos de computação e engenharia de software, um assunto que deve ser tratado em todo o curso.

Em relação ao tipo de conhecimento (declarativo ou funcional) e ao nível de compreensão ou desempenho a ser alcançado no aprendizado de cada tópico, o Quadro 8 exterioriza as respostas das questões “C2” e “C3”. Da lista de verbos de conhecimento declarativo e funcional, lembrando que esses verbos de aprendizagem são derivados dos níveis da taxonomia cognitiva SOLO (BIGGS; COLLIS, 1982; BIGGS; TANG, 2011), cada participante identificou um ou dois

verbos de aprendizagem¹² por tipo de conhecimento, visando o aprendizado de cada um dos cinco temas escolhidos na questão anterior. Vale frisar que dos 13 temas dessa área de conhecimento, os três sem nenhuma escolha foram excluídos, visto que não constam dados para serem analisados e interpretados qualitativamente.

Quadro 8 – Verbos de aprendizagem para cada tópico por participante

	FM1	FM2	TM1	TM2	TM3	TM4	TM5	FA1	FA2	FA3
P 1	Descrever (1) Ilustrar (1)		Explicar (1) Construir (1)			Extrapolar (1)		Refletir/ Melhorar (1)	Comparar/ Contrastar (1)	
P 2	Identificar (1)		Classificar (1) Ilustrar (1)	Classificar (1) Ilustrar (1)		Analisar (1) Aplicar (1)				Teorizar (1) Refletir/Melhorar (1)
P 3	Descrever (2) Ilustrar (2)	Explicar (1)	Explicar (2) Construir (2)	Explicar (1) Construir (1)				Refletir/ Melhorar (2)		
P 4	Analisar (1) Resolver (1)			Construir (2)	Analisar (1)			Refletir/ Melhorar (3)	Analisar (1)	
P 5		Argumentar (1)	Descrever (1) Ilustrar (2)		Analisar (2) Construir (1)	Identificar (1)			Refletir/ Melhorar (1)	
P 6	Descrever (3) Ilustrar (3)		Descrever (2) Ilustrar (3)		Analisar (3) Construir (2)		Comparar/ Contrastar (1)	Refletir/ Melhorar (4)		
P 7	Identificar (2)		Comparar/ Contrastar (1) Construir (3)	Comparar/ Contrastar (1) Construir (3)			Descrever (1) Ilustrar (1)	Refletir/ Melhorar (5)		
P 8	Identificar (3)		Analisar (1) Aplicar (1)	Analisar (1) Construir (4)		Comparar e Contrastar (1) Resolver (1)		Refletir/ Melhorar (6)		
P 9	Identificar (4)		Descrever (3) Ilustrar (4)		Explicar (1) Aplicar (1)			Refletir/ Melhorar (7)	Refletir/ Melhorar (2)	
P 10	Descrever (4) Ilustrar (4)		Analisar (2) Construir (4)	Analisar (2) Construir (5)		Comparar/ Contrastar (2) Prever (1)		Hipotetizar (1) Refletir/ Melhorar (8)		

Fonte: Dados da pesquisa.

Ainda sobre os resultados relacionados ao tipo de conhecimento e ao nível de compreensão ou desempenho a ser alcançado no aprendizado de cada tópico, o Quadro 9 expõe as respostas das questões “C2” e “C3” novamente, mas desta

¹² Há verbos de aprendizagem compostos elencados na listagem original, assim sendo, esta pesquisa contabilizou como sendo um, por exemplo, Comparar/Contrastar e Refletir/Melhorar.

vez fazendo o cruzamento dos dados (verbo de aprendizagem *versus* tópico) escolhidos por cada participante.

Quadro 9 – Verbo de aprendizagem *versus* tópico por participante

			FM1	FM2	TM1	TM2	TM3	TM4	TM5	FA1	FA2	FA3
Nível SOLO(A)	Tipo(B)	Verbo										
U	D	Identificar	P2 P7 P8 P9					P5				
	F											
M	D	Classificar			P2	P2						
		Descrever	P1 P3 P6 P10		P5 P6 P9				P7			
	F	Ilustrar	P1 P3 P6 P10		P2 P5 P6 P9	P2			P7			
R	D	Analisar	P4		P8 P10	P8 P10	P4 P5 P6	P2			P4	
		Argumentar		P5								
		Explicar*		P3	P1 P3	P3	P9					
		Comparar/ Contrastar			P7	P7		P8 P10	P6		P1	
	F	Aplicar			P8		P9	P2				
		Construir			P1 P3 P7 P10	P3 P4 P7 P8 P10	P5 P6	P1				
		Resolver						P8				
	Prever						P10					
AE	D	Teorizar										P2
		Hipotetizar								P10		
	F	Refletir/ Melhorar								P1 P3 P4 P6 P7 P8 P9 P10	P5 P9	P2
		Extrapolar							P1			

(A): (U) Uniestrutural; (M) Multiestrutural; (R) Relacional; (AE) Abstrato Estendido.
(B): (D) Declarativo; (F) Funcional.

Fonte: Dados da pesquisa.

Após análise qualitativa densa e prolongada, fazendo correspondências

entre verbos de mesmo tipo de conhecimento e nível de compreensão, e buscando coerência e clareza na concepção dos resultados pretendidos de aprendizagem, o Quadro 10 apresenta os verbos de aprendizagem e tópicos de MAS eleitos para esta pesquisa, no qual se percebe uma predominância de verbos de aprendizagem de nível Relacional.

Quadro 10 – Verbos de aprendizagem e tópicos de MAS eleitos

SOLO (A)	Tipo (B)	Verbo	FM1 Princípios de Modelagem	TM1 Modelagem de Informações	TM2 Modelagem Comportam ental	TM3 Modelagem Arquitetural	TM4 Modelagem de Domínio	FA1 Análise da Construção
U	D	Identificar	X					
M	D	Descrever	X	X				
	F	Ilustrar	X	X				
R	D	Analisar		X	X			
		Explicar				X		
		Comparar/ Contrastar					X	
	F	Aplicar				X		
		Construir		X	X			
		Prever					X	
AE	D	Hipotetizar						X
	F	Refletir/ Melhorar						X

(A): (U) Uniestructural; (M) Multiestructural; (R) Relacional; (AE) Abstrato Estendido.
(B): (D) Declarativo; (F) Funcional.

Fonte: Elaboração do autor.

Sobre os Resultados Pretendidos de Aprendizagem para os tópicos de MAS, a fim de aumentar a noção de abstração no ensino de engenharia de software, e fazendo alusão à questão “C4”, foi preciso fazer uma correspondência (mapeamento) por meio do critério de agrupamento por similaridade.

Dado que o tópico de MAS, tipo de conhecimento e verbo de aprendizagem e respectivo nível SOLO foram descobertos, o pesquisador selecionou somente os RPAs passíveis de ser mapeados aos primeiros resultados. Apenas como adendo, é válido comunicar que cada participante redigiu um total de cinco resultados no decorrer da entrevista, inclusive mencionando o contexto de MAS onde o verbo deveria ser efetivado, portanto, o pesquisador tinha em mãos uma lista considerável de resultados para analisar a correspondência entre as informações.

Diante do exposto, se obteve as informações necessárias para redigir devidamente o RPA, as quais são:

- a) O tipo de conhecimento e o verbo no nível apropriado de compreensão ou desempenho pretendido;
- b) O conteúdo do tópico que o verbo deve abordar, quer dizer, o objeto do verbo;
- c) O contexto do conteúdo da disciplina (área de conhecimento de MAS, no caso desta pesquisa) em que o verbo deve ser implantado.

Após compilação e análise qualitativa dos dados (tipo de conhecimento + verbo + tópico + contexto) realizada via método indutivo, o Quadro 11 apresenta os oito Resultados Pretendidos de Aprendizagem descobertos para esta pesquisa.

Quadro 11 – Resultados pretendidos de aprendizagem descobertos

Tópico de MAS	Verbo (Tipo) (A)	Nível SOLO (B)	Os estudantes devem ser capazes de:	ID RPA
Princípios de Modelagem	Identificar (D)	U	Identificar elementos em uma representação com base nos princípios de modelagem orientada a objetos em um dado domínio de problema.	RPA1a
Princípios de Modelagem	Descrever (D) Ilustrar (F)	M	Descrever princípios de modelagem orientada a objetos e Ilustrar notações em representações conforme o domínio do problema.	RPA1b
Modelagem de Informações	Descrever (D) Ilustrar (F)	M	Descrever elementos de um modelo de informações e Ilustrar notações em representações conforme o domínio do problema.	RPA2a
Modelagem de Informações	Analisar (D) Construir (F)	R	Analisar um problema de um dado domínio de negócio e Construir modelos de informações para representar os elementos do problema analisado.	RPA2b
Modelagem Comportamental	Analisar (D) Construir (F)	R	Analisar um problema de um dado domínio de negócio e Construir modelos comportamentais para representar os elementos do problema analisado.	RPA3
Modelagem Arquitetural	Explicar (D) Aplicar (F)	R	Explicar as características de um dado padrão arquitetural e Aplicar o presente na representação de modelos em uma perspectiva	RPA4

			de arquitetura dentro do mesmo domínio de negócio.	
Modelagem de Domínio	Comparar/ Contrastar (D) Prever (F)	R	Comparar e Contrastar informações de representações distintas de um dado domínio de negócio e Prever a construção de modelos relacionados dentro do mesmo domínio.	RPA5
Análise da Construção	Hipotetizar (D) Refletir/ Melhorar (F)	AE	Refletir sobre o processo de análise, Melhorar os artefatos construídos e Hipotetizar possíveis soluções de modelagem e representação para outros domínios de negócio.	RPA6
(A): (D) Declarativo; (F) Funcional. (B): (U) Uniestructural; (M) Multiestructural; (R) Relacional; (AE) Abstrato Estendido.				

Fonte: Elaboração do autor.

Ainda sobre os RPAs, é cabível frisar que em virtude de ter havido a descoberta de dois resultados referentes ao tópico de Princípios de Modelagem (RPA1a e RPA1b), como também de dois resultados concernentes ao tópico de Modelagem de Informações (RPA2a e RPA2b), porém com verbos de aprendizagem de níveis cognitivos distintos, se decidiu manter estes quatro resultados no presente estudo.

Dando prosseguimento, é viável mostrar o mapeamento realizado entre os RPAs descobertos neste estudo (coluna) e os RPAs redigidos pelos participantes (linha) durante a coleta de dados, conforme Quadro 12, ainda mais levando em conta o uso de tal correspondência na análise das duas próximas questões.

Quadro 12 – Mapeamento entre os RPAs da pesquisa e os dos participantes

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
RPA1a		RPA1					RPA1	RPA1	RPA1	
RPA1b	RPA1		RPA1			RPA1				RPA1
RPA2a		RPA2			RPA2	RPA2			RPA2	
RPA2b	RPA2		RPA3				RPA2	RPA2		RPA2
RPA3			RPA4	RPA2			RPA3	RPA3		RPA3
RPA4				RPA3	RPA3	RPA3			RPA3	
RPA5	RPA3	RPA4						RPA4		RPA4
RPA6	RPA4		RPA5	RPA4		RPA5	RPA5	RPA5	RPA4	RPA5

Fonte: Elaboração do autor.

Referente às Atividades de Ensino e Aprendizagem (AEAs), com o intuito de incentivar o engajamento tanto do professor quanto do aluno para atingir o resultado pretendido de aprendizagem, o Quadro 13 exhibe as respostas analisadas para a questão “C5”. É importante realçar que somente foram consideradas as atividades de ensino e aprendizagem associadas aos RPAs redigidos pelos participantes cujos resultados foram previamente mapeados aos resultados deste estudo, em conformidade ao Quadro 12, exibido de antemão.

Quadro 13 – Mapeamento entre os RPAs da pesquisa e as AEAs dos participantes

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
RPA1a		ABInv (1)					ABProb (1)	ABProb (1)	ABProb (1)	
RPA1b	ABProb (1)		ABInv (1)			ABProb (2)				ABProb (3)
RPA2a		EC (1)			ABProj (1)	EC (2)			ABProb (1)	
RPA2b	EC (1)		ABInv (1)				ABProj (1)	ABProj (2)		ABProj (1)
RPA3			ABInv (1)	EC (1)			ABProj (1)	ABProb (1)		ABProj (2)
RPA4				ABProb (1)	EC (1)	ABProj (1)			ABProj (2)	
RPA5	ABProj (1)	EC (1)						ABProb (1)		EC (2)
RPA6	ABProb (1)		ABProb (2)	ABInv (1)		ABProj (1)	EC (1)	ABProj (2)	ABProj (9)	Game (1)

Legenda: (ABInv) Aprendizagem Baseada em Investigação; (ABProb) Aprendizagem Baseada em Problema; (ABProj) Aprendizagem Baseada em Projeto; (EC) Estudo de Caso; (Game) Gamificação.

Fonte: Elaboração do autor.

Ainda sobre as AEAs, cabe informar que devido à variedade de respostas em relação ao tamanho da amostra, à dinâmica do instrumento de coleta de dados e, em particular, o conhecimento e experiência de cada especialista, foi percebido certas limitações para se fazer o agrupamento por similaridade, assim sendo, este estudo optou por considerar todas as atividades respondidas pelos participantes.

Por fim, no que diz respeito às tarefas de avaliação (TAs), com a intenção de motivar o estudante nesse processo, como também permitir ao professor observar se o resultado pretendido de aprendizagem foi atingido, o Quadro 14 expõe as respostas analisadas para a questão “C6”. É importante frisar que apenas foram consideradas as tarefas de avaliação associadas aos RPAs

redigidos pelos participantes cujos resultados foram previamente mapeados aos resultados deste estudo, em conformidade ao Quadro 12, exibido anteriormente.

Quadro 14 – Mapeamento entre os RPAs da pesquisa e as TAs dos participantes

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
RPA1a		ProjI (1)					ARProb (1)	ARProb (2)	ARProb (3)	
RPA1b	PortfA (1)		ProjI (1)			ARProb (1)				ARProb (2)
RPA2a		ProjG (1)			ProjG (2)	ProjI (1)			ARProb (1)	
RPA2b	AOral (1)		ProjI (1)				APares (1)	ARProb (1)		ProjG (1)
RPA3			ProjI (1)	ARProb (1)			ProjG (1)	ARProb (2)		ProjG (2)
RPA4				ARProb (1)	ProjG (1)	ProjI (1)			ProjG (2)	
RPA5	ProjI (1)	ProjG (1)						ARProb (1)		AOral (1)
RPA6	ARProb (1)		AOral (1)	DBordo (1)		ProjG (1)	AOral (2)	ProjG (2)	ProjG (3)	AOral (3)

Legenda: (ProjI) Projetos Individuais; (ARProb) Avaliação de Resolução de Problemas; (PortfA) Portfólio de Aprendizagem; (ProjG) Projetos em Grupo; (AOral) Apresentação Oral; (APares) Avaliação por Pares; (DBordo) Diário de Bordo.

Fonte: Elaboração do autor.

Ainda sobre as TAs, vale informar que devido à variedade de respostas em relação ao tamanho da amostra à dinâmica do instrumento de coleta de dados e, particularmente, o conhecimento e experiência de cada especialista, foi notado certas limitações para se fazer o agrupamento por similaridade, assim sendo, este estudo optou também por considerar todas as tarefas de avaliação respondidas pelos participantes.

4.3 Análise dos Resultados sobre Padrões de Abstração

Nesta seção, são apresentados os resultados das perguntas da Seção D, concernentes aos Padrões de Abstração, com a intenção de adotar esses padrões para contextualizar diferentes propósitos, formas e níveis de abstração no processo de ensino e aprendizagem em Modelagem e Análise de Software (MAS). No total, são oito questões, cada qual voltada para um padrão e dividida em duas partes. Cabe lembrar que os detalhes das perguntas se encontram no Roteiro de Entrevista, disponível no Apêndice A.

Antes de apresentar os resultados, é importante ressaltar que os padrões

são provenientes (dos padrões de) das perguntas sobre abstração de Hazzan e Kramer (2006; 2016), os quais foram instanciados no contexto de MAS com o apoio das Representações de Apoio cujo artefato se encontra disponível no Anexo A. Os oito padrões estão descritos no Quadro 15.

Quadro 15 – Descrição dos padrões de abstração

<p>Padrão 1: Dadas duas representações de um sistema específico, os alunos são solicitados a explicar qual representação é mais abstrata e por quê. Essas representações podem ser o código do programa, um diagrama UML, uma descrição escrita, uma foto e assim por diante.</p>
<p>Padrão 2: Categorização de uma coleção de representações de diferentes sistemas de acordo com a escolha de abstração do estudante.</p>
<p>Padrão 3: Descrição de um sistema específico, com o qual os alunos estão familiarizados, em diferentes níveis de abstração.</p>
<p>Padrão 4: Dada uma representação do sistema, os alunos são solicitados a fornecer uma representação mais abstrata do que a fornecida e uma representação menos abstrata do que a fornecida.</p>
<p>Padrão 5: Os alunos são solicitados a explicar algum tópico (sistema, representação...) para outra pessoa. Eles são solicitados a descrever como explicariam o tópico escolhido em dois casos: quando a pessoa para quem estão explicando é capaz de ver esse tópico e quando não pode vê-lo. Também são solicitados a explicar as considerações que orientaram a formulação de cada descrição e como essas formulações se relacionam com diferentes níveis de abstração.</p>
<p>Padrão 6: Os alunos são solicitados a explicar algum tópico (sistema, representação...) com uma determinada restrição ou restrições. Os alunos também são convidados a explicar suas considerações na escolha de suas explicações.</p>
<p>Padrão 7: Os alunos são solicitados a sugerir uma metáfora ou uma analogia para um determinado sistema.</p>
<p>Padrão 8: Perguntas de reflexão: este tipo de perguntas pode ser usado em muitas situações. Eles convidam os alunos a refletir sobre seus processos de pensamento e, assim, aumentar o nível de abstração de seus processos mentais.</p>

Fonte: Adaptado de Hazzan e Kramer (2006; 2016).

Todas as oito perguntas sobre os padrões de abstração têm a mesma estrutura, isto é, duas partes.

A primeira parte (“D1a”, “D2a”, “D3a”, “D4a”, “D5a”, “D6a”, “D7a”, “D8a”) avaliou se cada padrão era adequado para contextualizar a abstração em modelagem e análise de software, em uma escala de 1 (nada adequado) a 10 (bastante adequado). A Tabela 1 mostra os resultados para os oito padrões por meio de estatística descritiva. Tendo como base a média, se nota que os padrões 3, 8, 4 e 5 foram os mais bem avaliados, porém, como a média é influenciada pelos extremos, se optou por complementar a avaliação com a mediana, o desvio padrão (DP) e o coeficiente de variação (CV), buscando assim demonstrar um conjunto de medidas para auxiliar a interpretação dos resultados.

Tabela 1– Avaliação de adequação dos padrões de abstração por participante

Padrão <i>versus</i> Participante	Padrão 1	Padrão 2	Padrão 3	Padrão 4	Padrão 5	Padrão 6	Padrão 7	Padrão 8
P1	8	6	7	7	7	7	7	6
P2	9	7	9	7	9	7	9	10
P3	7	7	8	8	9	9	6	8
P4	6	3	8	6	8	9	8	4
P5	10	7	10	10	8	8	7	10
P6	7	6	10	7	6	5	6	10
P7	5	8	10	10	9	10	10	10
P8	8	9	10	10	6	6	10	10
P9	8	7	6	8	9	8	8	9
P10	10	8	9	10	10	5	9	10
Total	10	10	10	10	10	10	10	10
Média	7,8	6,8	8,7	8,3	8,1	7,4	8	8,7
Mediana	8	7	9	8	8,5	7,5	8	10
DP	1,619328	1,619328	1,418136	1,567021	1,37032	1,712698	1,490712	2,110819
CV (%)	20,76061	23,81364	16,30042	18,87977	16,91753	23,14456	18,6339	24,26228

Fonte: Dados da pesquisa.

A segunda parte (“D1b”, “D2b”, “D3b”, “D4b”, “D5b”, “D6b”, “D7b”, “D8b”), por sua vez, verificou se o padrão poderia ser instanciado para ajudar a atingir algum dos resultados pretendidos de aprendizagem descobertos e redigidos preliminarmente na Seção 4.2. O Quadro 16 exhibe o mapeamento entre os RPAs eleitos para esta pesquisa e os oito padrões de abstração a partir da análise das respostas dos participantes. No que tange aos padrões (linha), se percebe que o

Padrão 8 (27x) foi o que apresentou mais ocorrências, seguido na ordem, pelo Padrão 4 (26x), Padrão 3 (23x), Padrão 5 (22x), Padrão 2 e 7 (20x), Padrão 1 (16x) e Padrão 6 (13x). Acerca dos RPAs (coluna), se nota que o RPA2b (31x) foi o que apresentou mais ocorrências, seguido na ordem, pelo RPA3 (29x), RPA6 (25x), RPA5 (22x), RPA1b (15x), RPA2a (14x), RPA4 (13x) e RPA1a (12x). Na ótica do cruzamento dos dados, se observa que a interseção RPA6 X Padrão 8 (7x) foi a que apresentou mais correspondências, seguidamente das interseções RPA6 X Padrão 4 (5x), RPA2b X Padrão 5 (5x), RPA3 X Padrão 2 (5x) e RPA3 X Padrão 4 (5x); os demais cruzamentos obtiveram quatro correspondências ou menos.

Quadro 16 – Mapeamento entre RPAs da pesquisa e padrões de abstração por participante

Padrão <i>versus</i> RPA	Padrão 1 (16x)	Padrão 2 (20x)	Padrão 3 (23x)	Padrão 4 (26x)	Padrão 5 (22x)	Padrão 6 (13x)	Padrão 7 (20x)	Padrão 8 (27x)
RPA1a (12x)	P7 P8 P9	P2	P2	P2	P2 P8	P8	P2 P7	P2
RPA1b (15x)	P6 P10		P1	P6	P3 P10	P1 P3 P6	P3 P6 P10	P1 P6 P10
RPA2a (14x)	P5	P2 P5 P6 P9	P2	P2 P5	P2	P6	P2	P2 P5 P6
RPA2b (31x)	P1 P3 P8 P10	P3 P7 P8 P10	P1 P3 P7 P10	P3 P7 P8 P10	P1 P3 P7 P8 P10	P1 P3 P8	P3 P7 P8 P10	P3 P8 P10
RPA3 (29x)	P3 P10	P3 P4 P7 P8 P10	P3 P7 P8 P10	P3 P4 P7 P8 P10	P3 P7 P8 P10	P3 P8	P3 P7 P8	P3 P4 P8 P10
RPA4 (13x)	P5	P4	P4 P5 P9	P5 P6 P9	P4		P4	P4 P5 P6

RPA5 (22x)	P1 P8	P1 P2 P8 P10	P1 P2 P8 P10	P1 P2 P8 P10	P1 P2 P10		P1 P2 P10	P8 P10
RPA6 (25x)	P4		P1 P3 P6 P8	P3 P4 P8 P9 P10	P4 P9 P10	P3 P4 P9	P3 P4	P1 P3 P6 P7 P8 P9 P10

Fonte: Elaboração do autor.

Ainda sobre os resultados apresentados na Tabela 1 e no Quadro 16, cabe salientar que todos os padrões foram adotados no presente estudo, entretanto, mediante análise e interpretação dos dados, é plausível observar quais padrões lograram melhor avaliação, bem como os padrões de maior correspondência com cada RPA. Em todo o caso, se pretende aqui evidenciar um grau de importância do uso dos padrões de abstração para contribuir com os resultados pretendidos de aprendizagem e, reciprocamente, melhorar a capacidade cognitiva de abstração.

4.4 Análise dos Resultados sobre Itens de Conhecimento e Erros de Abstração

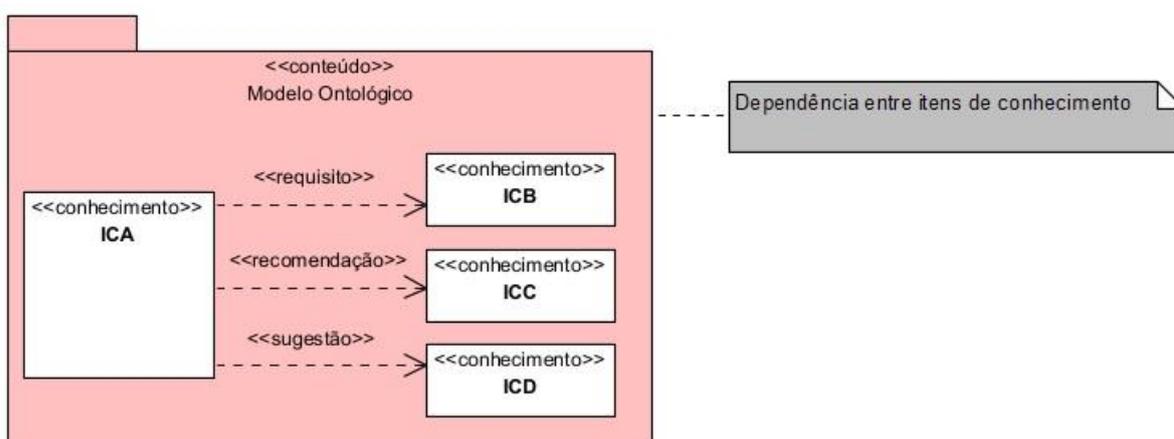
Nesta seção, são apresentados os resultados das perguntas da Seção E, as quais se referem aos Itens de Conhecimento e Erros de Abstração. No total, são duas questões cujas respostas são difundidas em seguida. A primeira pretende identificar a rede de conteúdos apropriada para se atingir cada RPA originado nesta pesquisa. A segunda, por sua vez, avalia coletivamente um conjunto de quatro erros de abstração, com o intuito de verificar a chance dessa lista se tornar um novo padrão de abstração. Vale lembrar que os detalhes das perguntas se encontram no Roteiro de Entrevista, disponível no Apêndice A.

No que diz respeito aos itens de conhecimento (IC) para se atingir cada RPA, sobretudo com enfoque no tópico deste último, e fazendo alusão à questão “E1”, foi adotada a estrutura usual de relações entre itens de conhecimento baseada

no plano organizacional de conteúdos¹³ da técnica OC2-RD2 (VEGA, 2018), conforme mostra a Figura 17, no qual se observa os três tipos de dependências (relacionamentos estereotipados), assim sendo:

- a) Requisito: essa relação, estereotipada como <<requisito>>, define os pré-requisitos para se adquirir conhecimento de um determinado conteúdo, melhor dizendo, para compreender o Item de Conhecimento “A” (ICA), é requerido ter a compreensão do Item de Conhecimento “B” (ICB);
- b) Recomendação: essa relação, estereotipada como <<recomendação>>, indica recomendações de itens de conhecimento que visem auxiliar a compreensão de um certo conteúdo, em outras palavras, para compreender o Item de Conhecimento “A”, se recomenda a compressão do Item de Conhecimento “C” (ICC);
- c) Sugestão: essa relação, estereotipada como <<sugestão>>, indica eventuais complementações de estudo que possam ajudar na compreensão de um dado conteúdo, quer dizer, para compreender o Item de Conhecimento “A” (ICA), se sugere o estudo do Item de Conhecimento “D” (ICD).

Figura 17 – Estrutura de dependências no plano de conteúdos OC2-RD2



Fonte: Adaptado de Vega (2018, p. 18).

Após análise qualitativa densa e prolongada, fazendo correspondências por

¹³ Coincidentemente ao termo ontológico adotado neste trabalho, o plano organizacional de conteúdos é denominado modelo ontológico na técnica OC2-RD2.

meio do critério de agrupamento por similaridade entre os diversos tópicos julgados pelos participantes e buscando preservar a ontologia da área de conhecimento de MAS, o Quadro 17 apresenta a compilação dos conteúdos, separados como requisito, recomendação e sugestão. Vale comunicar que um dos participantes optou por não responder à questão “E1”.

Quadro 17 – Mapeamento entre RPA e Itens de Conhecimento

RPA	Tópico	Itens de Conhecimento		
		Requisito	Recomendação	Sugestão
RPA1a	Princípios de Modelagem	Paradigma OO; Fundamentos de Sistemas de Informação	UML; Lógica de Programação	Modelagem de Processos de Negócio Estrutura de Dados
RPA1b	Princípios de Modelagem	Paradigma OO; Modelagem OO; Engenharia de Requisitos	UML	Modelagem de Processos de Negócio; Análise Essencial e suas representações
RPA2a	Modelagem de Informações	Modelagem OO; Modelagem de Dados	UML; DER	Álgebra Relacional; Notação Formal
RPA2b	Modelagem de Informações	Modelagem OO; Modelagem de Dados; Engenharia de Requisitos	UML; DER	Álgebra Relacional; Notação Formal
RPA3	Modelagem Comportamental	Modelagem OO; Engenharia de Requisitos; Modelagem de Processos de Negócio	UML; BPMN	Implementação OO; Estruturas de Dados
RPA4	Modelagem Arquitetural	APOO; Engenharia de Requisitos; Padrões Arquiteturais	Modelo de Visões 4+1; UML; Princípios SOLID	Padrões de Arquitetura de Aplicações Corporativas
RPA5	Modelagem de Domínio	Modelagem OO; Engenharia de Requisitos; Modelagem de Processos de Negócio	UML; BPMN	Notação Formal; Padrões de Domínio (modelo de referência)
RPA6	Análise da Construção	Engenharia de Requisitos; Inspeção e Revisão de Software; Modelagem OO	UML; APOO (refinamento dos modelos); Padrões de Software (Projeto, Arquitetura, Domínio)	Técnicas OORT; Taxonomia de Defeitos; Notação Formal

Fonte: Elaboração do autor.

Ainda sobre o mapeamento entre cada RPA e os Itens de Conhecimento,

cabe antecipar que este trabalho vai apresentar mais adiante na Seção 5.2.3 as representações diagramáticas com suporte da UML para cada Modelo Ontológico (Plano de Conteúdos), como também a incorporação de cada um destes planos na arquitetura do modelo pedagógico proposto.

Quanto ao conjunto de erros de abstração extraídos de Halladay e Wiebel (1993) e fazendo referência à questão "E2", os participantes avaliaram se estes eram adequados para contextualizar a abstração no ensino de modelagem e análise de software e, conseqüentemente, contribuir para atingir os resultados pretendidos de aprendizagem em uma escala de 1 (nada adequado) a 10 (bastante adequado). Esses erros também foram instanciados no contexto de MAS com o apoio das Representações de Apoio cujo artefato se encontra no Anexo A. Os quatro erros comuns de abstração estão descritos no Quadro 18.

Quadro 18 – Quatro erros comuns de abstração

- | |
|---|
| <p>a) Particionar elementos illogicamente para um nível de abstração;</p> <p>b) Reduzir ou pular níveis de abstração;</p> <p>c) Não identificar todos os elementos para um nível de abstração;</p> <p>d) Identificar elementos inexistentes para um nível de abstração.</p> |
|---|

Fonte: Adaptado de Halladay e Wiebel (1993).

A Tabela 2 mostra os resultados da avaliação do catálogo de erros de abstração mencionados via estatística descritiva, na qual se observa medidas como média, mediana, desvio padrão (DP) e coeficiente de variação (CV), visando assim ajudar na interpretação dos resultados. Ademais, cabe informar que um dos participantes optou por não responder à esta questão.

Tabela 2 – Avaliação de adequação dos erros de abstração

Participante	Erros de Abstração
P1	9
P2	10
P3	9
P4	8
P5	10
P6	4
P7	–
P8	10
P9	10
P10	10
Total	9
Média	8,9
Mediana	10
DP	1,96497102
CV (%)	22,10592398

Fonte: Dados da pesquisa.

A análise das respostas da avaliação dessa lista de erros de abstração, assim como a comparação destes números aos resultados dos padrões de abstração avaliados na Seção 4.3, se percebe uma avaliação bastante positiva, o que levou este estudo a considerar esse conjunto de erros como um novo padrão de abstração.

Além do mais, convém reforçar aqui a importância deste novo padrão de abstração, uma vez que o erro faz parte do processo de aquisição de conhecimento, havendo assim consonância à teoria de aprendizagem neopiagetiana e ao paradigma construtivista, conforme mencionado em Biggs e Collis (1982), Becker (2012), Biggs e Tang (2011), como também está alinhada ao aprendizado por meio de erros de modelagem, apontado no trabalho de Chourio *et al.* (2019).

4.5 Análise dos Resultados sobre o Modelo Pedagógico Proposto

Nesta seção, são apresentados os resultados das perguntas da Seção F, alusivas ao modelo pedagógico de correlação proposto, a fim de identificar impressões iniciais a respeito da proposição apresentada por este estudo. Ao todo

são quatro questões cujos enunciados são exibidos no Quadro 19. Vale recordar que os detalhes das perguntas se encontram no Roteiro de Entrevista, disponível no Apêndice A.

Quadro 19 – Enunciados das perguntas sobre o modelo proposto

F1) A existência de um modelo pedagógico que apresente planos de conteúdos temáticos e reutilizáveis de modelagem e análise de software o qual busca auxiliar o professor na elaboração de planos de disciplina e na preparação de aulas pode a vir se tornar um recurso educacional adequado para o planejamento de ensino de engenharia de software?
F2) A existência de um modelo pedagógico que direcione atividades de ensino e aprendizagem e tarefas de avaliação para atingir resultados pretendidos de aprendizagem em relação aos tópicos de modelagem e análise de software pode a vir se tornar um recurso educacional adequado para o planejamento de ensino de engenharia de software?
F3) A existência de um modelo pedagógico que presume a contextualização da abstração em modelagem e análise de software, oferecendo lentes de abstração distintas na abordagem dos conteúdos pode a vir se tornar um recurso educacional adequado para o planejamento de ensino de engenharia de software?
F4) Uma fábula, comumente narrada por meio de uma história, costuma mexer com a emoção das pessoas enquanto é contada para um determinado público. A narrativa de uma história em sala de aula poderia causar emoção ao estudante, em outras palavras, provocar o desequilíbrio/equilíbrio cognitivo do sujeito que aprende, mediante o grau de aprendizagem durante a narração da história, onde essa equilibração oscilaria entre a presença e a ausência de conhecimento sobre o tema abordado. Tal narração pode ser traduzida como, usando aqui palavras de Becker (2012), um jogo cognitivo de assimilações e acomodações. Posto isto, a existência de um modelo pedagógico que apresente dimensões de conteúdo e de cognição pode ser um recurso educacional adequado para apoiar o planejamento de roteiros narrativos no ensino de engenharia de software?

Fonte: Elaboração do autor.

Os participantes avaliaram a proposição do modelo pedagógico avistando tal proposta se concretizando no futuro como um recurso educacional para o planejamento de ensino de engenharia de software. Cada questão, possivelmente vista como uma premissa aqui, foi avaliada, em termos de adequação, em uma escala de 1 (nada adequado) a 10 (bastante adequado). A Tabela 3 mostra os resultados das quatro questões via estatística descritiva, na qual se nota medidas como média, mediana, desvio padrão (DP) e coeficiente de variação (CV), para auxiliar a interpretação dos resultados. Ademais, cabe comunicar que um dos participantes não respondeu à questão F4.

Tabela 3 – Avaliação de adequação do modelo pedagógico proposto

Pergunta versus Participante	F1	F2	F3	F4
P1	9	9	9	8
P2	10	10	10	9
P3	9	9	8	7
P4	8	7	8	6
P5	9	10	8	5
P6	10	7	7	5
P7	10	10	9	–
P8	10	10	10	10
P9	10	10	10	10
P10	10	10	10	10
Total	10	10	10	9
Média	9,5	9,2	8,9	7,8
Mediana	10	10	9	8
DP	0,707106781	1,229272594	1,100504935	2,108185107
CV (%)	7,443229276	13,36165863	12,36522398	27,10523709

Fonte: Dados da pesquisa.

Ainda sobre os resultados do modelo pedagógico proposto, se nota uma avaliação significativamente positiva, revelando indícios que motivaram ainda mais a concretização da referida proposição cujos detalhes de construção são apresentados no Capítulo 5.

Além das quatro questões já mencionadas, ainda é cabível informar que, ao final da entrevista, os participantes poderiam, se assim desejassem, fazer alguma contribuição adicional para viabilizar a proposta do modelo pedagógico, assim como sugerir melhoria para as perguntas apresentadas.

4.6 Validade e Confiabilidade Qualitativa dos Resultados

Esta seção apresenta a validade e confiabilidade dos resultados da pesquisa em relação ao procedimento de pesquisa adotado.

No que concerne a validade dos resultados, este estudo seguiu recomendações propostas pela literatura buscando assim assegurar a precisão

dos resultados, levando em consideração pontos de vista do pesquisador, dos participantes e dos leitores.

De acordo com Creswell (2010), mesmo havendo validação durante todas as etapas do instrumento de pesquisa, este processo está presente principalmente na etapa de validade e confiabilidade dos resultados, a fim de possibilitar que o pesquisador adote procedimentos para validar os resultados a serem utilizados em uma determinada pesquisa. Esse autor ainda salienta que a validade na pesquisa qualitativa não leva as mesmas conotações aplicadas na pesquisa quantitativa e que o estudo qualitativo também não é um aliado da confiabilidade.

Ainda segundo Creswell (2010), a validade é um dos pontos fortes da pesquisa qualitativa porque permite determinar se os resultados são precisos sob a ótica do pesquisador, do participante ou dos leitores de um relato. Uma orientação é que o pesquisador adote em seu estudo certas estratégias de validade para melhorar a capacidade de avaliação da precisão dos resultados, assim como para convencer os leitores dessa certeza. Dentre as estratégias recomendadas por esse autor, seguem as adotadas neste estudo, buscando, desta maneira, aumentar a validade dos resultados:

- a) Triangulação de fontes de informação distintas examinando as evidências das fontes e utilizando-as para construir uma justificativa coerente para o estudo. Este trabalho se baseou nas evidências coletadas de 10 participantes considerados pelo pesquisador como especialistas na temática;
- b) Descrição rica e densa para comunicar os resultados. Os quadros, tabelas e figuras utilizadas no registro dos dados, assim como na análise e interpretação dos resultados foram apresentados de forma detalhada;
- c) Esclarecimento do viés de pesquisa trazido para o estudo para possibilitar que o pesquisador faça uma reflexão e registre seus comentários sobre a análise e interpretação dos resultados. O pesquisador procurou ser cauteloso ao fazer seus próprios comentários no decorrer das etapas de análise e interpretação dos resultados;
- d) Apresentação de informações negativas ou discrepantes para contradizer a perspectiva geral a respeito do tema estudado. As ocorrências contraditórias corroboraram para que o estudo se tornasse ainda mais real

e, conseqüentemente, válido;

- e) Tempo prolongado no campo para permitir que haja uma compreensão profunda do fenômeno estudado, informando detalhes sobre as pessoas que conferem credibilidade ao estudo. Neste estudo, apesar das entrevistas terem sido conduzidas remotamente, o procedimento de coleta de dados foi consideravelmente longo. Apenas como adendo, vale lembrar que a entrevista mais curta teve a duração aproximada de 1h30min, a mais longa durou cerca de 2h40min e a duração média das 10 entrevistas foi de 2h15min. Dos 10 participantes, sete optaram por participar da entrevista em duas sessões, enquanto outros três responderam as questões em uma única sessão. A dinâmica, formato e duração das entrevistas permitiram ao pesquisador passar um tempo duradouro junto aos entrevistados.

No que diz respeito à confiabilidade dos resultados, este estudo se deparou com determinadas limitações para garantir a consistência das respostas por se tratar de um estudo de cunho qualitativo. Mesmo havendo restrições, este trabalho também seguiu recomendações propostas pela literatura com o intuito de assegurar a densidade dos resultados.

De acordo com Gray (2010), um instrumento de pesquisa é confiável quando é possível medir de maneira consistente o que se pretende medir. Em relação à entrevista, há pelo menos algum potencial para garantir essa estabilidade, melhor dizendo, quando uma entrevista é padronizada e se faz as mesmas perguntas para cada participante, evitando assim o chamado “efeito entrevistador”, onde o viés de quem entrevista se traduz em risco de interferir no instrumento de coleta de dados de formas variadas, algumas mais sutis e outras nem tanto. A única forma de evitar esse erro é padronizando a entrevista e o comportamento do entrevistador. Nessa direção, dentre as estratégias recomendadas pelo autor predito, seguem as adotadas neste trabalho, buscando, deste modo, aumentar a confiabilidade dos resultados:

- a) Desvio das instruções da entrevista. As instruções foram explicadas para todos os participantes tomarem ciência a respeito do roteiro de entrevista, impedindo assim a possibilidade de desvios;
- b) Sintonia ruim entre entrevistador e entrevistado. A todo o momento e em todas as entrevistas, o pesquisador buscou uma sinergia com os

participantes, aliás se ressalta a presença de colaboração intensa por parte dos entrevistados;

- c) Modificação das perguntas. Nenhuma indagação foi modificada ao longo da entrevista. As perguntas foram feitas da forma em que estavam redigidas;
- d) Reformulação das perguntas. É verossímil que determinadas questões foram mais complicadas para os entrevistados compreenderem. Em certas situações, o pesquisador explicou o propósito, forma e contexto da indagação por mais de uma vez, mas sem modificar a pergunta durante a entrevista;
- e) Descuido no fornecimento de estímulos. Sempre que possível, o pesquisador buscou ter cautela para não estimular indevidamente os entrevistados;
- f) Questões de aprofundamento com muito viés. Mesmo nas perguntas que necessitavam de reflexão e aprofundamento maiores, o pesquisador evitou que as perguntas fossem realizadas com algum tipo de viés;
- g) Realização de perguntas fora da sequência. Por se tratar de uma modalidade semiestruturada de entrevista, a dinâmica não permitiu realizar as perguntas de modo sequencial em todas as entrevistas. Mesmo assim, cabe informar que, ao menos dentro da mesma seção, a ordem das questões foi devidamente seguida;
- h) Registro de respostas com bastante viés. As respostas foram confirmadas junto aos entrevistados, buscando assim evitar registrar respostas viesadas.

Ainda visando resultados consistentes e confiáveis, de acordo com Creswell (2010), é fundamental que os pesquisadores qualitativos documentem o máximo de etapas dos instrumentos de pesquisa. Nesse caminho, as etapas do procedimento de pesquisa empregado neste trabalho foram documentadas de maneira detalhada, conforme se observou no decorrer do registro, análise e interpretação dos resultados.

5 MODELO PEDAGÓGICO PROPOSTO

“La connaissance progresse en intégrant en elle l'incertitude, non en l'exorcisant.”

(Edgar Morin)

Este capítulo apresenta a construção e validação do modelo pedagógico proposto neste trabalho. Inicialmente, o modelo é caracterizado no contexto deste estudo. Depois disto, o processo para se construir o modelo em estudo é explicado em pormenores, resultando um conjunto de diagramas UML que representam a arquitetura pedagógica do modelo-alvo; tais representações diagramáticas foram modeladas por meio da ferramenta Visual Paradigm¹⁴. Em seguida, com o intuito de fazer uma validação preliminar, é demonstrada, à título de exemplo, a aplicação do modelo em quatro planos de disciplina cujas informações foram extraídas de dois projetos pedagógicos de cursos superiores de tecnologia distintos.

5.1 Caracterização do Modelo Pedagógico

Mediante a análise, interpretação e validade dos resultados provenientes da pesquisa, se originou o modelo pedagógico denominado Modelo de Planos Ontológicos de Modelagem e Análise de Software (MPO_MAS). Apesar do termo modelo ser adotado no decurso desta tese, a estrutura da proposição se caracteriza também como metamodelo, podendo ampliar essa denominação, até mesmo, para metametamodelo

O modelo em foco, por que não metamodelo, é direcionado para educadores de cursos de computação, o qual oferece um conjunto de planos de conteúdo e cognição, temáticos e reutilizáveis, da área de conhecimento de Modelagem e Análise de Software, elaborados sob a perspectiva docente, e incorporados em uma arquitetura pedagógica alinhada construtivamente, a fim de nortear o professor, do ponto de vista de desenvolvimento cognitivo e progressão do aprendizado, no planejamento de ensino (cursos e aulas) de engenharia de software e/ou disciplinas correlatas conforme resultados pretendidos de aprendizagem identificados a partir dos objetivos de aprendizagem do componente curricular em questão. Além disto, este

14 Visual Paradigm Community Edition. Disponível em <https://www.visual-paradigm.com/download/community.jsp>.

modelo busca oferecer recursos pedagógicos para reforçar a contextualização da abstração, aspirando assim a elevação dessa capacidade cognitiva.

Dentre os diversos conceitos sobre modelo encontrados na literatura, se traz aqui a conceituação de Behar, Passerino e Bernardi (2007).

A ciência busca através de leis, princípios e modelos generalizar e simplificar a realidade. O conceito de modelo surge, portanto, com o viés de estabelecer uma relação por analogia com a realidade. O modelo é um sistema figurativo que reproduz a realidade de forma mais abstrata, quase esquemática e que serve de referência. (BEHAR; PASSERINO; BERNARDI, 2007, p. 3).

No âmbito educacional e em consonância com este trabalho, é importante realçar o conceito de modelo para uma melhor compreensão da proposta deste estudo.

Na educação o conceito de modelo foi erroneamente considerado sinônimo de teorias de aprendizagem como as desenvolvidas por Piaget, Vygotsky, Wallon, Roger, entre outros ou como metodologia de ensino. Embora um modelo pedagógico possa ser embasado numa ou mais teorias de aprendizagem, de forma geral os modelos são “reinterpretações” de teorias a partir de concepções individuais dos professores que se apropriam parcial ou totalmente de tais construtos teóricos imbuídos num paradigma vigente. Desta forma, o modelo construído muitas vezes recebe o nome de uma teoria (piagetiana, rogeriana, etc.) ou de um paradigma (construtivista, interacionista, etc.) sem contanto, ter propriamente sua epistemologia embasada nos mesmos paradigmas ou teorias mencionados. (BEHAR; PASSERINO; BERNARDI, 2007, p. 3).

Em uma perspectiva epistemológica, o modelo foi inspirado na teoria de aprendizagem piagetiana, como também no paradigma construtivista. Sob um olhar mais específico, houve forte influência de uma abordagem neopiagetiana de desenvolvimento cognitivo para o ensino de adultos em novos contextos de compreensão, suportada, em sua grande parte, pela taxonomia SOLO.

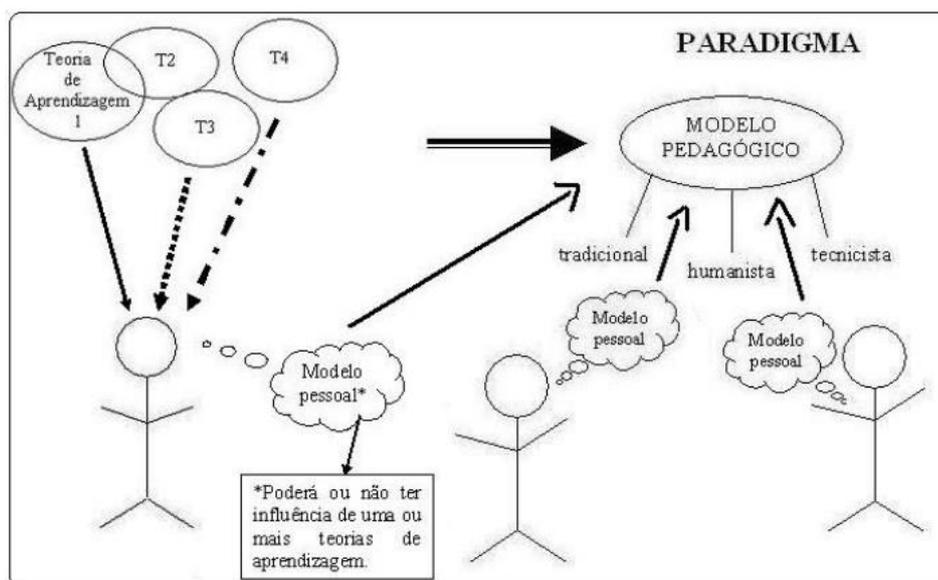
Apesar do modelo proposto estar focado no ensino, Behar, Passerino e Bernardi (2007), salientam que embora o termo modelo pedagógico seja frequentemente interpretado como metodologia de ensino, que por sinal este é

um dos elementos inerentes ao mesmo, essa redução na interpretação do modelo à sua parte visível ignora outros recursos/componentes que o constituem e que são fundamentais de serem explicitados para a compreensão do processo educativo.

Becker (2012) alerta também que é comum ver no âmbito escolar interpretações que reduzem o construtivismo, com exemplos do tipo: não se pode mais aplicar prova, não se deve memorizar mais nada, não há mais conteúdo a ser passado ao estudante, não há mais cálculo nem gramática, não há mais ensino etc. Para o autor, quem enxerga o construtivismo dessa maneira, distorce a proposta construtivista, quer dizer, descaracteriza a contribuição da epistemologia genética voltada à educação, a qual busca, na verdade, compreender e realizar o sujeito.

Ainda segundo Behar, Passerino e Bernardi (2007), um modelo pedagógico é construído a partir de um paradigma dominante que, em geral influencia as teorias de aprendizagem vigentes, assim como outras teorias científicas. A partir deste, os sujeitos constroem um modelo pessoal próprio que é compartilhado com os pares gerando, assim, um modelo pedagógico compartilhado, como mostra a Figura 18.

Figura 18 – Construção de modelos pedagógicos



Fonte: Behar, Passerino e Bernardi (2007, p. 4).

O MPO_MAS é um modelo pessoal (mental), construído com base no

paradigma construtivista e em teorias de aprendizagem neopiagetianas, além dos resultados oriundos coletados a partir da perspectiva docente, considerando as particularidades do contexto deste trabalho. Uma vez existente, a intenção é compartilhá-lo com outros professores que, aliás, também têm mentalmente seus modelos pessoais.

5.2 Construção do Modelo

O modelo proposto foi idealizado a partir da organização das etapas do alinhamento construtivo e conexão destas aos planos organizacionais da técnica OC2-RD2, acarretando uma correlação suportada pelos verbos de aprendizagem da taxonomia SOLO. A incorporação de padrões de abstração ao modelo em estudo também fazia parte desta idealização. Não menos importante, os resultados da pesquisa, principalmente os oriundos das perguntas da Seção F do Roteiro de Entrevista, revelaram indícios que motivaram a construção do modelo pedagógico.

O processo de construção para se chegar ao modelo pedagógico pessoal em estudo seguiu as seguintes etapas:

- 1) Correlação entre as etapas substanciais do Alinhamento Construtivo e os planos organizacionais da técnica OC2-RD2, sobretudo o Plano de Conteúdos;
- 2) Concepção dos Módulos de Alinhamento Construtivo, formado pelos Resultados Pretendidos de Aprendizagem, Atividades de Ensino e Aprendizagem e Tarefas de Avaliação;
- 3) Concepção do Módulo de Modelo Ontológico (ou de Plano de Conteúdos);
- 4) Concepção do Módulo de Taxonomia SOLO;
- 5) Concepção do Módulo de Padrões de Abstração;
- 6) Definição dos Planos Ontológicos;
- 7) Mapeamento dos Planos Ontológicos para Desempenhos Cognitivos de Abstração;
- 8) Design Arquitetural do Modelo.

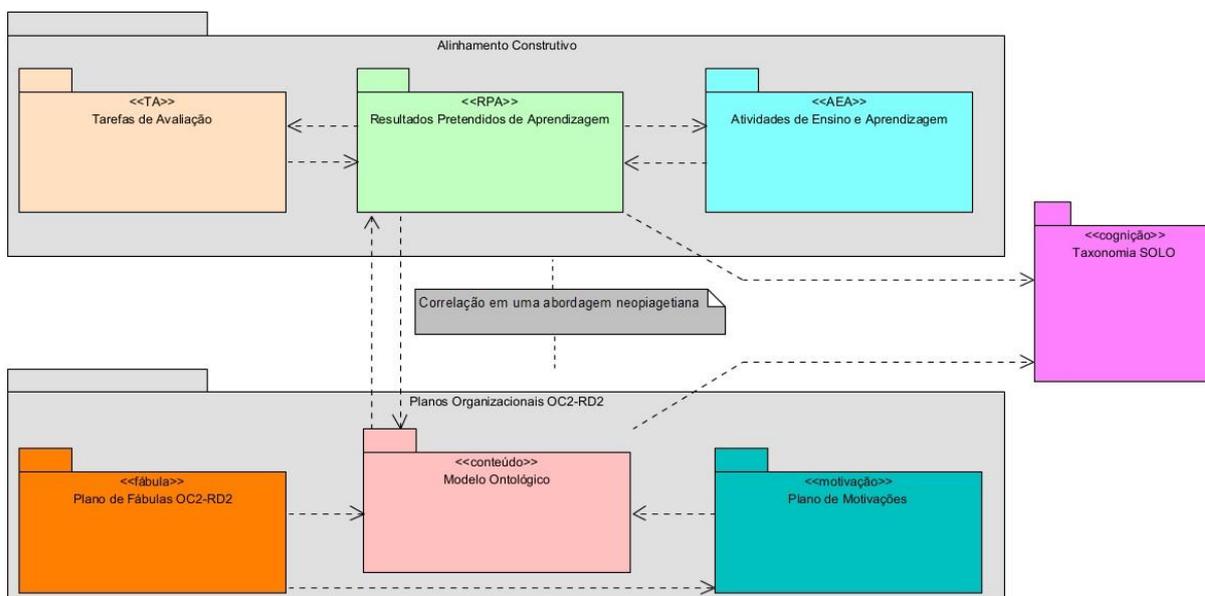
Após o preâmbulo do modelo-alvo deste trabalho, cada uma das etapas de construção é detalhada na sequência, divididas em subseções.

5.2.1. Correlação entre o Alinhamento Construtivo e a Técnica OC2-RD2

Após a investigação da teoria do Alinhamento Construtivo (AC) e das estruturas conceituais da técnica de narrativas OC2-RD2, se verificou a possibilidade de correlação entre as etapas substanciais do AL e os planos organizacionais OC2-RD2, inclusive em uma perspectiva neopiagetiana direcionada ao ensino de adultos universitários, dado que ambas as abordagens pedagógicas comumente são aplicadas em cursos superiores.

A correlação é exibida na Figura 19, na qual se nota as três etapas do AC e os três planos OC2-RD2 representados como pacotes estereotipados. O cerne da correlação é representado por uma relação bidirecional entre os Resultados Pretendidos de Aprendizagem e o Modelo Ontológico (Plano de Conteúdos), visto que a descrição de um RPA destaca o tópico a ser aprendido e o Plano de Conteúdos OC2-RD2 relaciona as dependências entre os itens de conhecimento para a aprendizagem desse tópico. Esta correlação é suportada cognitivamente pelos verbos de aprendizagem e níveis cognitivos da taxonomia SOLO, visto que, originalmente, o AC se apoia nessa taxonomia; a técnica OC2-RD2, por sua vez, também é apoiada, não exclusivamente, pela SOLO.

Figura 19 – Correlação entre o alinhamento construtivo e a técnica OC2-RD2



Fonte: Elaboração do autor.

No que tange às etapas fundamentais do AC, cada RPA estabelece o resultado almejavél, conforme nível de compreensão SOLO, para que o estudante

atinja sobre determinado tema. Cada AEA define as atividades que o estudante deve realizar para atingir o RPA e, cada TA, por sua parte, indica as atividades de avaliação para averiguar se o estudante atingiu o RPA.

A respeito dos planos organizacionais OC2-RD2, o Modelo Ontológico (Plano de Conteúdos) contém a rede de itens de conhecimento a serem adquiridos pelo aprendiz na construção do conhecimento de certo tema, o Plano de Motivações contempla elementos que buscam despertar o interesse do estudante contextualizando a aplicação de conteúdos em situações-exemplo, e o Plano de Fábulas, por sua vez, se trata de um plano adicional cujo propósito é elaborar as cenas e a relação destas para estabelecer o futuro fluxo narrativo central das histórias. Além do planejamento de ensino, esses planos oferecem condições para se elaborar as avaliações de acordo com o nível cognitivo SOLO, constatando assim se o estudante construiu o conhecimento sobre o referido tema.

Cabe enfatizar que os Resultados Pretendidos de Aprendizagem, as Atividades de Ensino e Aprendizagem e as Tarefas de Avaliação são alinhados construtivamente, assim como se presume aqui que o Modelo Ontológico também é alinhado construtivamente ao Plano de Motivações e ao Plano de Fábulas. Da mesma forma que as AEA's e TA's são planejadas em função dos RPA's, os Planos de Motivações e de Fábulas são elaborados conforme o Plano de Conteúdos, portanto, esses alinhamentos seguem uma perspectiva (paradigma) construtivista acerca do processo de ensino e aprendizagem, incluindo o processo de avaliação.

A intenção desta correlação é auxiliar no planejamento de cursos e aulas, norteando tanto as atividades do professor quanto as do aluno, buscando assim um engajamento conjunto para atingir o RPA e, desse modo, vislumbrando uma aprendizagem ativa e efetiva e, por consequência, uma construção de conhecimento mais profunda, qualitativa e duradoura. Ademais, é importante salientar que tanto o AC quanto a OC2-RD2 dão enfoque na aprendizagem centrada no aluno, e o estágio cognitivo de momento é marcado como o ponto de partida.

A correlação resultou em um conjunto de seis módulos, cada qual com o seu propósito de contribuição para com o modelo proposto. Esses módulos são apresentados na sequência deste trabalho.

5.2.2. Concepção dos Módulos de Alinhamento Construtivo

Após a correlação entre as etapas substanciais do alinhamento construtivo e os planos organizacionais OC2-RD2, houve o prosseguimento da construção do modelo pedagógico.

Os resultados da pesquisa, principalmente os oriundos das perguntas da Seção C do Roteiro de Entrevista, originaram oito Resultados Pretendidos de Aprendizagem, conforme descreve o Quadro 20. Observe que a redação de cada RPA inclui o verbo de aprendizagem, o tipo de conhecimento, o tópico a ser aprendido e um possível contexto, ou seja, “tipo de conhecimento + verbo + tópico + contexto”.

Quadro 20– Descrição dos Resultados Pretendidos de Aprendizagem

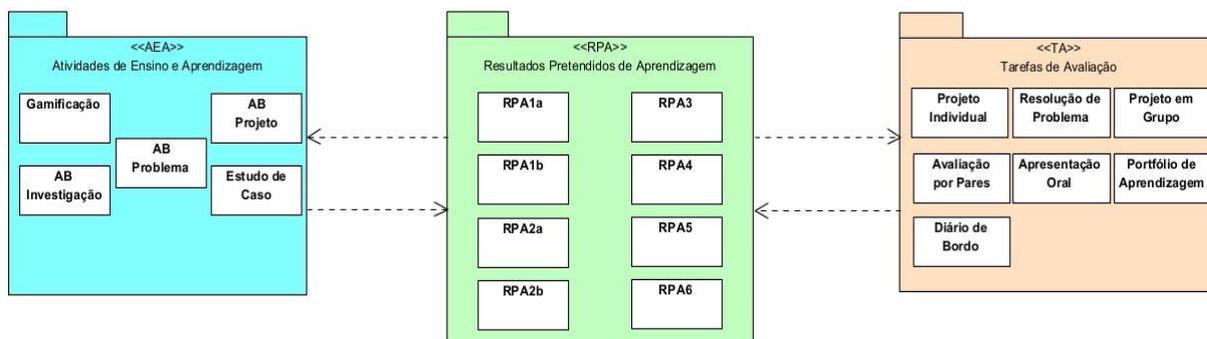
RPA	Descrição
RPA1a	Identificar elementos em uma representação com base nos princípios de modelagem orientada a objetos em um dado domínio de problema.
RPA1b	Descrever princípios de modelagem orientada a objetos e Ilustrar notações em representações conforme o domínio do problema.
RPA2a	Descrever elementos de um modelo de informações e Ilustrar notações em representações conforme o domínio do problema.
RPA2b	Analisar um problema de um dado domínio de negócio e Construir modelos de informações para representar os elementos do problema analisado.
RPA3	Analisar um problema de um dado domínio de negócio e Construir modelos comportamentais para representar os elementos do problema analisado.
RPA4	Explicar as características de um dado padrão arquitetural e Aplicar o presente na representação de modelos em uma perspectiva de arquitetura dentro do mesmo domínio de negócio.
RPA5	Comparar e Contrastar informações de representações distintas de um dado domínio de negócio e Prever a construção de modelos relacionados dentro do mesmo domínio.
RPA6	Refletir sobre o processo de análise, Melhorar os artefatos construídos e Hipotetizar possíveis soluções de modelagem e representação para outros domínios de negócio.

Fonte: Elaboração do autor.

Tais resultados também possibilitaram a concepção dos três módulos referentes ao alinhamento construtivo, os quais são representados por pacotes estereotipados na Figura 20, na qual se percebe o pacote <<RPA>> composto por oito Resultados Pretendidos de Aprendizagem, o pacote <<AEA>> constituído por cinco Atividades de Ensino e Aprendizagem e o pacote <<TA>> formado por

sete Tarefas de Avaliação.

Figura 20 – Módulos de Alinhamento Construtivo



Fonte: Elaboração do autor.

Vale ressaltar que o relacionamento bidirecional entre os módulos representa tanto a dependência de um Resultado Pretendido de Aprendizagem sobre o rol atual de Atividades de Ensino e Aprendizagem e Tarefas de Avaliação, como também a adequação dos dois últimos em prol do primeiro, visto que no AC, as AEAs e TAs são planejadas em função do RPA para que este possa ser atingido.

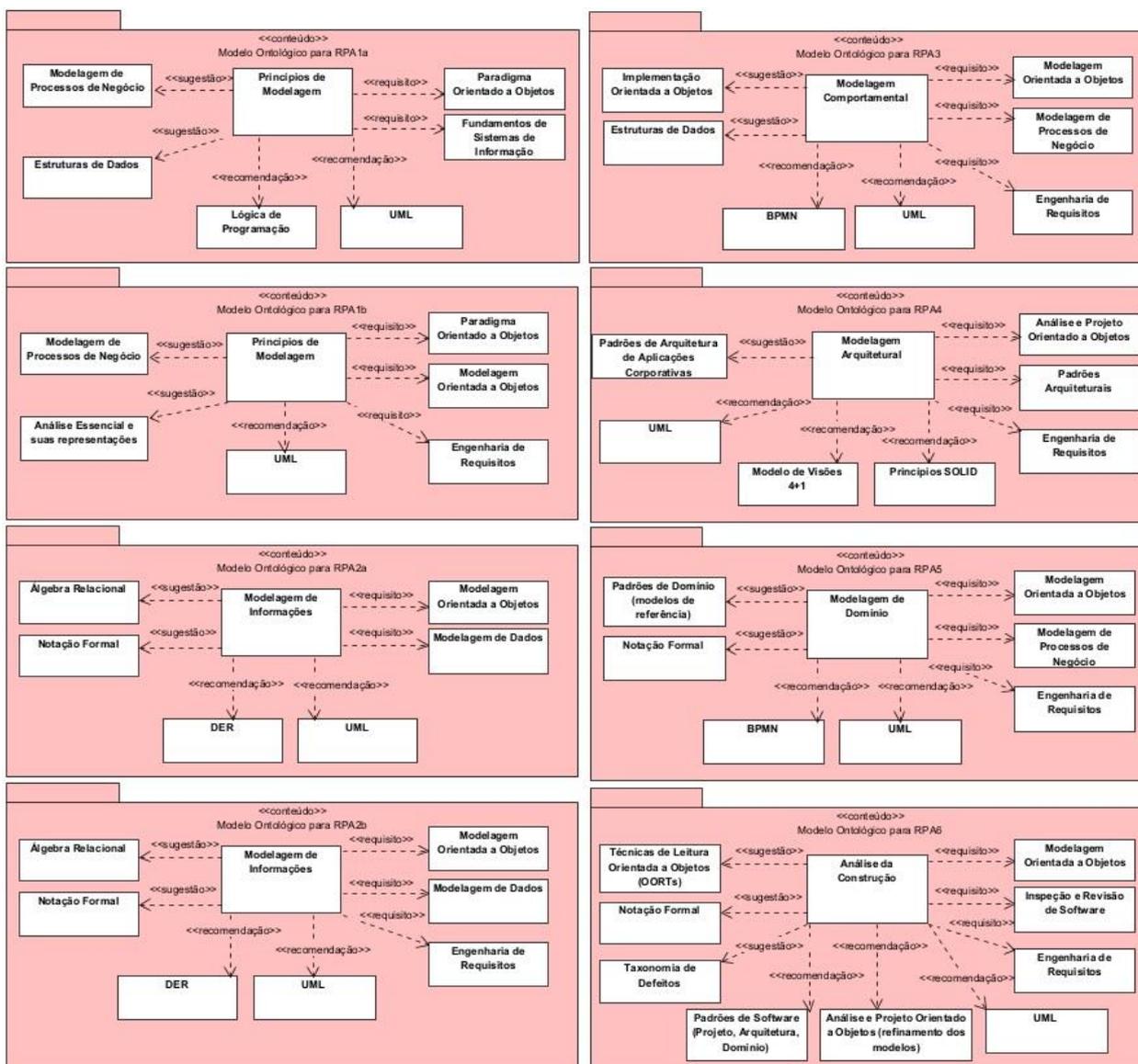
5.2.3. Concepção do Módulo de Modelos Ontológicos

No que tange aos três planos organizacionais da técnica OC2-RD2, o Modelo Ontológico¹⁵, em outros termos, o Plano de Conteúdos é o recurso pedagógico de destaque para este estudo.

Os resultados da pesquisa, sobretudo os provindos da pergunta E1 do Roteiro de Entrevista facultou a concepção do módulo referente a uma rede de conteúdos pertinente para alcançar cada RPA. Este módulo, representado por um conjunto de pacotes de Modelo Ontológico (Plano de Conteúdos) e estereotipados como <<conteúdo>>, é mostrado na Figura 21. Se ressalta aqui que as dependências entre itens de conhecimento vão depender do tópico de MAS a ser aprendido em cada RPA.

¹⁵ A técnica OC2-RD2 tomou como base o conceito de ontologia. Disponível em: [https://en.wikipedia.org/wiki/Ontology_\(computer_science\)](https://en.wikipedia.org/wiki/Ontology_(computer_science)).

Figura 21 – Módulo de Modelos Ontológicos para os oito RPAs da pesquisa



Fonte: Elaboração do autor.

Vale frisar ainda que apesar destas relações terem sido definidas com base no julgamento do pesquisador a partir da análise qualitativa dos resultados providos da pesquisa, as dependências entre itens de conhecimento devem preservar, quando houver, aquela da ontologia da área-alvo (domínio) de conhecimento, conforme o referencial teórico sobre o tema a ser tratado.

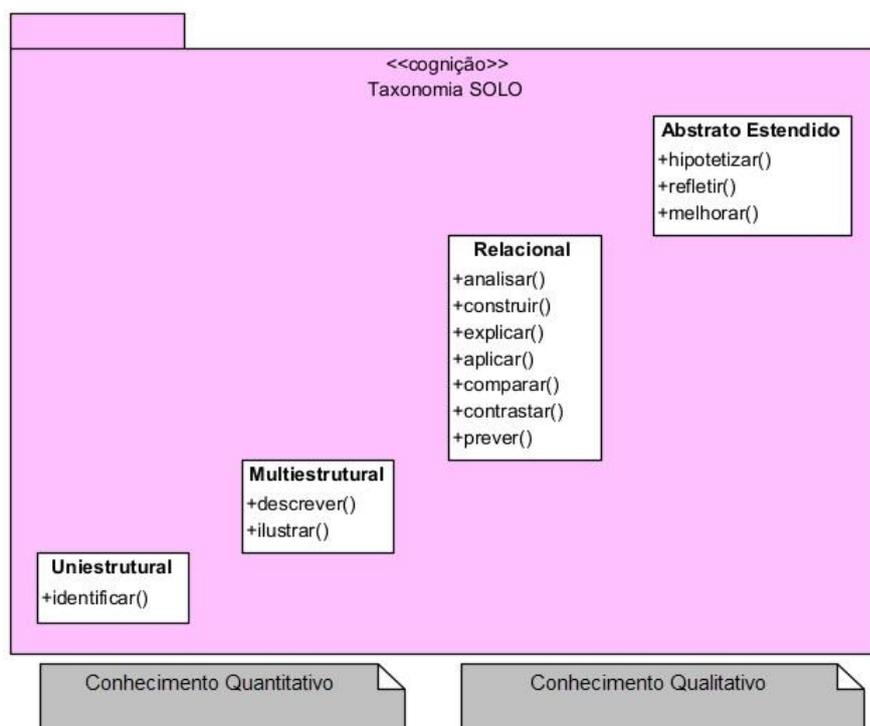
Desde já, vale comunicar que cada um desses oito planos de conteúdos vai ser melhor explorado mais adiante.

5.2.4. Concepção do Módulo de Taxonomia SOLO

Cabe lembrar que a correlação entre o Alinhamento Construtivo e a Técnica OC2-RD2, já apresentada na Figura 19, mostrou que ambas as abordagens (estruturas) pedagógicas são norteadas cognitivamente pela taxonomia SOLO, a qual tem o intuito de orientar o professor, do ponto de vista de desenvolvimento cognitivo e progressão do aprendizado, no planejamento de cursos e aulas, direcionando as atividades docentes e discentes para atingir o RPA em pauta.

Os resultados da pesquisa, em especial os procedentes das perguntas da Seção C do Roteiro de Entrevista ensejou a concepção do módulo relativo aos níveis e verbos de aprendizagem, divididos como de praxe, nos respectivos níveis da taxonomia SOLO, conforme mostra a Figura 22, na qual se nota o presente módulo homônimo e representado pelo pacote estereotipado <<cognição>>, formado por quatro níveis da taxonomia em evidência, cada qual contendo seus verbos de aprendizagem para adquirir conhecimento quantitativo nos níveis Uniestrutural e Multiestrutural, como também para adquirir conhecimento qualitativo nos níveis Relacional e Abstrato Estendido.

Figura 22 – Módulo Taxonomia SOLO



Fonte: Elaboração do autor.

É importante salientar que o nível Pré-estrutural não foi considerado

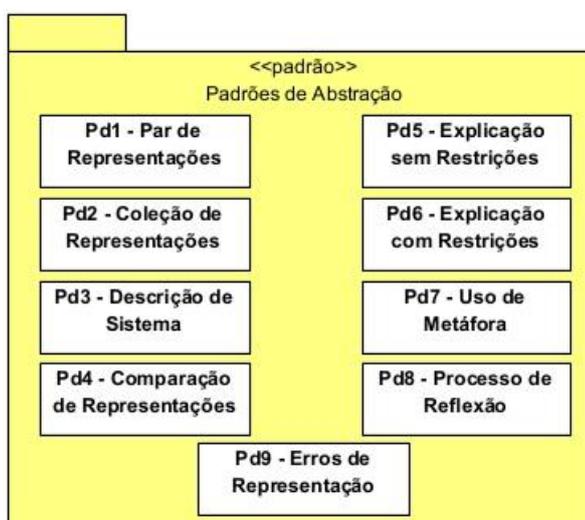
explicitamente no modelo proposto, uma vez que esse nível significa ausência de informações e, por consequência, nenhum verbo cognitivo de aprendizagem é reconhecido para tal nível.

Convém também frisar que a técnica OC2-RD2 pode ser orientada, em termos cognitivos, por outras taxonomias, como é o caso da Taxonomia de Bloom (KRATHWOHL, 2002), porém este estudo se concentrou na Taxonomia SOLO, uma vez que é a taxonomia de suporte ao Alinhamento Construtivo (BIGGS, 2003; BIGGS; TANG, 2011; RIZZO; POLETTI, 2019, 2021).

5.2.5. Concepção do Módulo de Padrões de Abstração

Os resultados da pesquisa, particularmente os derivados das perguntas da Seção D do Roteiro de Entrevista proporcionou a concepção do módulo concernente aos padrões de abstração, conforme exhibe a Figura 23, na qual se percebe o módulo Padrões de Abstração, representado pelo pacote estereotipado <<padrão>>, constituído pelos oito padrões de abstração identificados nos trabalhos de Hazzan e Kramer (2006; 2016) e avaliados em termos de adequação por este estudo. Ademais, os quatro erros comuns de abstração de Halladay e Wiebel (1993) investigados neste trabalho resultaram em um novo padrão de abstração. Cada um dos padrões foi enumerado de 1 a 9 (Pd1 a Pd9), assim como nomeado conforme seu propósito de aplicação.

Figura 23 – Módulo de padrões de abstração



Fonte: Elaboração do autor.

Convém reforçar aqui a significância do nono padrão de abstração (Pd9), visto

que o erro faz parte do processo de construção de conhecimento, havendo assim consonância à teoria de aprendizagem neopiagetiana e ao paradigma construtivista, conforme visto em Biggs e Collis (1982), Becker (2012), Biggs e Tang (2011), como também está consoante ao aprendizado por meio de erros de modelagem, observado no trabalho de Chourio *et al.* (2019).

5.2.6. Criação dos Planos Ontológicos

Os conceitos e módulos concebidos via resultados da pesquisa foram analisados e incorporados para se criar os planos temáticos e reutilizáveis pertencentes ao modelo proposto, os quais foram chamados aqui de planos ontológicos. A junção destes planos produziu uma Ontologia de Domínio de Modelagem e Análise de Software, a qual integra conteúdo e cognição de temas desse domínio de conhecimento, buscando abarcar a dialética entre forma e conteúdo no ensino de engenharia de software.

Conforme visto na Seção 5.2.3, o termo ontologia é adotado pela técnica de narrativas OC2-RD2 ao tratar da construção do Modelo Ontológico, em outras palavras, na elaboração do Plano de Conteúdos. Para reforçar o conceito de ontologia, este trabalho se baseia na definição de Guarino (2009).

Uma ontologia é uma teoria lógica responsável pelo significado pretendido de um vocabulário formal, ou seja, seu compromisso ontológico com uma conceitualização particular do mundo. Os modelos pretendidos de uma linguagem lógica usando tal vocabulário são limitados por seu compromisso ontológico. Uma ontologia reflete indiretamente esse compromisso (e a conceitualização subjacente) ao aproximar esses modelos pretendidos. (GUARINO, 2009, p. 7).

No que diz respeito aos tipos de ontologias e considerando seu grau de dependência em relação a uma determinada tarefa ou ponto de vista, Guarino (2009) apresenta as seguintes categorias:

- a) Ontologias de alto nível descrevem conceitos muito gerais como espaço, tempo, matéria, objeto, evento, ação etc., que são independentes de um problema ou domínio particular: parece, portanto, razoável, pelo menos em teoria, ter unificado ontologias para grandes comunidades de usuários;

- b) Ontologias de domínio e ontologias de tarefas descrevem, respectivamente, o vocabulário relacionado a um domínio genérico (como remédios ou automóveis) ou uma tarefa ou atividade genérica (como diagnosticar ou vender), especializando os termos introduzidos na ontologia de nível superior;
- c) As ontologias de aplicação descrevem conceitos que dependem tanto de um domínio particular quanto de uma tarefa, que geralmente são especializações de ambas as ontologias relacionadas. Esses conceitos geralmente correspondem a papéis desempenhados por entidades de domínio durante a execução de uma determinada atividade, como unidade substituível ou componente sobressalente.

Dando continuidade, a denominação de cada plano foi constituída pelas iniciais do nome da área de conhecimento em foco (Modelagem e Análise de Software), de um tema (tópico) desta área e do nível cognitivo SOLO, conforme clarifica o Quadro 21.

Quadro 21 – Denominações dos planos ontológicos

Área de Conhecimento: Modelagem e Análise de Software (MAS)			
RPA	Tema	Nível SOLO	Plano Ontológico
RPA1a	Princípios de Modelagem (PM)	Uniestrutural (U)	MAS_PM_U
RPA1b	Princípios de Modelagem (PM)	Multiestrutural (M)	MAS_PM_M
RPA2a	Modelagem de Informações (MI)	Multiestrutural (M)	MAS_MI_M
RPA2b	Modelagem de Informações (MI)	Relacional (R)	MAS_MI_R
RPA3	Modelagem Comportamental (MC)	Relacional (R)	MAS_MI_R
RPA4	Modelagem Arquitetural (MA)	Relacional (R)	MAS_MA_R
RPA5	Modelagem de Domínio (MD)	Relacional (R)	MAS_MD_R
RPA6	Análise da Construção (AC)	Abstrato Estendido (AE)	MAS_AC_AE

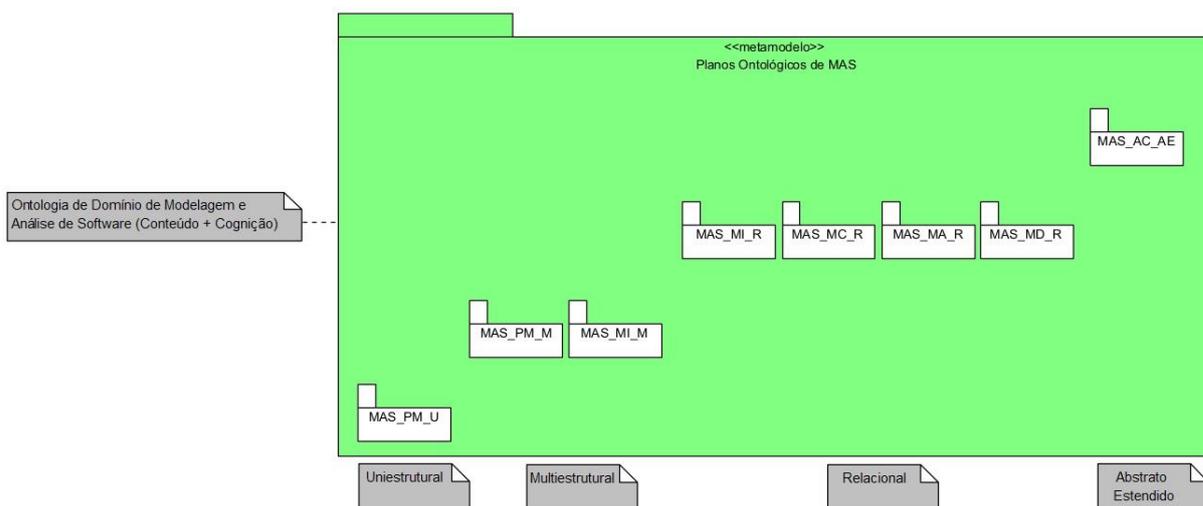
Fonte: Elaboração do autor.

O conjunto desses planos ontológicos implicaram no surgimento do modelo pedagógico denominado Modelo de Planos Ontológicos de Modelagem e Análise de Software (MPO_MAS). Apesar do termo modelo ter sido adotado no decurso desta tese, a estrutura da proposição em estudo se caracteriza também como metamodelo, podendo ampliar essa denominação, até mesmo, para metametamodelo.

A Figura 24 expõe os oito planos ontológicos pertencentes ao MPO_MAS,

cada qual representado por um pacote e separado por nível de compreensão SOLO, onde se observa o agrupamento de todos os planos em uma estrutura mais ampla, a qual também é representada por um pacote, porém estereotipado como <<metamodelo>>.

Figura 24 – Planos ontológicos de modelagem e análise de software por nível SOLO



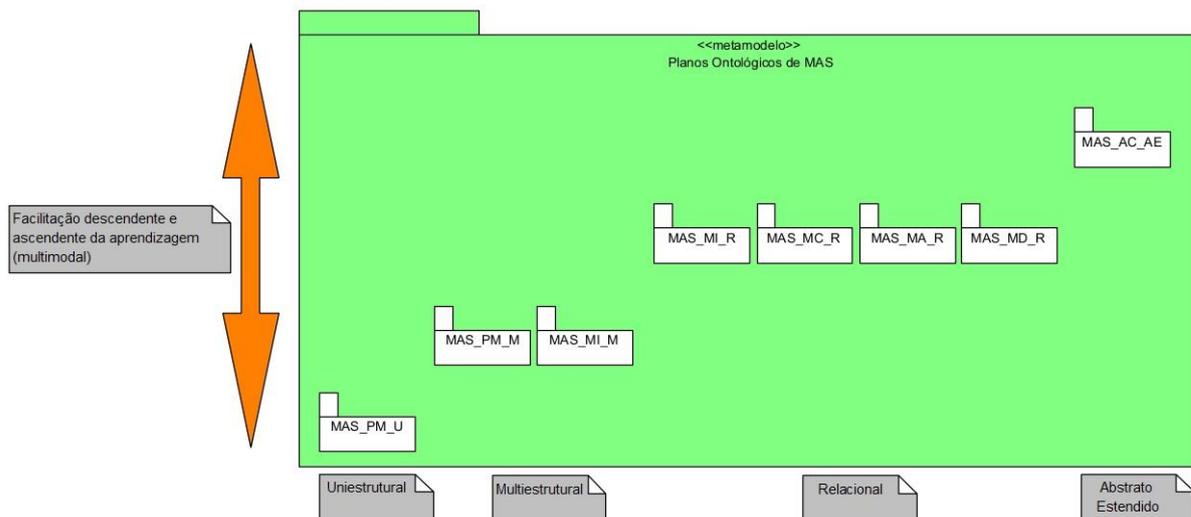
Fonte: Elaboração do autor.

Muito importante ter em mente que fatores como motivação, experiência e conhecimentos prévios podem interferir no desempenho de uma tarefa e, conseqüentemente, no desenvolvimento cognitivo e na progressão do aprendizado. Nesse sentido, é prudente considerar o reuso de planos ontológicos de níveis de compreensão mais baixos, buscando, desta forma, viabilizar o processo de construção do conhecimento assim como diminuir a ponte (distância) cognitiva entre determinados verbos (níveis) de aprendizagem.

A Figura 25 ilustra os oito planos ontológicos pertencentes ao MPO_MAS e separados por nível cognitivo SOLO novamente, mas desta vez evidenciando dois tipos qualitativos de desempenho cognitivo, em outros termos, a facilitação descendente da aprendizagem de ordem inferior e a facilitação ascendente da aprendizagem de ordem superior, sendo ambas consideradas multimodais; o termo multimodal significa que o aprendizado possui mais de um modo de funcionamento. Convém lembrar que esses modos fazem alusão aos principais estágios piagetianos de desenvolvimento cognitivo: sensório-motor, intuitivo (ou pré-operacional), concreto e formal, porém, conforme visto em conforme visto em Biggs e Collis (1982), Biggs (1992) e Mol e Matos (2019), cada modo tem suas características típicas de

funcionamento cognitivo.

Figura 25 – Planos ontológicos por nível SOLO em uma abordagem de facilitação descendente e ascendente de aprendizagem



Fonte: Elaboração do autor.

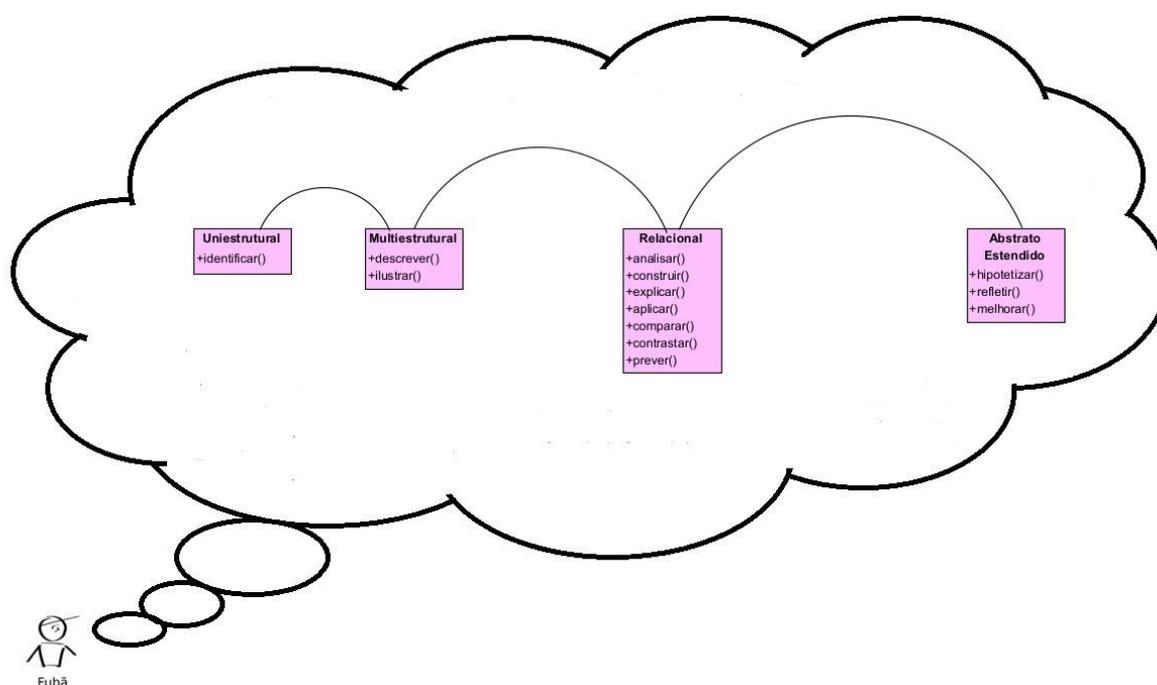
Ainda sobre os fatores supracitados que podem influenciar o desenvolvimento cognitivo e a progressão do aprendizado, é salutar entender que existem distâncias cognitivas maiores ou menores, a depender dos verbos de aprendizagem e do desempenho desejado para a tarefa em questão. Por exemplo, a distância cognitiva entre verbos de aprendizagem de nível Uniestructural e Multiestructural é menor e exige menos esforço do que a distância entre verbos de aprendizagem de nível Multiestructural e Relacional, e menor ainda quando se compara ao distanciamento entre um verbo de nível Relacional e outro de nível Abstrato Estendido; esse espaço e esforço aumentam ainda mais entre os modos de funcionamento.

Biggs e Collis (1982) salientam que o caminho de uma resposta de nível Relacional no modo concreto para uma resposta de nível Abstrato Estendido no modo formal, onde acontece um movimento para um novo modo de funcionamento é muito maior do que o caminho entre os outros níveis de funcionamento dentro do mesmo modo em foco. Isso significa que o espaço entre os modos de funcionamento (por exemplo, entre concreto e formal) torna-se maior e requer mais esforço para fechar essa lacuna, quando comparado entre os níveis Uniestructural, Multiestructural e Relacional no mesmo modo de funcionamento (por exemplo, intuitivo ou concreto).

Com o intuito de clarificar melhor a situação relatada, a Figura 26 busca

representar uma possível ponte (*bridge*¹⁶) cognitiva (na mente) de Fubã¹⁷ cujo personagem deseja atingir desempenho em uma dada tarefa, a ser manifestado via estrutura da resposta, de acordo com o verbo de aprendizagem pretendido. À vista disto, a reutilização de planos ontológicos de níveis de compreensão mais baixos deve ser levada em consideração, a fim de facilitar a aprendizagem tanto de baixo para cima quanto de cima para baixo, assim como reduzir a ponte cognitiva existente entre verbos de aprendizagem de níveis cognitivos SOLO e, principalmente, entre modos de funcionamento. Em suma, deve haver coerência por parte do responsável pela disciplina no planejamento de ensino que se deseja níveis de compreensão e modos de funcionamento mais complexos.

Figura 26 – Ponte cognitiva na mente de Fubã



Fonte: Elaboração do autor.

Cabe ressaltar ainda que, embora os fatores mencionados sejam reconhecidos no Alinhamento Construtivo de Biggs e Tang (2011), especialmente devido à influência da visão neopiagetiana realçada em Biggs e Collis (1982) e Biggs (1992), tais aspectos também são notados na OC2-RD2, conforme trabalhos de Vega (2018) e Buttignon (2020), presumindo uma evidência da presença da abordagem

¹⁶ Termo extraído de Biggs e Collis (1982).

¹⁷ Um dos personagens da técnica de narrativas OC2-RD2 que representa a faceta da curiosidade investigativa.

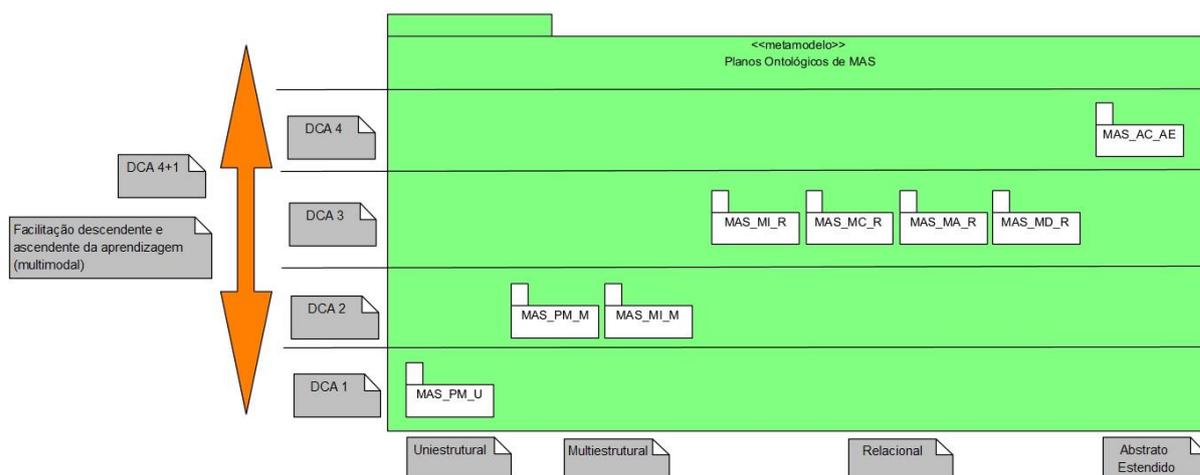
neopiagetiana em foco nessa técnica de narrativas.

5.2.7. Mapeamento dos Planos Ontológicos para Desempenhos Cognitivos de Abstração

Ainda considerando os resultados da pesquisa, sobretudo os oriundos das perguntas da Seção B do Roteiro de Entrevista, se vislumbrou também uma perspectiva a respeito do desenvolvimento da capacidade cognitiva da abstração.

A Figura 27 mostra novamente a alocação de cada plano ontológico, horizontalmente, no respectivo nível SOLO, como também agora o mapeamento de cada plano, verticalmente, em relação ao Desempenho Cognitivo de Abstração (DCA). Metaforicamente, se pode pressupor que cada plano está situado em uma determinada célula, melhor dizendo, na intersecção entre o nível SOLO e o DCA.

Figura 27 – Planos ontológicos de modelagem e análise de software por nível SOLO e capacidade cognitiva de abstração



Fonte: Elaboração do autor.

Para se entender o significado de cada desempenho cognitivo de abstração, o Quadro 22 descreve cada um deles. Vale frisar que tais descrições foram compiladas a partir da visão neopiagetiana de desenvolvimento cognitivo, mencionada em Lister (2011), Gluga *et al.* (2012) e Falkner, Vivian e Falkner (2013) e, principalmente, em Biggs e Collis (1982). Já o conceito de facilitação ascendente e descendente de aprendizagem foi baseado no trabalho de Biggs (1992). Por fim, é cabível frisar que este estudo adaptou os verbos “construir e compreender” com base na expressão “fazer e compreender” de Piaget (1978), originalmente redigidos como “*réussir et comprendre*” (1974), tentando fazer uma correspondência alusiva aos dois tipos de

conhecimento tratados neste trabalho, ou seja, o verbo “fazer” como conhecimento funcional e o verbo “compreender” como conhecimento declarativo.

Quadro 22 – Descrição dos desempenhos cognitivos de abstração

DCA	Descrição
DCA 1	Construir e compreender abstrações dentro do mesmo domínio do problema, se concentrando em apenas uma propriedade abstrata em um determinado momento.
DCA 2	Construir e compreender abstrações dentro do mesmo domínio do problema, se concentrando em poucas propriedades abstratas e isoladas em um determinado momento.
DCA 3	Construir e compreender abstrações dentro do mesmo domínio do problema, se concentrando em várias propriedades abstratas em um determinado momento, mas restritas a situações familiares e reais.
DCA 4	Construir e compreender abstrações que ultrapassem o domínio do problema, se concentrando em várias propriedades abstratas em um determinado momento sobre situações ainda não vivenciadas diretamente.
DCA 4+1	As quatro habilidades de abstração, facilitadas tanto de baixo para cima quanto de cima para baixo.

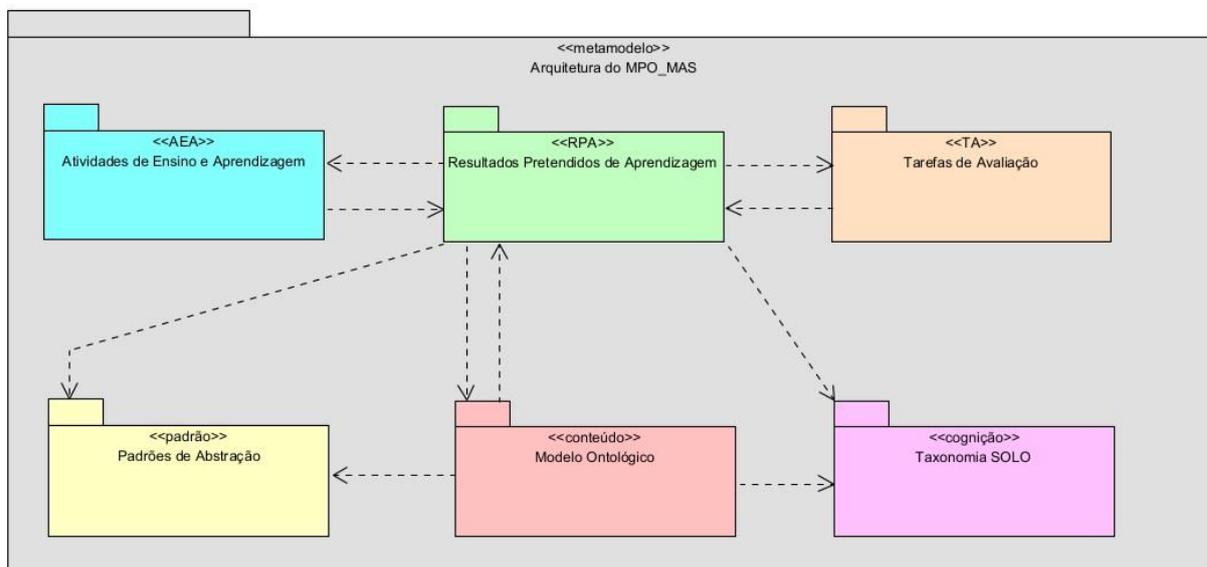
Fonte: Elaboração do autor.

Cabe destacar mais uma vez a significância da facilitação tanto de cima para baixo quanto de baixo para cima na reutilização dos planos ontológicos disponíveis no MPO_MAS, visando tanto a aprendizagem de conteúdos de Modelagem e Análise de Software, como também a elevação da capacidade cognitiva de abstração, reforçando que essa habilidade está presente entre os níveis cognitivos SOLO, mas sobretudo, entre os modos de funcionamento.

5.2.8. Design Arquitetural do Modelo

Uma vez que os planos ontológicos foram criados, se realizou o *design* arquitetural do modelo pedagógico (ontológico), acarretando a integração entre os seis módulos concebidos e, por conseguinte, a arquitetura pedagógica (ontológica) do MPO_MAS, conforme mostra a Figura 28, na qual cada módulo é representado por um pacote estereotipado.

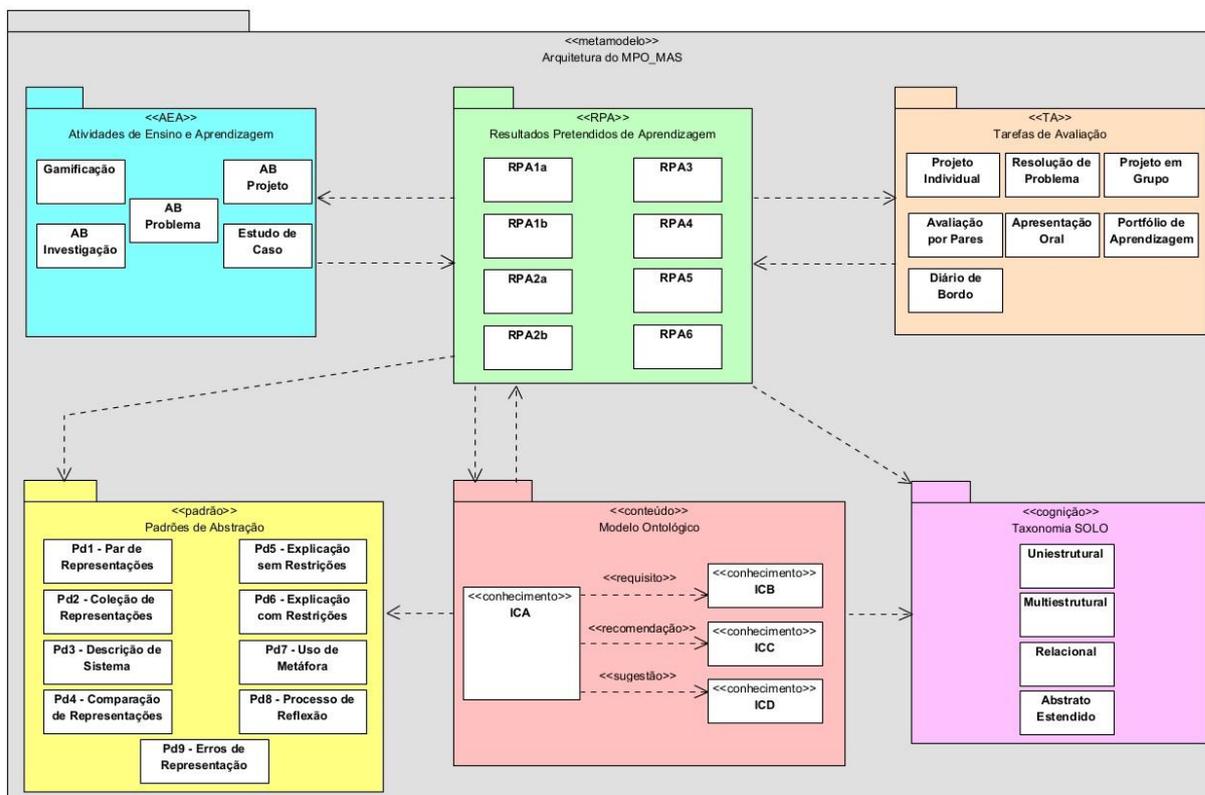
7Figura 28 – Módulos da arquitetura do MPO_MAS



Fonte: Elaboração do autor.

O metamodelo MPO_MAS contém atualmente oito planos ontológicos, temáticos e reutilizáveis, cada qual direcionado para seu RPA. Adentrando neste metamodelo, é possível conhecer em maiores detalhes os conceitos (conteúdo e cognição), representados como classes, de cada um dos módulos (partes integrantes) da arquitetura predita, conforme mostra a Figura 29. Vale ressaltar que, devido à legibilidade do diagrama, os pacotes <<RPA>>, <<conteúdo>> e <<cognição>> exibem apenas os conceitos gerais inclusos, entretanto, vale lembrar de que os conceitos específicos foram representados preliminarmente durante a concepção de cada módulo.

Figura 29 – Módulos e Conceitos da Arquitetura do MPO_MAS

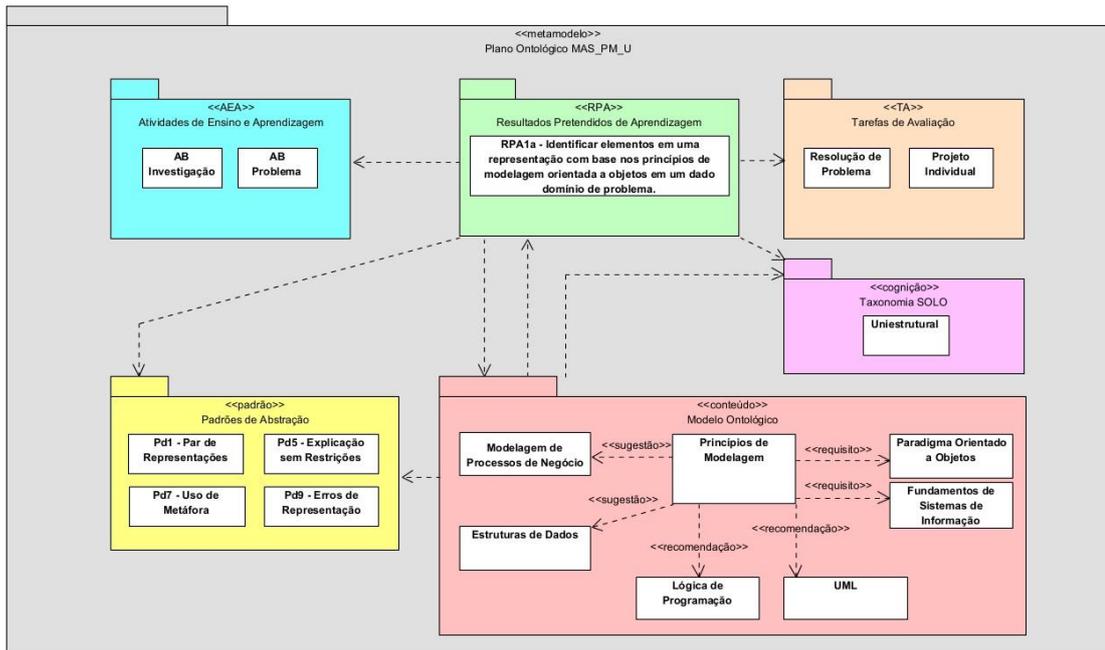


Fonte: Elaboração do autor.

A construção de todos os planos ontológicos do presente metamodelo seguiu as diretrizes arquiteturais explanadas, recordando que cada plano foi projetado de acordo com o RPA em questão. Na sequência, os oito planos são retratados, precisamente um a um.

O plano ontológico MAS_PM_U é apresentado na Figura 30. Os conceitos envolvidos com o RPA1a são representados como classes em seus respectivos pacotes.

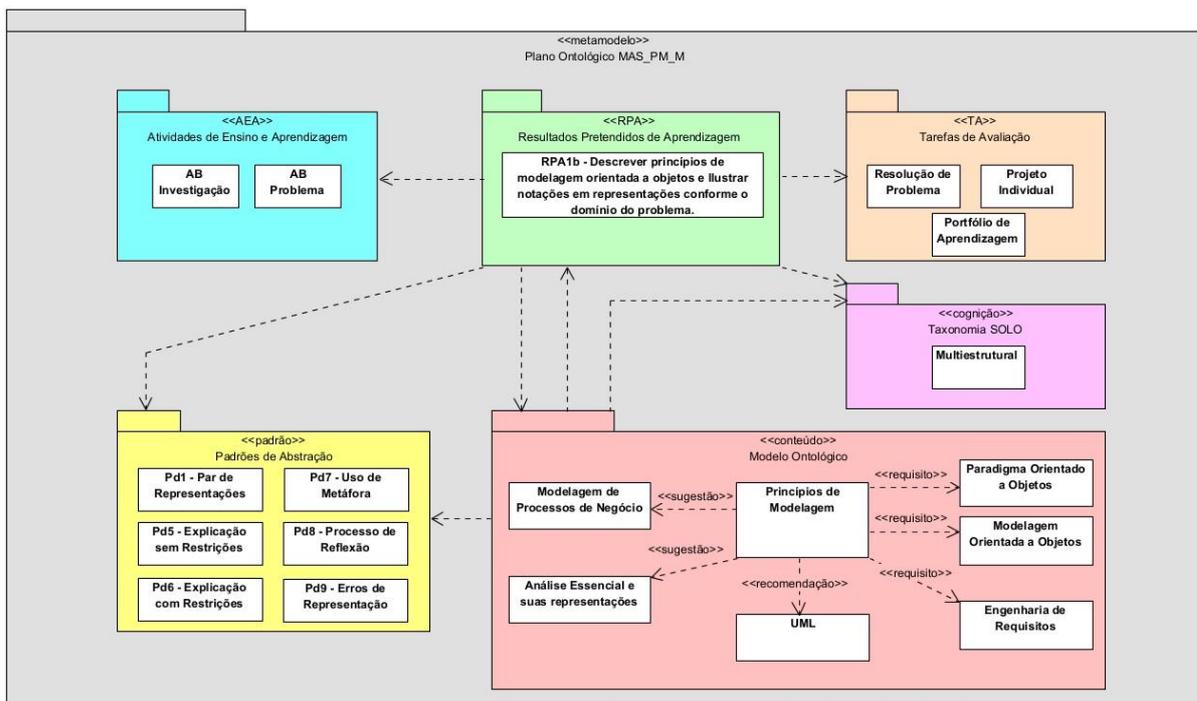
Figura 30 – Plano ontológico de MAS_PM_U



Fonte: Elaboração do autor.

O plano ontológico MAS_PM_M é apresentado na Figura 31. Os conceitos relacionados com o RPA1b são representados como classes em seus devidos pacotes.

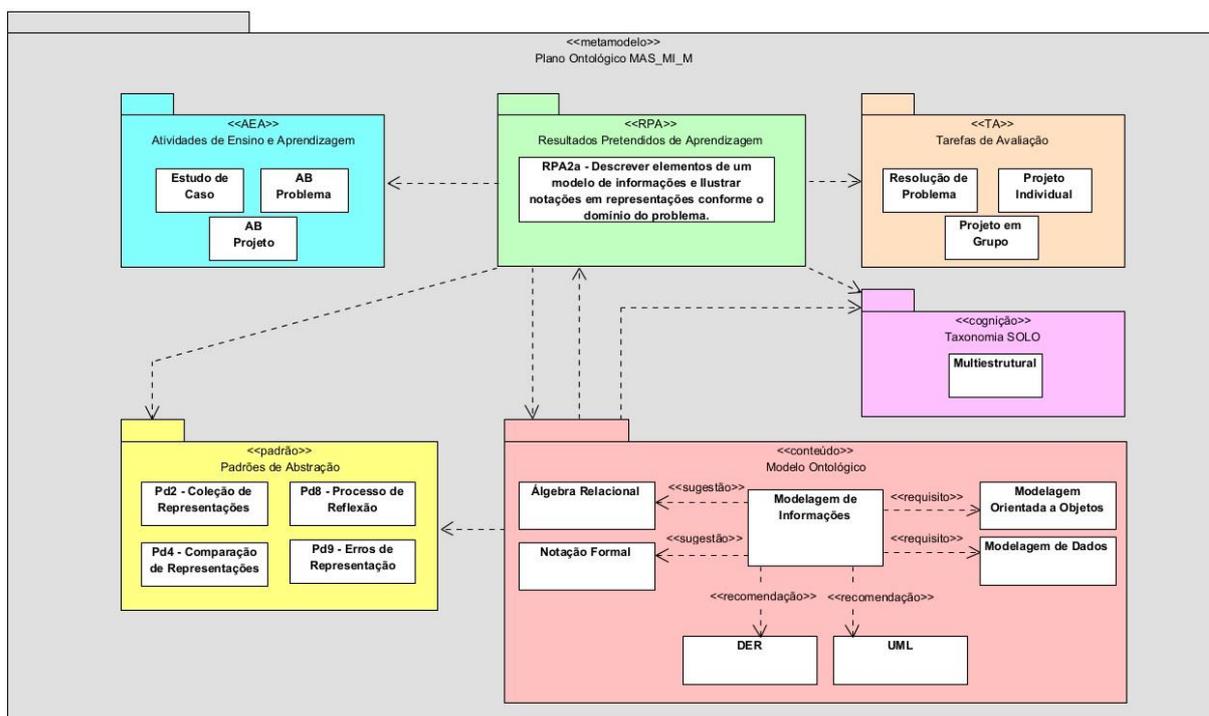
Figura 31 – Plano ontológico de MAS_PM_M



Fonte: Elaboração do autor.

O plano ontológico MAS_MI_M é apresentado na Figura 32. Os conceitos envolvidos com o RPA2a são representados como classes em seus respectivos pacotes.

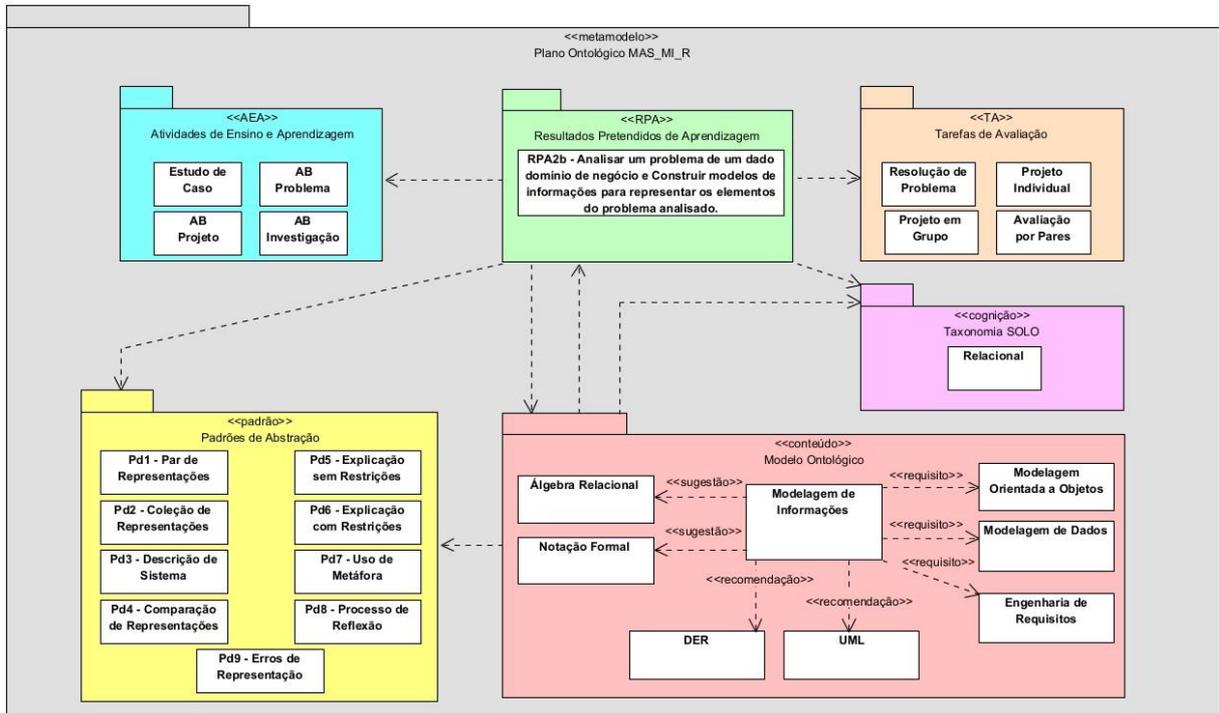
Figura 32 – Plano ontológico de MAS_MI_M



Fonte: Elaboração do autor.

O plano ontológico MAS_MI_R é apresentado na Figura 33. Os conceitos relacionados com o RPA2b são representados como classes em seus devidos pacotes.

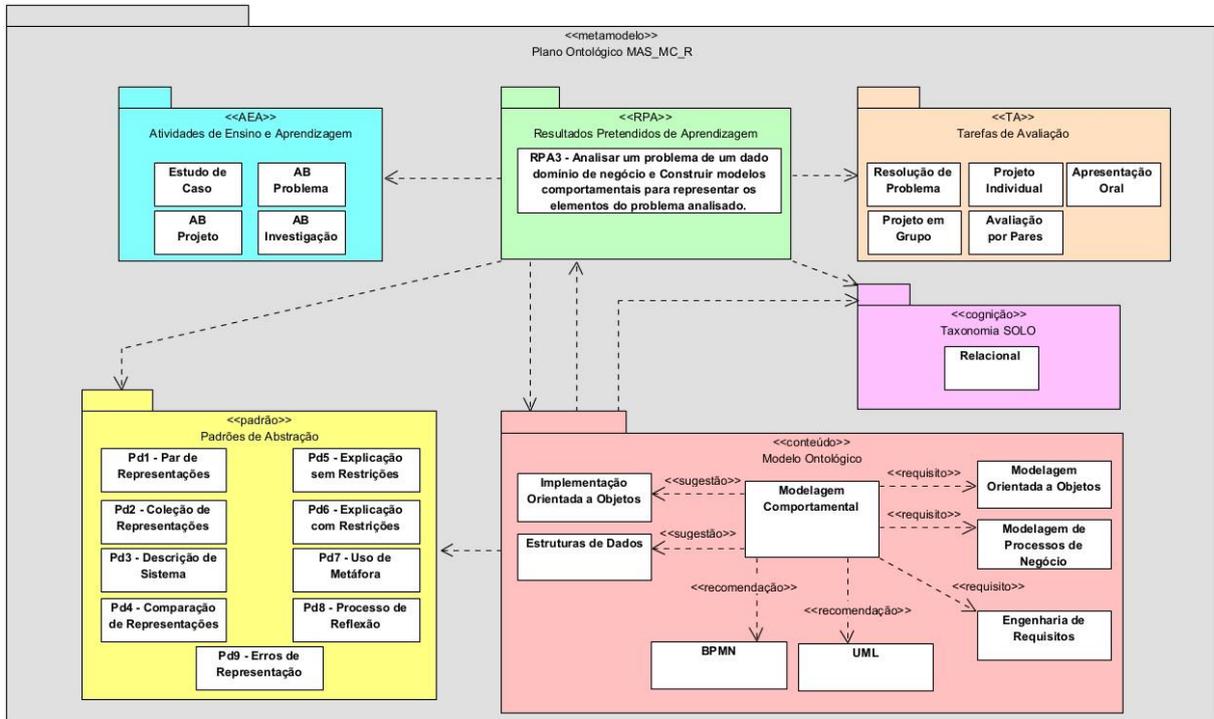
Figura 33 – Plano ontológico de MAS_MI_R



Fonte: Elaboração do autor.

O plano ontológico MAS_MC_R é apresentado na Figura 34. Os conceitos envolvidos com o RPA3 são representados como classes em seus respectivos pacotes.

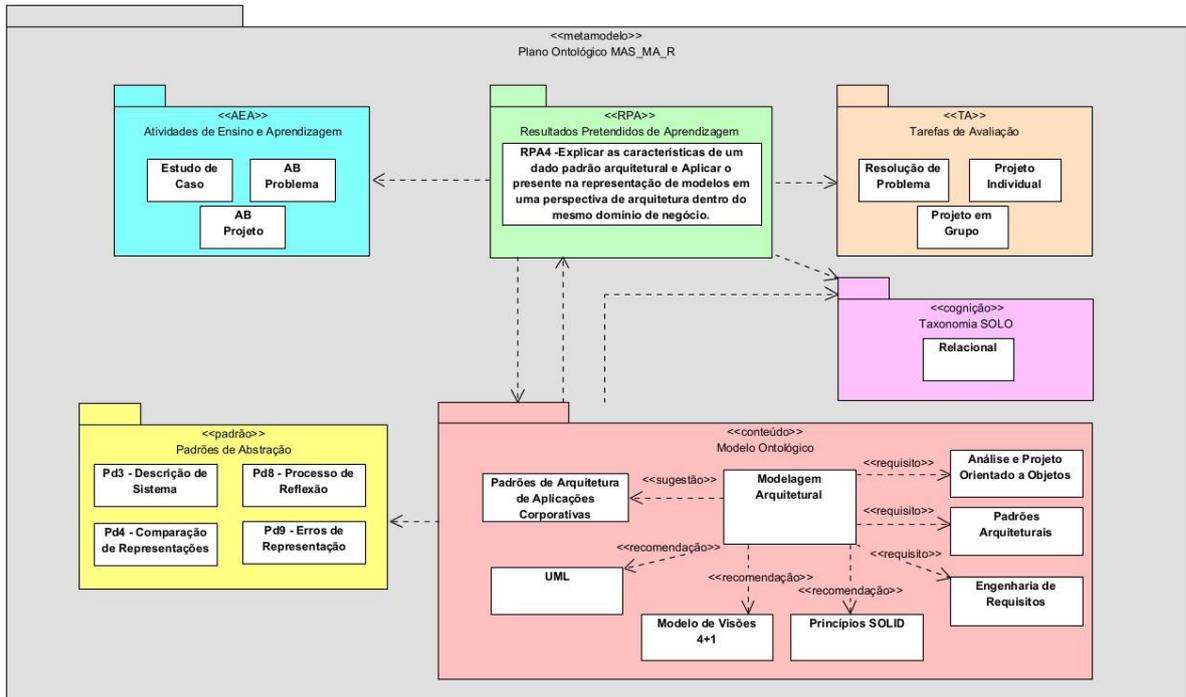
Figura 34 – Plano ontológico de MAS_MC_R



Fonte: Elaboração do autor.

O plano ontológico MAS_MA_R é apresentado na Figura 35. Os conceitos relacionados com o RPA4 são representados como classes em seus devidos pacotes.

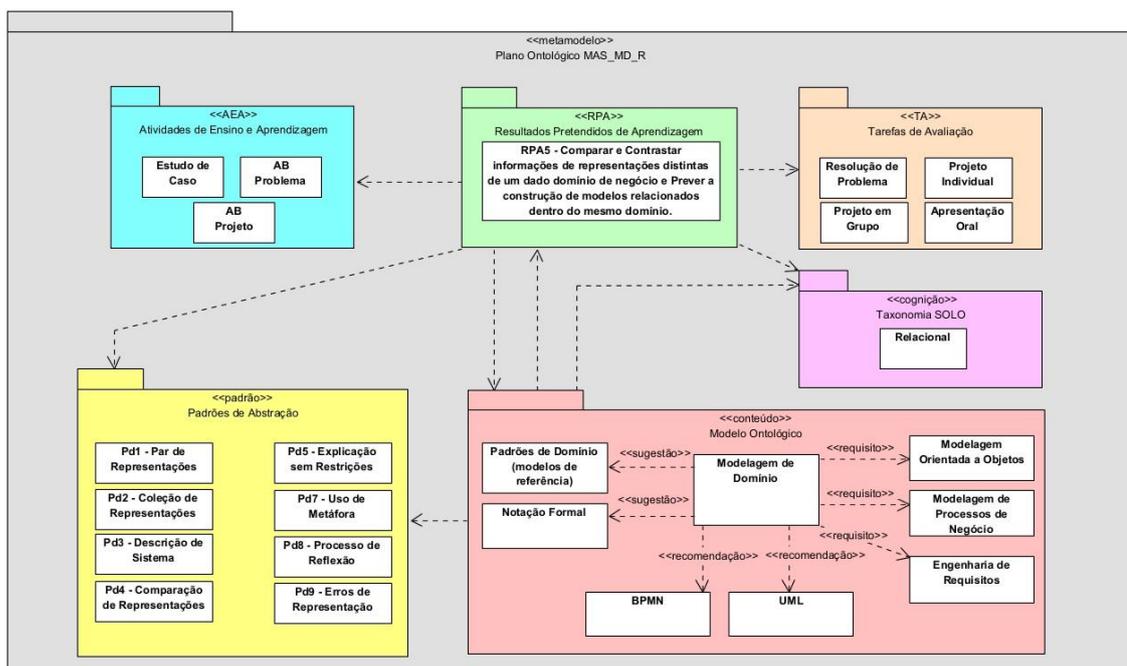
Figura 35 – Plano ontológico de MAS_MA_R



Fonte: Elaboração do autor.

O plano ontológico MAS_MD_R é apresentado na Figura 36. Os conceitos envolvidos com o RPA5 são representados como classes em seus respectivos pacotes.

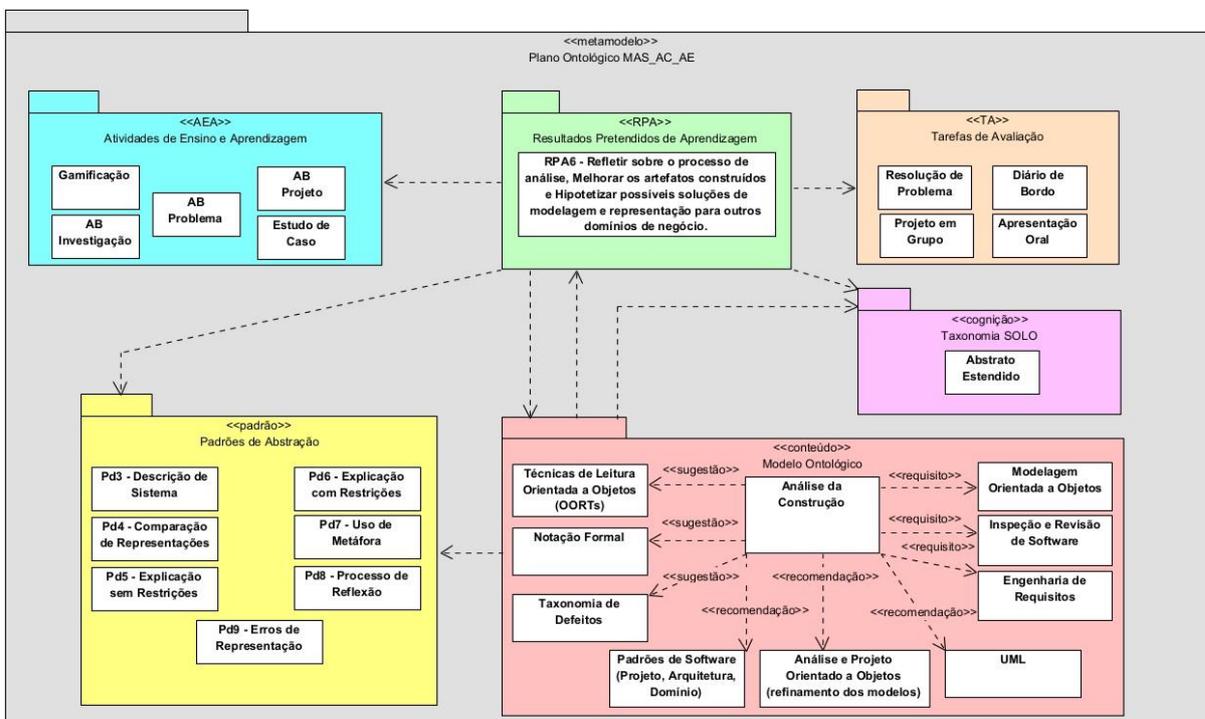
Figura 36 – Plano ontológico de MAS_MD_R



Fonte: Elaboração do autor.

O plano ontológico MAS_AC_AE é apresentado na Figura 37. Os conceitos relacionados com o RPA6 são representados como classes em seus devidos pacotes.

Figura 37 – Plano ontológico de MAS_AC_AE



Fonte: Elaboração do autor.

Após a apresentação de cada um dos oito planos ontológicos com seus módulos e conceitos correspondentes, a próxima seção busca validar, ao menos de modo preliminar o modelo proposto.

5.3 Validação do Modelo

A fim de validar a aplicabilidade do modelo proposto, este estudo, selecionou ao todo quatro planos de disciplina de dois cursos superiores de tecnologia distintos como situações-exemplo de aplicação do MPO_MAS. Cabe informar previamente que esses planos foram extraídos dos respectivos Projetos Pedagógicos de Curso (PPC). Antes da apresentação dos exemplos, o procedimento e cenários de aplicação do modelo são explanados.

5.3.1. Procedimento de Aplicação do Modelo

Como já foi mencionado, embora o termo modelo tenha sido adotado no

decorrer desta tese, a arquitetura pedagógica resultante caracteriza o modelo em estudo como um metamodelo, até mesmo como um metametamodelo, conforme será visto logo adiante.

Antes de aplicar o MPO_MAS nos planos de disciplina selecionados, é importante lembrar que o metamodelo abarca um conjunto de planos de conteúdo e cognição, temáticos e reutilizáveis, de uma determinada ontologia de domínio de conhecimento, melhor dizendo, de modelagem e análise de software. Em suma, este metamodelo oferece atualmente oito planos ontológicos dessa área de conhecimento cujos temas são frequentemente contemplados em componentes curriculares de engenharia de software e/ou disciplinas afins.

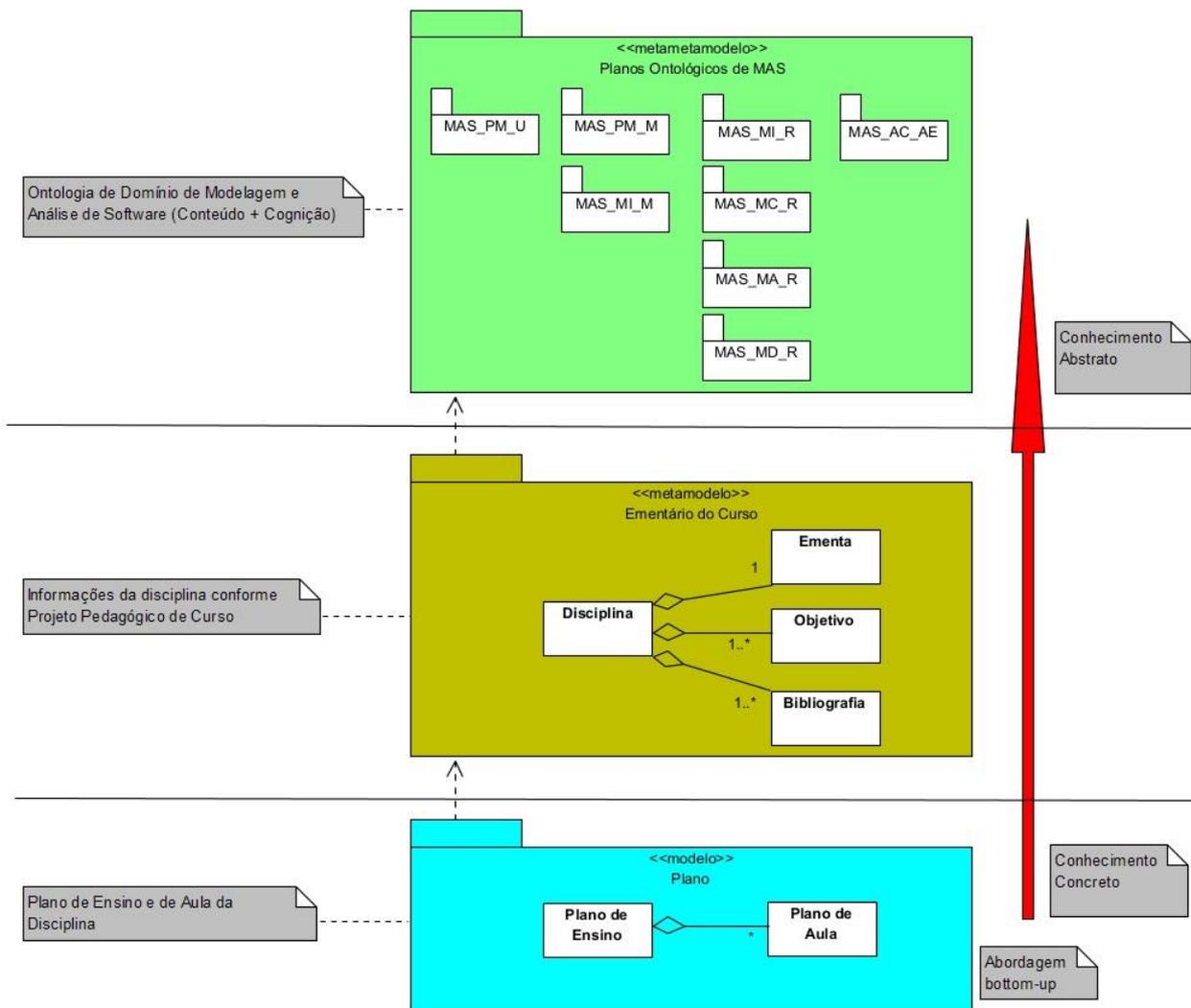
Para aplicar o MPO_MAS na elaboração de planos de ensino e de aula, é preciso seguir uma abordagem *bottom-up* de uso, constituída essencialmente por três etapas, sendo:

- 1- Selecionar uma determinada disciplina que se tenha a incumbência de elaborar o plano de ensino ou de aula e tomar ciência dos elementos de informação que devem ser apresentados na estrutura desse plano;
- 2- Constatar os objetivos de aprendizagem da disciplina selecionada com base nas informações providas do respectivo Projeto Pedagógico de Curso (PPC). Não obstante, outras informações desse componente curricular, como ementa e bibliografias, também podem servir de suporte informacional;
- 3- Consultar o catálogo de planos ontológicos e identificar a possibilidade, conforme associação (mapeamento) do objetivo de aprendizagem ao resultado pretendido de aprendizagem, de um ou mais planos serem reutilizados para atender um ou mais objetivos de aprendizagem da disciplina. Sendo plausível, o plano ontológico deve ser adequado ao contexto do plano de ensino ou de aula da disciplina de que se trata.

A Figura 38 ilustra as três etapas essenciais para aplicação do MPO_MAS para elaboração de planos de ensino e/ou de aula, na qual se percebe a abordagem *bottom-up* de aplicação predita, a qual deve partir do conhecimento (conteúdo e cognição) concreto desejado no referido plano e seguir em direção ao conhecimento abstrato disponibilizado no plano ontológico. Cabe observar também, que diante do processo de aplicação apresentado, o MPO_MAS passou a ser considerado como

metametamodelo, ainda mais abstrato, dado que o plano de disciplina contido no PPC foi interpretado como metamodelo e o plano de ensino, por sua vez, como modelo.

Figura 38 – Etapas para aplicação do MPO_MAS



Fonte: Elaboração do autor.

Ainda sobre a aplicação do MPO_MAS, deve ficar claro que a intenção do modelo não é a de substituir o PPC, mas sim de ser utilizado como modelo pedagógico (ontológico) para apoiar a elaboração de planos de ensino e de aula, por intermédio do projeto pedagógico em questão, sabendo que este último é tratado como documento público que fundamenta e sistematiza a organização do conhecimento de um dado curso, apresentando inclusive, os fundamentos conceituais, metodológicos e avaliativos de cada disciplina pertencente a sua matriz curricular.

Diante do que foi explanado, é relevante deixar claro que o MPO_MAS não possui grandes restrições quanto a sua aplicabilidade, contudo se faz necessário que

o docente responsável tenha conhecimento da área de conhecimento da disciplina, como também dos objetivos do curso previstos no projeto pedagógico, avistando, desta forma, uma aplicação mais efetiva do modelo. Por fim, se espera que a proposição deste estudo se torne um modelo pedagógico reutilizável para professores de cursos de computação distintos que precisem abordar temas de modelagem e análise de software em disciplinas de engenharia de software e/ou componentes curriculares análogos.

5.3.2. Cenários e Exemplos de Aplicação do MPO_MAS

Com a finalidade de melhorar a compreensão da aplicação do MPO_MAS, esta subseção fornece noções relevantes no que concerne os cenários (situações-exemplo) em que o modelo proposto foi aplicado.

Como já mencionado, as informações dos quatro planos das disciplinas foram extraídas de dois PPCs cujos documentos públicos se encontram disponíveis na Web. A escolha desses componentes curriculares levou em consideração os seguintes critérios:

- a) Experiência do pesquisador nas disciplinas;
- b) Matriz curricular e contribuição da disciplina em relação aos objetivos do curso;
- c) Base de conhecimento que tais disciplinas fornecem na área de Engenharia de Software e;
- d) Clareza dos objetivos de aprendizagem estabelecidos nos respectivos planos analisados.

A aplicação do modelo enfatizou as etapas 2 e 3, melhor dizendo, no uso do PPC (metamodelo) e no reuso dos planos ontológicos (metametamodelo), respectivamente, conforme ilustrado preliminarmente na Figura 38. No entanto, cabe recordar que a etapa 1 se refere à elaboração do próprio plano de ensino ou de aula.

Conforme explicado anteriormente, fatores como motivação, experiência e conhecimentos prévios influenciam o desenvolvimento cognitivo e a progressão do aprendizado, portanto, a reutilização de planos ontológicos de níveis de compreensão mais baixos foram prudentemente considerados, visando facilitar o processo de construção de conhecimento tanto de baixo para cima quanto de cima pra baixo, assim como buscando reduzir a ponte cognitiva existente entre certos verbos (níveis) de

aprendizagem.

Os objetivos de aprendizagem foram transcritos na íntegra, consoante ao plano da disciplina. Para cada objetivo, os seguintes passos foram seguidos:

- a) O verbo de aprendizagem foi reconhecido;
- b) O tipo de conhecimento (declarativo ou funcional) foi identificado;
- c) O nível cognitivo SOLO foi classificado;
- d) Mediante as informações dos itens anteriores e levando em conta o tópico a ser aprendido, se consultou o catálogo da ontologia de domínio, usando o RPA como expressão de análise para se fazer o mapeamento;
- e) Após consulta e mapeamento, se fez a seleção de um ou mais planos ontológicos.

Vale salientar que, muito possivelmente, os objetivos originais dos planos das disciplinas não tenham sido redigidos como o emprego de verbos de aprendizagem da taxonomia SOLO. Posto isto, é notório que verbos de determinados objetivos precisaram ser interpretados/inferidos com ponderação, a fim de aproximá-los aos verbos da taxonomia cognitiva predita. Não obstante, a variedade de sinônimos para alguns verbos pode ter influenciado este processo de interpretação/inferência.

Finalmente, apesar de não ter havido a identificação direta do plano ontológico MAS_AC_AE, isso não significa que não se espera respostas de nível Abstrato Estendido para tarefas de níveis de compressão inferiores, porém se pressupõe que tais respostas vão estar relacionadas ao ciclo de aprendizagem dentro do mesmo modo de funcionamento, por exemplo, respostas de nível Uniestrutural, Multiestrutural, Relacional e Abstrato Estendido no modo Intuitivo ou Concreto, ao invés da resposta ser manifestada entre modos de funcionamento distintos, isto é, entre os modos Sensório-motor, Intuitivo, Concreto e Formal. Sobre a ausência do referido plano ontológico nas situações-exemplo usadas neste trabalho, pode ser que:

- a) O componente curricular não carece do conteúdo e nível de compreensão previsto para tal plano;
- b) Tenha faltado a presença de um ou mais verbos de maior especificidade na descrição do objetivo de aprendizagem ou;
- c) Tenha havido falha na interpretação/inferência do objetivo por parte do autor deste estudo.

Os quatro exemplos de aplicação do MPO_MAS, cada qual fazendo alusão a um determinado componente curricular e em conformidade com o plano da disciplina oriundo do respectivo PPC são apresentados logo na sequência.

Exemplo 1 de Aplicação – Plano da Disciplina “A”

O plano da disciplina “A” se refere ao componente curricular de Engenharia de Software I, o qual faz parte da matriz curricular do Curso Superior de Tecnologia em Desenvolvimento de Software Multiplataforma, oferecido pela Faculdade de Tecnologia da Zona Leste (Fatec Zona Leste).

O Quadro 23 descreve os objetivos de aprendizagem do componente curricular selecionado e os dados básicos do curso superior o qual ele pertence, conforme resumo do plano, disponível no Anexo C e redigido com base no referido PPC¹⁸. Cada objetivo de aprendizagem é identificado pelas iniciais do nome do curso (DSM), pela sigla da disciplina (ESWI) e pelo número do objetivo (Obj#).

Quadro 23 – Objetivos de aprendizagem do plano da disciplina “A”

IES	Faculdade de Tecnologia da Zona Leste
Curso	Tecnologia em Desenvolvimento de Software Multiplataforma (DSM)
Duração do curso	6 semestres
Disciplina	Engenharia de Software I (ESWI)
Período de oferta	1º semestre
Identificador	Descrição do Objetivo de Aprendizagem
DSM_ESWI_Obj1	Identificar as características de Sistemas de Informação, seus tipos, viabilidade técnica, características de custo, valor e qualidade da informação.
DSM_ESWI_Obj2	Explicar as características de um sistema, seus componentes e relacionamentos.
DSM_ESWI_Obj3	Compreender o ciclo de vida utilizando concepções do modelo cascata.
DSM_ESWI_Obj4	Utilizar conceitos da UML na análise de requisitos e na elaboração de diagramas focando na modelagem de sistemas.

Fonte: Adaptado do PPC de DSM da Fatec Zona Leste (2021).

Os resultados da aplicação do MPO_MAS na disciplina “A” são sintetizados no

¹⁸ Fatec Zona Leste – Projeto Pedagógico do Curso Superior de Tecnologia em Desenvolvimento de Software Multiplataforma. Disponível em: <http://www.fateczl.edu.br/cursos/desenvolvimento-de-software-multiplataforma>.

Quadro 24, o qual reconhece o verbo de aprendizagem de cada objetivo, identifica o tipo de conhecimento, classifica o nível de compreensão SOLO e relaciona os planos ontológicos escolhidos para serem reutilizados, do nível cognitivo mais alto para o mais baixo.

Quadro 24 – Aplicação do MPO_MAS na disciplina “A”

Objetivo	Verbo(s) de Aprendizagem	Tipo de Conhecimento	Nível SOLO	Plano Ontológico
DSM_ESWI_Obj1	Identificar	Declarativo	Uniestructural	MAS_PM_U
DSM_ESWI_Obj2	Explicar (sentido de descrever)	Declarativo	Multiestructura I	MAS_PM_M MAS_MI_M MAS_PM_U
	Explicar (sentido próprio)	Declarativo	Relacional	MAS_MC_R MAS_MI_R MAS_MI_M MAS_PM_M MAS_PM_U
DSM_ESWI_Obj3	Compreender	Não identificado	Não classificado	Nenhum
DSM_ESWI_Obj4	Utilizar (sentido de fazer uso de => aplicar)	Funcional	Relacional	MAS_MD_R MAS_MC_R MAS_MI_R MAS_MI_M MAS_PM_M MAS_PM_U

Fonte: Elaboração do autor.

Ainda sobre os resultados, não foi possível reconhecer pertinentemente o nível de compreensão desejado para o DSM_ESWI_Obj3. Visando uma utilização proveitosa do modelo, seria significativo uma maior especificidade para esse objetivo.

Exemplo 2 de Aplicação – Plano da Disciplina “B”

O plano da disciplina “B” se refere ao componente curricular de Engenharia de Software II, o qual faz parte da matriz curricular do Curso Superior de Tecnologia em Desenvolvimento de Software Multiplataforma, oferecido pela Faculdade de Tecnologia da Zona Leste (Fatec Zona Leste).

O Quadro 25 descreve os objetivos de aprendizagem do componente curricular

selecionado e os dados básicos do curso superior o qual ele pertence, conforme resumo do plano, disponível no Anexo D e redigido com base no mesmo PPC mencionado na seção imediatamente anterior. Cada objetivo de aprendizagem também é identificado pelas iniciais do nome do curso (DSM), pela sigla da disciplina (ESWII) e pelo número do objetivo (Obj#).

Quadro 25 – Objetivos de aprendizagem do plano da disciplina “B”

IES	Faculdade de Tecnologia da Zona Leste
Curso	Tecnologia em Desenvolvimento de Software Multiplataforma (DSM)
Duração do curso	6 semestres
Disciplina	Engenharia de Software II (ESWII)
Período de oferta	2º semestre
Identificador	Descrição do Objetivo de Aprendizagem
DSM_ESWII_Obj1	Conhecer e aplicar padrões ao processo de software.
DSM_ESWII_Obj2	Mapear modelos de representação.
DSM_ESWII_Obj3	Empregar os diversos tipos de Arquitetura de Software.
DSM_ESWII_Obj4	Conhecer a aplicar padrões de documentação, integração de sistemas e manutenção.

Fonte: Adaptado do PPC de DSM da Fatec Zona Leste (2021).

Os resultados da aplicação do MPO_MAS na disciplina “B” são sintetizados no Quadro 26, o qual reconhece o verbo de aprendizagem de cada objetivo, identifica o tipo de conhecimento, classifica o nível de compreensão SOLO e relaciona os planos ontológicos escolhidos para serem reutilizados, do nível cognitivo mais alto para o mais baixo.

Quadro 26 – Aplicação do MPO_MAS na disciplina “B”

Objetivo	Verbo(s) de Aprendizagem	Tipo de Conhecimento	Nível SOLO	Plano Ontológico
DSM_ESWII_Obj1	Conhecer (sentido de saber explicar) Aplicar	Funcional	Relacional	MAS_MD_R MAS_MA_R MAS_MC_R MAS_MI_R MAS_MI_M MAS_PM_M MAS_PM_U
DSM_ESWII_Obj2	Mapear (sentido de	Funcional	Multiestrutura	MAS_MI_M

	representar elementos graficamente => ilustrar)		I	MAS_PM_M MAS_PM_U
	Mapear (sentido de traduzir o processo de análise em representações => construir)	Funcional	Relacional	MAS_MC_R MAS_MI_R MAS_MI_M MAS_PM_M MAS_PM_U
DSM_ESWII_Obj3	Empregar (sentido de fazer uso de => aplicar)	Funcional	Relacional	MAS_MA_R MAS_MC_R MAS_MI_R MAS_MI_M MAS_PM_M MAS_PM_U
DSM_ESWII_Obj4	Conhecer (sentido de saber explicar) Aplicar	Funcional	Relacional	MAS_MD_R MAS_MA_R MAS_MC_R MAS_MI_R MAS_MI_M MAS_PM_M MAS_PM_U

Fonte: Elaboração do autor.

A respeito dos resultados ainda, cabe frisar que o nível de compreensão desejado para DSM_ESWII_Obj1 e DSM_ESWII_Obj4 foram interpretados em sua maior parte com base no verbo de aprendizagem “Aplicar” enquanto o verbo “Conhecer” foi inferido como “Explicar”, todavia, para uma utilização favorável do modelo, seria significativo a presença de um verbo mais específico no objetivo.

Exemplo 3 de Aplicação – Plano da Disciplina “C”

O plano da disciplina “C” se refere ao componente curricular de Análise Orientada a Objetos, o qual faz parte da matriz curricular do Curso Superior de Tecnologia Análise e Desenvolvimento de Sistemas, oferecido pelo Instituto Federal de São Paulo – Câmpus Bragança Paulista.

O Quadro 27 descreve os objetivos de aprendizagem do componente curricular selecionado e os dados básicos do curso superior o qual ele pertence, conforme

resumo do plano, disponível no Anexo E e redigido com base no referido PPC¹⁹. Cada objetivo de aprendizagem é identificado pelas iniciais do nome do curso (ADS), pela sigla da disciplina (AOO) e pelo número do objetivo (Obj#).

Quadro 27 – Objetivos de aprendizagem do plano da disciplina “C”

IES	Instituto Federal de São Paulo (IFSP) – Câmpus Bragança Paulista
Curso	Tecnologia em Análise e Desenvolvimento de Sistemas (ADS)
Duração do curso	6 semestres
Disciplina	Análise Orientada a Objetos (AOO)
Período de oferta	3º semestre
Identificador	Descrição do Objetivo de Aprendizagem
ADS_AOO_Obj1	Conhecer e aplicar técnicas para levantamento e especificação de requisitos baseadas em casos de uso.
ADS_AOO_Obj2	Conhecer e aplicar técnicas para modelagem estrutural e comportamental de sistemas orientados a objeto.
ADS_AOO_Obj3	Identificar adequadamente aspectos de qualidade da modelagem.
ADS_AOO_Obj4	Utilizar corretamente a notação de modelagem orientada a objetos.
ADS_AOO_Obj5	Compreender a influência da orientação a objeto no processo de desenvolvimento de software.

Fonte: Adaptado do PPC de ADS do Instituto Federal de São Paulo (2020).

Os resultados da aplicação do MPO_MAS na disciplina “C” são sintetizados no Quadro 28, o qual reconhece o verbo de aprendizagem de cada objetivo, identifica o tipo de conhecimento, classifica o nível de compreensão SOLO e relaciona os planos ontológicos escolhidos para serem reutilizados, do nível cognitivo mais alto para o mais baixo.

Quadro 28 – Aplicação do MPO_MAS na disciplina “C”

Objetivo	Verbo(s) de Aprendizagem	Tipo de Conhecimento	Nível SOLO	Plano Ontológico
ADS_AOO_Obj1	Conhecer (sentido de saber explicar) Aplicar	Funcional	Relacional	MAS_MD_R MAS_PM_M MAS_PM_U

19 Instituto Federal de São Paulo – Projeto Pedagógico do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas. Disponível em: <https://bra.ifsp.edu.br/cursos-artigos/121-tecnologia-em-analise-e-desenvolvimento-de-sistemas>.

ADS_AOO_Obj2	Conhecer (sentido de saber explicar) Aplicar	Funcional	Relacional	MAS_MC_R MAS_MI_R MAS_MI_M MAS_PM_M MAS_PM_U
ADS_AOO_Obj3	Identificar (sentido próprio)	Declarativo	Uniestrutural	MAS_PM_U
	Identificar (sentido de classificar ou descrever)	Declarativo	Multiestrutural	MAS_MI_M MAS_PM_M MAS_PM_U
ADS_AOO_Obj4	Utilizar (sentido de fazer uso de => aplicar)	Funcional	Relacional	MAS_MC_R MAS_MI_R MAS_MI_M MAS_PM_M MAS_PM_U
ADS_AOO_Obj5	Compreender (sentido de saber explicar o paradigma orientado a objetos no processo de desenvolvimento)	Funcional	Abstrato Estendido	MAS_MC_R MAS_MI_R MAS_MI_M MAS_PM_M MAS_PM_U

Fonte: Elaboração do autor.

Ainda sobre os resultados, cabe salientar que o nível de compreensão desejado para ADS_AOO_Obj1 e ADS_AOO_Obj2 foram interpretados em sua maior parte com base no verbo de aprendizagem “Aplicar” enquanto o verbo “Conhecer” foi inferido como “Explicar”, todavia, para uma utilização favorável do modelo, seria significativo haver um verbo mais específico em tais objetivos. Já o nível cognitivo esperado para ADS_AOO_Obj5 foi inferido como “Explicar”, contudo, para uma utilização proveitosa do modelo, seria relevante a presença de um verbo de maior especificidade nesse objetivo.

Exemplo 4 de Aplicação – Plano da Disciplina “D”

O plano de disciplina “D” se refere ao componente curricular de Arquitetura de Software, o qual faz parte da matriz curricular do Curso Superior de Tecnologia Análise e Desenvolvimento de Sistemas, oferecido pelo Instituto Federal de São Paulo –

Câmpus Bragança Paulista.

O Quadro 29 descreve os objetivos de aprendizagem do componente curricular selecionado e os dados básicos do curso superior o qual ele pertence, conforme resumo do plano, disponível no Anexo F e redigido com base no mesmo PPC mencionado na seção imediatamente anterior. Cada objetivo de aprendizagem também é identificado pelas iniciais do nome do curso (ADS), pela sigla da disciplina (ASW) e pelo número do objetivo (Obj#).

Quadro 29 – Objetivos de aprendizagem do plano da disciplina “D”

IES	Instituto Federal de São Paulo (IFSP) – Câmpus Bragança Paulista
Curso	Tecnologia em Análise e Desenvolvimento de Sistemas (ADS)
Duração do curso	6 semestres
Disciplina	Arquitetura de Software (ASW)
Período de oferta	4º semestre
Identificador	Descrição do Objetivo de Aprendizagem
ADS_ASW_Obj1	Empregar métodos e técnicas de análise e projeto no processo de desenvolvimento de sistemas de software orientado a objetos.
ADS_ASW_Obj2	Representar a arquitetura de software utilizando notações de modelagem.

Fonte: Adaptado do PPC de ADS do Instituto Federal de São Paulo (2020).

Os resultados da aplicação do MPO_MAS na disciplina “D” são sintetizados no Quadro 30, o qual reconhece o verbo de aprendizagem de cada objetivo, identifica o tipo de conhecimento, classifica o nível de compreensão SOLO e relaciona os planos ontológicos escolhidos para serem reutilizados, do nível cognitivo mais alto para o mais baixo.

Quadro 30 – Aplicação do MPO_MAS na disciplina “D”

Objetivo	Verbo de Aprendizagem	Tipo de Conhecimento	Nível SOLO	Plano Ontológico
ADS_ASW_Obj1	Empregar (sentido de fazer uso para construir representações => construir	Funcional	Relacional	MAS_MC_R MAS_MI_R MAS_MI_M MAS_PM_M MAS_PM_U
ADS_ASW_Obj2	Representar (sentido de traduzir o processo de	Funcional	Relacional	MAS_MA_R MAS_MC_R

	análise em representações, aplicando padrões arquiteturais => aplicar)			MAS_MI_R MAS_MI_M MAS_PM_M MAS_PM_U
--	--	--	--	--

Fonte: Elaboração do autor.

Ainda sobre os resultados, é cabível salientar que o nível de compreensão desejado para ADS_ASW_Obj2 foi interpretado com base no verbo de aprendizagem “Aplicar”, contudo, para uma utilização favorável do modelo, seria relevante a existência de um verbo mais específico nesse objetivo, uma vez que a representação pode acontecer de diversas formas e em diferentes níveis cognitivos.

6 CONCLUSÕES

O concreto no mundo real é abstrato no mundo artificial. O concreto no mundo artificial é abstrato no mundo real.

(Abstração do autor)

Este capítulo final tem o propósito de concluir o presente estudo, apresentar as considerações finais para esta tese, apontar as limitações da pesquisa e sugerir trabalhos futuros.

Este estudo investigou uma lacuna relacionada ao ensino de engenharia de software, com o intuito de permitir ao estudante condições de atingir níveis de complexidade cognitiva crescente na construção de conhecimento concreto e abstrato.

Para responder à questão de pesquisa, o trabalho em evidência propôs um modelo pedagógico de ensino de engenharia de software construído a partir da correlação entre o alinhamento construtivo e os planos organizacionais da técnica OC2-RD2, a fim de auxiliar o professor no planejamento de disciplinas e aulas de engenharia de software, proposição esta definida como objetivo geral desta tese.

Para se alcançar o objetivo geral, se pode afirmar que este trabalho:

- 1- Correlacionou as etapas substanciais do Alinhamento Construtivo e os planos organizacionais da técnica OC2-RD2, com enfoque no plano de conteúdos;
- 2- Concebeu uma ontologia de domínio para a área de modelagem e análise de software;
- 3- Construiu uma arquitetura pedagógica e representou visualmente um metamodelo ontológico;
- 4- Incorporou um conjunto de padrões para contextualizar o ensino de abstração;
- 5- Por fim, validou, ainda que preliminarmente, o modelo em determinados planos de disciplinas, possibilitando, deste modo, obter impressões iniciais a respeito de sua aplicabilidade.

Como resultado desta pesquisa, se originou um modelo pedagógico de ensino de engenharia de software denominado Modelo de Planos Ontológicos de

Modelagem e Análise de Software (MPO_MAS). Tal modelo pode ser visto como um modelo pedagógico de ensino, bem como um modelo ontológico de apoio ao ensino. Cabe repetir aqui que os planos ontológicos temáticos e reutilizáveis presentes no MPO_MAS estão fundamentados no alinhamento construtivo, na taxonomia cognitiva SOLO, no plano organizacional de conteúdos OC2-RD2 e em uma coleção de padrões de abstração. Ademais, os resultados oriundos do conhecimento e experiência dos participantes da pesquisa permitem dizer que o MPO_MAS é resultado de compartilhamento de diversos modelos mentais, dado que este modelo foi construído sob a perspectiva de um grupo de docentes, considerados especialistas na área de conhecimento de Engenharia de Software. Ainda vale lembrar que uma reflexão transformadora não é interessante de ser feita isoladamente, mas sim, de maneira integrada, onde outro colega da mesma área de conhecimento pode contribuir com esse processo reflexivo. É verdade que o MPO_MAS foi construído a partir de uma reflexão transformadora, mas também de maneira conjunta.

A hipótese considerada para este estudo foi validada, uma vez que um modelo pedagógico de ensino de engenharia de software foi idealizado, projetado e materializado, resultando no MPO_MAS, modelo este estruturado para ajudar o professor no planejamento de ensino de engenharia de software, principalmente quando o plano em questão deve conter temas da área de modelagem e análise de software.

Diante do que foi apresentado nesta seção, se pode concluir que este trabalho cumpriu os objetivos específicos estabelecidos e, por conseguinte, o objetivo geral.

6.1 Considerações Finais

Algumas considerações finais devem ser feitas para este trabalho:

- a) É fato que se apresentou aqui uma primeira versão do MPO_MAS cujo metamodelo foi construído sob a perspectiva docente mediante a compilação de muitos e variados resultados, os quais foram analisados e interpretados qualitativamente, portanto, outras perspectivas podem surgir durante sua aplicação;
- b) Em certas passagens textuais no decurso deste trabalho, não houve um termo uniforme para definir se o MPO_MAS é terminologicamente um

modelo, metamodelo ou metametamodelo. Na verdade, ao observar as etapas de aplicação deste modelo (Figura 38), o MPO_MAS é configurado como um metametamodelo, porém a adoção dos outros dois termos não significa estar incorreto, ainda mais a depender do contexto de sua utilização;

- c) Apesar do complemento pedagógico ter sido adotado predominantemente para o termo modelo no decurso deste trabalho, é notório constatar que o modelo proposto retrata uma ontologia de domínio, podendo, dessa forma, também ser visto como um modelo ontológico, assim sendo, se adotou coerentemente ambas as terminologias;
- d) Dado que o MPO_MAS é resultado de uma correlação de duas abordagens pedagógicas, o uso deste modelo pode ser feito por direções distintas. De um lado, o MPO_MAS pode ser utilizado tanto por docentes que desejem apenas usufruir da rede de conteúdo oferecido pela estrutura do modelo ontológico (plano de conteúdos) da técnica OC2-RD2. Por outro lado, o MPO_MAS pode ser usado por professores que já adotam ou queiram adotar a estrutura de narrativas OC2-RD2 em ambientes de aprendizagem e desejem usar os módulos de Alinhamento Construtivo. Não obstante, o docente pode fazer proveito de ambos os lados. Cabe frisar que em todas essas situações, o módulo de Padrões de Abstração está disponível para ser utilizado.

6.2 Limitações da Pesquisa

As limitações observadas nesta pesquisa são:

- a) O tamanho da amostra, visto que os resultados do estudo em foco são provenientes da participação de 10 entrevistados, especificamente na fase de coleta de dados;
- b) A análise qualitativa dos resultados foi realizada via método amplo indutivo e a interpretação para se agrupar os dados por similaridade se deu com base no julgamento do próprio pesquisador;
- c) Por ora, a arquitetura do MPO_MAS cobre uma área de conhecimento de engenharia de software, isto é, modelagem e análise de software;
- d) Não houve aplicação de um pré-teste junto a um especialista com o

intuito de se buscar ajustes no agrupamento, ordem e redação das perguntas contidas no roteiro de entrevista;

- e) As três etapas do MPO_MAS, vide Figura 38, não foram aplicadas na elaboração de planos de disciplina oferecidas no momento, visto que este estudo se concentrou na aplicação da segunda e terceira etapas do modelo, isto é, no nível de metamodelo (plano da disciplina extraído do PPC) e metametamodelo (ontologia de domínio), respectivamente. De qualquer forma, a aplicação do MPO_MAS buscou uma validação preliminar, levando em consideração determinados planos de disciplina (situações-exemplo) selecionados pelo pesquisador deste estudo;
- f) Dos três planos organizacionais OC2-RD2, o estudo em questão enfatizou essencialmente o Plano de Conteúdos, não tanto os demais planos, isto é, plano de motivações e plano de fábulas.

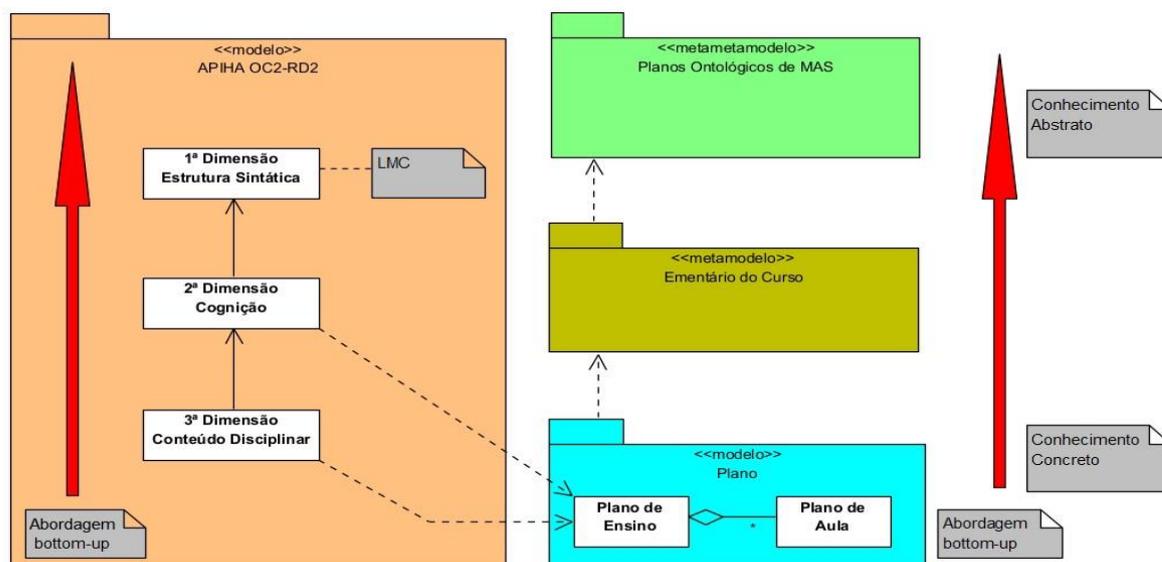
6.3 Sugestões de Trabalhos Futuros

Levando em consideração que a limitação de pesquisa é um ponto relevante para a sugestão de novos estudos, os seguintes trabalhos futuros são sugeridos:

- a) Aplicar o MPO_MAS na elaboração de um plano de ensino no início de um dado semestre letivo e acompanhá-lo no decorrer das aulas. Vale destacar que a clareza e consistência da redação dos objetivos de aprendizagem são aspectos relevantes para uma aplicação adequada do modelo pedagógico;
- b) Refinar os planos ontológicos do MPO_MAS mediante nova pesquisa de campo junto a uma nova amostra de professores, com o intuito de buscar melhorias para um ou mais módulos da arquitetura pedagógica atual;
- c) Reutilizar a estrutura do MPO_MAS para se construir novas ontologias de domínio para outras áreas de conhecimento de Engenharia de Software, havendo, desse modo, a possibilidade de integração entre metamodelos ontológicos interdependentes e complementares;
- d) Elaborar um plano de fábulas OC2-RD2 para algum dos Resultados Pretendidos de Aprendizagem do MPO_MAS e, por conseguinte, planejar um roteiro narrativo para ser adotado na ocasião da oferta de

- um determinado componente curricular;
- e) Uma vez que o MPO_MAS se apresenta atualmente em uma configuração de uso estático, se pode vislumbrar a chance de disponibilizar o modelo para ser utilizado também de maneira dinâmica, por meio do desenvolvimento de ferramenta de software;
 - f) Avistar a possibilidade de modelagem e implementação de um motor de busca semântica das informações disponíveis nos planos ontológicos conforme termos-chave identificados no objetivos de aprendizagem dos próprios planos das disciplinas;
 - g) Averiguar se o modelo pedagógico proposto é passível de integração com as dimensões de conteúdo disciplinar e cognição de Ambiente Presencial Interativo e Híbrido de Aprendizagem (APIHA) proposto por Buttignon (2020) para apoiar o planejamento de roteiros narrativos no ensino de engenharia de software, levando em consideração que o referido ambiente suporta a utilização da técnica OC2-RD2, em duas dimensões, especificamente Cognição e Conteúdo Disciplinar, que podem ser vinculadas semanticamente ao MPO_MAS. Apenas como vislumbre, a Figura 39 exibe uma perspectiva preliminar dessa integração, na qual se nota a 3ª dimensão (Conteúdo Disciplinar) e a 2ª dimensão (Cognição) do APIHA com viabilidade de consumir conteúdos e formas dos planos de ensino e aula produzidos com o suporte da estrutura do MPO_MAS. Cabe salientar também que em ambos os modelos, a abordagem de aplicação é *bottom-up*.

Figura 39 – Perspectiva de integração entre APIHA e MPO_MAS



Fonte: Elaboração do autor com adaptação de Buttignon (2020, p. 142)

Sobre a Figura 39, é conveniente deduzir que a integração entre APIHA e MPO_MAS contribuiria para uma correlação ainda mais completa e consistente entre as etapas do alinhamento construtivo e os planos organizacionais da técnica OC2-RD2, possibilitando abranger a elaboração dos planos de motivações e fábulas, além da criação e implementação de histórias presentes originalmente na técnica de narrativas predita.

REFERÊNCIAS

- ACM; IEEE. **Software Engineering 2014**: curriculum guidelines for undergraduate degree programs in software engineering. Association for Computing Machinery e IEEE Computer Society, 2014. Disponível em: <https://www.acm.org/binaries/content/assets/education/se2014.pdf> e <https://ieeecs-media.computer.org/assets/pdf/se2014.pdf>.
- ACM;IEEE. **Computing Curricula 2020**: paradigms for global computing education. Association for Computing Machinery e IEEE Computer Society, 2020. Disponível em: <https://www.acm.org/binaries/content/assets/education/curricula-recommendations/cc2020.pdf>.
- BAL, Mieke. **Narratology: Introduction to the theory of narrative**. 2nd.ed. University of Toronto Press, 2009.
- BAL, Mieke. **Narratology: Introduction to the theory of narrative**. 4th.ed. University of Toronto Press, 2017.
- BLAHA, Michael; RUMBAUGH, James. **Object-oriented modeling and design with UML**. 2nd.ed. Pearson Prentice Hall, 2005.
- BECKER, Fernando. **Educação e construção do conhecimento**: revista e ampliada. Porto Alegre: Penso, 2012.
- BEHAR, Patrícia Alejandra; PASSERINO, Liliana Maria; BERNARDI, Maira. Modelos Pedagógicos para Educação a Distância: pressupostos teóricos para a construção de objetos de aprendizagem. **RENOTE: revista novas tecnologias na educação**. Porto Alegre, RS, 2007.
- BIGGS, John. Aligning teaching and assessing to course objectives. **Teaching and learning in higher education: New trends and innovations**, Aveiro, v. 2, p. 1-9, abr. 2003. Disponível em: https://www.researchgate.net/publication/241251310_Aligning_teaching_and_assessing_to_course_objectives.
- BIGGS, John. Modes of learning, forms of knowing, and ways of schooling. In: DEMETRIOU, Andreas; SHAYER, Michael; EFKLIDES, Anastasia (Org.). **Neo-Piagetian theories of cognitive development: Implications and applications for education**. Routledge, 2016, cap. 2, p. 30-51.
- BIGGS, John B.; COLLIS, Kevin F. **Evaluating the quality of learning**: the SOLO taxonomy (Structure of the Observed Learning Outcome). Academic Press, 1982.
- BIGGS, John; TANG, Catherine. **Teaching for quality learning at university**. 4th.ed. McGraw-hill education, 2011.
- BOOCH, Grady. Architecture as a shared hallucination. **IEEE Software**, v. 27, n. 1, p. 96-96, 2009.

BOURQUE, Pierre; FAIRLEY, Richard E., eds. **Guide to the Software Engineering Body of Knowledge (SWEBOK)**. Version 3.0. IEEE Computer Society, 2014.
Disponível em: <https://www.computer.org/education/bodies-of-knowledge/software-engineering>.

BROOKS JR, Frederick P. No Silver Bullet: essence and accidents of software engineering. Computer. **IEEE Computer Society**, Washington, DC, v. 10, 1987.

BROOKS JR, Frederick P. **The mythical man-month**: essays on software engineering. Pearson Education, 1995.

BRUEGGE, Bernd; DUTOIT, Allen H. **Object-oriented software engineering**: using UML, patterns and Java. Learning, v. 5, n. 6, p. 7, 2009.

BUTTIGNON, Karina. **Projeto de uma linguagem de modelagem de cenários para um ambiente presencial, interativo e híbrido de aprendizagem OC2-RD2**. 2020. 206f. Tese de doutorado – Pontifícia Universidade Católica de São Paulo, 2020.

CHOURIO, Patricia; AZEVEDO, Romualdo; CASTRO, Alberto; GADELHA, Bruno. Most common errors in software modeling using UML. In: **Proceedings of the XXXIII Brazilian Symposium on Software Engineering**. 2019. p. 244-253.

COOPER, K. et al. Teaching experiences with UML at the University of Texas at Dallas. In: **Educators Symposium of the 8th International Conference on Model Driven Engineering Languages and Systems**. 2005.

CORNEY, Malcolm et al. Some empirical results for neo-Piagetian reasoning in novice programmers and the relationship to code explanation questions. In: **Proceedings of the fourteenth australasian computing education conference**. 2012. p. 77-86.

CRESWELL, John W. **Projeto de pesquisa**: métodos qualitativo, quantitativo e misto. Porto Alegre: Artmed, 2010.

CUNHA, José A. O. G.; MARQUES, Gabriel A.; LEMOS, Wellington L.; CAMARA JR, Uélio D.; VASCONCELOS, Francisco J. S. Software engineering education in Brazil: a mapping study. In: **Proceedings of the XXXII Brazilian Symposium on Software Engineering**. 2018. p. 348-356.

ENGELS, Gregor et al. Teaching UML is teaching software engineering is teaching abstraction. In: **International Conference on Model Driven Engineering Languages and Systems**. Springer, Berlin, Heidelberg, 2005. p. 306-319.

FALKNER, Katrina; VIVIAN, Rebecca; FALKNER, Nickolas JG. Neo-piagetian forms of reasoning in software development process construction. In: **2013 Learning and Teaching in Computing and Engineering**. IEEE, 2013. p. 31-38.

FACULDADE DE TECNOLOGIA DA ZONA LESTE. Projeto Pedagógico de Curso.

São Paulo: FATECZL, 2021. Disponível em:
<http://www.fateczl.edu.br/cursos/desenvolvimento-de-software-multiplataformas>.
 Acesso em: 19 dez. 2022.

FERREIRA, Thaís et al. Identifying emerging topics and difficulties in software engineering education in Brazil. In: **Proceedings of the XXXII Brazilian Symposium on Software Engineering**. 2018. p. 230-239.

GIRAFFA, Maria M.; MORA, Michael C. Evasão na disciplina de algoritmo e programação: um estudo a partir dos fatores intervenientes na perspectiva do aluno. In: **Congressos CLABES**. 2013.

GRAY, David E. **Pesquisa no mundo real**. 2.ed. Porto Alegre: Penso, 2009.

GUARINO, Nicola. Formal ontology in information systems. In: **Proceedings of the first international conference (FOIS'98)**, Trento, jun. 1998. Amsterdam, IOS Press, p. 3-15.

GLUGA, Richard et al. On the reliability of classifying programming tasks using a neo-piagetian theory of cognitive development. In: **Proceedings of the ninth annual international conference on International computing education research**. 2012. p. 31-38.

HALLADAY, Steve; WIEBEL, Michael. **Object-Oriented software engineering**. Lawrence, KS: R&D Publications, Inc., 1993.

HAZZAN, Orit. The reflective practitioner perspective in software engineering education. **Journal of Systems and Software**, v. 63, n. 3, p. 161-171, 2002.

HAZZAN, Orit. The influence of software intangibility on computer science and software engineering education. **ACM SIGSOFT Software Engineering Notes**, v. 32, n. 3, p. 7-8, 2007.

HAZZAN, Orit; KRAMER, Jeff. Abstraction in computer science & software engineering: A pedagogical perspective. **System Design Frontier Journal**, v. 3, n. 12, p. 1-9, 2006.

HAZZAN, Orit; KRAMER, Jeff. Assessing abstraction skills. **Communications of the ACM**, v. 59, n. 12, p. 43-45, 2016.

HUITT, William; HUMMEL, John. Piaget's theory of cognitive development. **Educational psychology interactive**, v. 3, n. 2, 2003. Disponível em:
<http://www.edpsycinteractive.org/topics/cognition/piaget.html>.

HUMPHREY, Watts S. The software engineering process: definition and scope. In: **Proceedings of the 4th international software process workshop on Representing and enacting the software process**. 1988. p. 82-83.

INSTITUTO FEDERAL DE SÃO PAULO. Projeto Pedagógico de Curso. Bragança Paulista: IFSPBRA, 2020. Disponível em: <https://bra.ifsp.edu.br/cursos-artigos/121->

[tecnologia-em-analise-e-desenvolvimento-de-sistemas](#). Acesso em: 19 dez. 2022.

JACKSON, Daniel. **Software Abstractions: logic, language, and analysis**. MIT press, 2012.

KRAMER, Jeff. Abstraction-is it teachable? 'the devil is in the detail'. In: **Proceedings 16th Conference on Software Engineering Education and Training**. IEEE Computer Society, 2003. p. 32-32.

KRAMER, Jeff. Is abstraction the key to computing?. **Communications of the ACM**, v. 50, n. 4, p. 36-42, 2007.

KRATHWOHL, David R. **A revision of Bloom's taxonomy: an overview**. *Theory into practice*, v. 41, n. 4, p. 212-218, 2002.

LARMAN, Craig. **Applying UML and patterns: an introduction to object-oriented analysis and design and iterative development**. 2nd.ed. Upper Saddle River, NJ: Pearson, 2002.

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

LIMA, José Vinícius et al. As Metodologias Ativas e o Ensino em Engenharia de Software: uma revisão sistemática da literatura. In: **Anais do Workshop de Informática na Escola**. 2019. p. 1014-1023.

LIMA, José V. V.; SILVA, Cleverton A. D.; ALENCAR, Fernanda M. R.; SANTOS, Wylliams B. Metodologias Ativas como forma de reduzir os desafios do ensino em Engenharia de Software: diagnóstico de um survey. In: **Anais do XXXI Simpósio Brasileiro de Informática na Educação**. SBC, 2020. p. 172-181.

LISKOV, Barbara; GUTTAG, John. **Program development in JAVA: abstraction, specification, and object-oriented design**. Pearson Education, 2000.

LISTER, Raymond. Concrete and other neo-Piagetian forms of reasoning in the novice programmer. In: **Conferences in research and practice in information technology series**. 2011.

LUTZ, Stacey T.; HUITT, William G. **Connecting cognitive development and constructivism: Implications from Theory for Instruction and Assessment**. 2004. Disponível em: <http://www.edpsycinteractive.org/papers/cogdev.pdf>.

MEC. **Diretrizes Curriculares Nacionais para os Cursos de Graduação em Computação**. Resolução CNE/CES nº 5, de 16 de novembro de 2016. 9p, 2016. Disponível em: <http://portal.mec.gov.br/docman/novembro-2016-pdf/52101-rces005-16-pdf/file>.

MOL, Solange M.; MATOS, Daniel A. S. Uma análise sobre a Taxonomia SOLO: aplicações na avaliação educacional. **Estudos em Avaliação Educacional**, v. 30, n. 75, p. 722-747, 2019.

MORRA, Sergio et al. **Cognitive development: neo-Piagetian perspectives**. Psychology Press, 2012.

OLIVEIRA JR, Edson et al. Ensino, Aprendizagem e Uso Profissional da UML em Maringá e Região. In: **Anais do XXIX Workshop sobre Educação em Computação**. SBC, 2021. p. 328-337.

OMG. **Unified Modeling Language Specification**. Version 2.5.1, 2017.

PARNAS, David L. Using mathematical models in the inspection of critical software. **Applications of Formal Methods, International Series in Computer Science**, p. 17-31, 1995.

PARNAS, David L. Software engineering programs are not computer science programs. **IEEE software**, v. 16, n. 6, p. 19-30, 1999.

PARNAS, David L. The use of mathematics in software quality assurance. **Frontiers of Computer Science**, v. 6, n. 1, p. 3-16, 2012.

PIAGET, Jean. **Biologia e conhecimento: ensaios sobre as regulações orgânicas e os processos cognoscitivos**. 3.ed. Petrópolis: Vozes, 2000.

PIAGET, Jean. **Fazer e compreender**. São Paulo: Melhoramentos - EDUSP, 1978.

PRESSMAN, Roger S. **Engenharia de software: uma abordagem profissional**. 7.ed. Porto Alegre: AMGH, 2011.

PRESSMAN, Roger S.; MAXIM, Bruce. **Software engineering: a practitioner's approach**. 8th.ed. McGraw-Hill Education, 2015.

RIZZO, Adriana Justina; POLETTI, Elaine Cristina Catapani. Alinhamento Construtivo e Taxonomia SOLO: relações com o Sistema de Domínio Khan Academy. **Revista Brasileira de Informática na Educação**, v. 29, p. 1091-1106, 2021.

RIZZO, Adriana J.; POLETTI, Elaine C. C. Conexão do alinhamento construtivo e metodologias ativas. In: **XIV Workshop de Pós-Graduação e Pesquisa do Centro Paula Souza**. Tecnologia, inovação e sustentabilidade: 50 anos de Cursos de Tecnologia no Brasil, p. 182-191, 2019.

ROBERTS, Patricia. Abstract thinking: a predictor of modelling ability?. In: **Educators Symposium of the ACM/IEEE 12th International Conference on Model Driven Engineering Languages and Systems**. Springer, 2009. p. 753-754.

SANTOS, Maria E. S.; ROCHA, Tamires S.; PERKUSICH, Mirko B. O ensino de engenharia de software no nível superior: um mapeamento sistemático. **Revista Principia**. n. 56, p. 116-125, 2021.

SIVERS, Derek. **Kurt Vonnegut explains drama**. Artigo de 01 setembro de 2009. Disponível em: <https://sivers.org/drama>. Acesso em: 03 jul. 2020.

VEGA, Ítalo S. Fábulas OCC-RDD: histórias didáticas para ambientes interativos híbridos e presenciais de aprendizagem. **Revista da Associação Brasileira de Tecnologia Educacional**, v. 31, p. 105-118, 2016.

VEGA, Ítalo. S. **Elaboração de histórias OC2-RD2**. 2018. 60 f. Relatório de Projeto PIPAD 5095 – Pontifícia Universidade Católica de São Paulo, São Paulo.

VEGA, Ítalo, S. **ES – Engenharia de Software**: aula ## – assunto?. 2021. ##f. Notas de Aula.

VENDRAMEL, Wilson; GUIRELLI, Henrique; VEGA, Ítalo S. A utilização de estruturas narrativas OC2-RD2 no ensino de computação: um relato de experiência. In: **Deflagração de ações voltadas à formação docente**. Ponta Grossa: Atena Editora, 2020, v.1, p. 107-115.

WIRFS-BROCK, Rebecca J. Principles in Practice. **IEEE Software**, v. 26, n. 4, p. 11-12, 2009.

WIRTH, Niklaus. A brief history of software engineering. **IEEE Annals of the History of Computing**, v. 30, n. 3, p. 32-39, 2008.

YOURDON, Edgard; ARGILA, Carl. **Análise e projeto orientados a objetos: estudos de casos**. São Paulo: Makron Books, 1999.

ZORZO, Avelino F.; NUNES, Daltro; MATOS, Ecivaldo S.; STEINMACHER, Igor; LEITE, Jair C.; ARAUJO, Renata; CORREIA, Ronaldo C. M.; MARTINS, Simone.

Referenciais

de Formação para os Cursos de Graduação em Computação. Sociedade Brasileira de Computação (SBC). 153p, 2017. Disponível em:

<https://www.sbc.org.br/documentos-da-sbc/summary/127-educacao/1155-referenciais-de-formacao-para-cursos-de-graduacao-em-computacao-outubro-2017>.

APÊNDICE A – Procedimento de Pesquisa

Roteiro de Entrevista

Este roteiro de entrevista contém seis seções de perguntas, divididas desta forma:

- a) A seção A é formada por perguntas que identificam o perfil do(a) participante da pesquisa;
- b) A seção B é composta por perguntas preliminares sobre desenvolvimento cognitivo no ensino e aprendizagem de modelagem e análise de software;
- c) A seção C é constituída por perguntas a respeito do alinhamento construtivo no ensino e aprendizagem de modelagem e análise de software;
- d) A seção D é formada por perguntas sobre padrões para contextualizar a abstração no ensino e aprendizagem de modelagem e análise de software;
- e) A seção E é composta por perguntas sobre itens de conhecimento e erros de abstração no ensino e aprendizagem de modelagem e análise de software;
- f) Finalmente, o grupo F é constituído por perguntas gerais a respeito do modelo pedagógico de correlação proposto.

A – Perfil do(a) Participante

A1) Titulação máxima:

- a) Graduação
- b) Especialização
- c) Mestrado
- d) Doutorado
- e) Livre Docência
- f) Outra:

A2) Anos de experiência como docente em cursos superiores de computação:

- a) 0-2
- b) 3-5
- c) 6-10
- d) 11-15
- e) mais de 15

A3) Organização Acadêmica onde atua (mais de uma alternativa pode ser respondida):

- a) Faculdade
- b) Centro Universitário
- c) Instituto
- d) Universidade
- e) Outra:

A4) Programa/Curso de atuação (mais de uma alternativa pode ser respondida):

- a) Sistemas de Informação
- b) Ciência da Computação
- c) Análise e Desenvolvimento de Sistemas
- d) Engenharia de Software
- e) Engenharia da Computação
- f) Ciência de Dados
- g) Gestão de Tecnologia da Informação
- h) Outro:

A5) Disciplina ministrada (mais de uma alternativa pode ser respondida):

- a) Engenharia de Software
- b) Qualidade de Software
- c) Arquitetura de Software
- d) Análise e Projeto Orientado a Objetos
- e) Programação Orientada a Objetos
- f) Laboratório de Programação
- g) Banco de Dados
- h) Outra:

B – Desenvolvimento cognitivo no ensino e aprendizagem de modelagem e análise de software

Essas perguntas foram baseadas na visão neopiagetiana de desenvolvimento cognitivo, especificamente nos estudos de Lister (2011), Gluga *et al.* (2012) e Falkner, Vivian e Falkner (2013), além dos trabalhos de Biggs e Collis (1982) e Biggs (1992).

B1) Visando refinar o processo de ensino e aprendizagem em modelagem e análise de software, os tópicos dessa área de conhecimento devem ser planejados para o estudante compreendê-los como:

- a) Processos.
- b) Artefatos.
- c) Processos e Artefatos.
- d) Nem como Processos nem como Artefatos.

B2) Buscando aprimorar o processo de ensino e aprendizagem em modelagem e análise de software, o professor deve planejar suas aulas para o estudante:

- a) Demonstrar raciocínio indutivo; pensar de maneira intuitiva; conseguir fazer e compreender abstrações individuais.
- b) Demonstrar raciocínio dedutivo; pensar de maneira concreta; conseguir fazer e compreender abstrações de situações reais.
- c) Demonstrar raciocínio hipotético-dedutivo; pensar de maneira abstrata; conseguir fazer e compreender abstrações que levam a realizações concretas.
- d) Demonstrar os três grupos de comportamentos, facilitados tanto de baixo para cima quanto de cima para baixo.

B3) A fim de melhorar o processo de ensino e aprendizagem em modelagem e análise de software, o professor deve planejar suas aulas para o estudante:

- a) Construir e compreender abstrações dentro do mesmo domínio do problema, se concentrando em apenas uma propriedade abstrata em um determinado momento.
- b) Construir e compreender abstrações dentro do mesmo domínio do problema, se concentrando em poucas propriedades abstratas e isoladas em um determinado momento.
- c) Construir e compreender abstrações dentro do mesmo domínio do problema, se concentrando em várias propriedades abstratas em um determinado momento, mas restritas a situações familiares e reais.
- d) Construir e compreender abstrações que ultrapassem o domínio do problema, se concentrando em várias propriedades abstratas em um determinado momento sobre situações ainda não vivenciadas diretamente.
- e) As quatro habilidades de abstração, facilitadas tanto de baixo para cima quanto de cima para baixo.

C – Alinhamento construtivo no ensino e aprendizagem de modelagem e análise de software

Para responder as perguntas desta seção, é preciso se apoiar no Quadro 1, o qual descreve a área de conhecimento de Modelagem e Análise de Software do currículo de Engenharia de Software (ACM; IEEE, 2004; ACM; IEEE, 2014), como também no Quadro 2, a qual elenca as unidades (módulos temáticos) e tópicos (conteúdos) dessa área de conhecimento (ACM; IEEE, 2014).

Quadro 1 – Descrição da área de modelagem e análise de software

Modelagem e Análise podem ser consideradas conceitos centrais em qualquer disciplina de engenharia de software, porque são essenciais para documentar e avaliar decisões e alternativas de projeto.

A modelagem e análise são aplicadas primeiro à análise, especificação e validação de requisitos. Os requisitos representam as necessidades reais dos usuários, clientes e outras partes interessadas afetadas pelo sistema. A construção de requisitos inclui uma análise da viabilidade do sistema desejado, elicitación e análise das necessidades das partes interessadas, a criação de uma descrição precisa do que o sistema deve e não deve fazer, juntamente com quaisquer restrições à sua operação e implementação, e a validação desta descrição ou especificação pelas partes interessadas.

Fonte: Adaptado de ACM e IEEE (2004, p. 25) e ACM e IEEE (2014, p.31).

Quadro 2 – Unidades e tópicos da área de modelagem e análise de software

Unidades	Tópicos
Fundamentos de Modelagem (FM)	FM1: Princípios de modelagem (por exemplo, decomposição, abstração, generalização, projeção/visões, e uso de abordagens formais)
	FM2: Pré-condições, pós-condições, invariantes e design por contrato
	FM3: Introdução aos modelos matemáticos e notação formal
Tipos de Modelos (TM)	TM1: Modelagem de informações (por exemplo, modelo entidade-relacionamento e diagramas de classe)
	TM2: Modelagem comportamental (por exemplo, diagramas de estado, análise de casos de uso, diagramas de interação, modos de falha e análise de efeitos, e análise de árvore de falha)
	TM3: Modelagem arquitetural (por exemplo, padrões de arquitetura e diagramas de componente)
	TM4: Modelagem de domínio (por exemplo, abordagens de engenharia de domínio)
	TM5: Modelagem empresarial (por exemplo, processos de negócios, organizações, metas e fluxo de trabalho)
	TM6: Modelagem de sistemas embarcados (por exemplo, análise de escalonamento em tempo real e protocolos de interface)
Fundamentos de Análise (FA)	FA1: Análise da construção (por exemplo, completude, consistência, e robustez)
	FA2: Análise da corretude (por exemplo, análise estática,

simulação, e verificação de modelo)
FA3: Análise da confiabilidade (por exemplo, análise de modo de falha e árvores de falha)
FA4: Análise formal (por exemplo, prova de teoremas)

Fonte: Adaptado de ACM e IEEE (2014, p. 31).

C1) Quais tópicos de modelagem e análise podem elevar a noção de abstração no ensino de engenharia de software? Identifique cinco tópicos.

Unidade	Tópico

A fim de auxiliar a identificação dos verbos pretendidos de aprendizagem, o Quadro 3 apresenta uma lista de verbos, de baixo a alto nível cognitivo e divididos em dois tipos de conhecimento: declarativo e funcional. Tais verbos são derivados dos níveis da taxonomia cognitiva SOLO (BIGGS; COLLIS, 1982; BIGGS; TANG, 2011).

Quadro 3 – Verbos de conhecimento declarativo e funcional por nível SOLO

Nível	Conhecimento Declarativo	Conhecimento Funcional
Uniestructural	Memorizar, identificar, recitar	Contar, combinar, ordenar
Multiestructural	Descrever, classificar	Calcular, ilustrar
Relacional	Comparar e contrastar, explicar, argumentar, analisar	Aplicar, construir, traduzir, resolver problemas próximos, prever dentro do mesmo domínio
Abstrato Estendido	Teorizar, hipotetizar, generalizar	Refletir e melhorar, inventar, criar, resolver problemas invisíveis, extrapolar para domínios desconhecidos

Fonte: Adaptado de Biggs e Tang (2011, p. 124).

C2) Qual tipo de conhecimento deve ser aprendido para o conteúdo identificado? a) Declarativo, o qual se refere a conhecer as coisas e se manifesta ao que é declarado de maneira falada e escrita ou; b) Funcional, o qual se refere ao conhecimento declarativo posto em ação.

C3) Qual o nível de compreensão ou desempenho deve ser alcançado para cada tópico? Identifique um verbo de aprendizagem para se aprender o conteúdo em questão.

Tópico	Tipo de Conhecimento	Verbo de Aprendizagem

C4) A partir de suas respostas, quais os resultados pretendidos de aprendizagem (RPA) para os tópicos de modelagem e análise de software, visando aumentar a noção de abstração no ensino de engenharia de software? Os estudantes devem ser capazes de:

Identificador	Descrição do Resultado Pretendido de Aprendizagem
RPA1	
RPA2	
RPA3	
RPA4	
RPA5	

C5) Qual atividade de ensino e aprendizagem pode incentivar o engajamento tanto do professor quanto do aluno para atingir o resultado pretendido de aprendizagem? O Quadro 4 lista, não de modo exclusivo, técnicas de ensino e aprendizagem para ajudar na resposta.

RPA	Técnica de Ensino e Aprendizagem
RPA1	
RPA2	
RPA3	
RPA4	
RPA5	

Quadro 4 – Técnicas de ensino e aprendizagem

- Aprendizagem baseada em investigação;
- Aprendizagem baseada em problema;
- Aprendizagem baseada em projeto;
- Estudo de Caso;
- Gamificação;
- Outra.

Fonte: Autor (2022) com adaptações de Biggs e Tang (2011).

C6) Qual tarefa de avaliação pode motivar o estudante nesse processo como também permitir ao professor observar se o resultado pretendido de aprendizagem foi atingido? O Quadro 5 relaciona, não exclusivamente, tarefas de avaliação para auxiliar na resposta.

RPA	Tarefa de Avaliação
RPA1	
RPA2	
RPA3	
RPA4	
RPA5	

Quadro 5 – Tarefas de Avaliação

- Teste de múltipla escolha;
- Avaliação escrita com questões abertas;
- Avaliação de resolução de problemas;
- Mapa conceitual;
- Portfólio de aprendizagem;
- Diário de bordo;
- Avaliação por pares;
- Autoavaliação;
- Projetos individuais;
- Projetos em grupo;
- Apresentação oral (projeto, solução de problema, experiência);
- A própria técnica de ensino e aprendizagem escolhida na questão anterior;
- Outra.

Fonte: Autor (2022) com adaptações de Biggs e Tang (2011).

D – Padrões para contextualizar a abstração no ensino e aprendizagem de modelagem e análise de software

Os padrões apresentados na sequência buscam contextualizar diferentes propósitos, formas e níveis de abstração em modelagem e análise nas aulas de engenharia de software. Os padrões foram adaptados de Hazzan e Kramer (2006; 2016).

D1a) O padrão 1 é adequado para contextualizar a abstração em modelagem e análise de software?

Padrão 1: Dadas duas representações de um sistema específico, os alunos são solicitados a explicar qual representação é mais abstrata e por quê. Essas representações podem ser o código do programa, um diagrama UML, uma descrição escrita, uma foto e assim por diante.

Nada adequado

1	2	3	4	5	6	7	8	9	10

 Bastante adequado

D1b) O padrão 1 pode ser instanciado para ajudar a atingir qual resultado pretendido de aprendizagem?

() Nenhum () RPA1 () RPA2 () RPA3 () RPA4 () RPA5 () Todos

D2a) O padrão 2 é adequado para contextualizar a abstração em modelagem e análise de software?

Padrão 2: Categorização de uma coleção de representações de diferentes sistemas de acordo com a escolha de abstração do estudante.

Nada adequado

1	2	3	4	5	6	7	8	9	10

 Bastante adequado

D2b) O padrão 2 pode ser instanciado para ajudar a atingir qual resultado pretendido de aprendizagem?

() Nenhum () RPA1 () RPA2 () RPA3 () RPA4 () RPA5 () Todos

D3a) O padrão 3 é adequado para contextualizar a abstração em modelagem e análise de software?

Padrão 3: Descrição de um sistema específico, com o qual os alunos estão familiarizados, em diferentes níveis de abstração.

Nada adequado

1	2	3	4	5	6	7	8	9	10

 Bastante adequado

D3b) O padrão 3 pode ser instanciado para ajudar a atingir qual resultado pretendido de aprendizagem?

() Nenhum () RPA1 () RPA2 () RPA3 () RPA4 () RPA5 () Todos

D4a) O padrão 4 é adequado para contextualizar a abstração em modelagem e análise de software?

Padrão 4: Dada uma representação do sistema, os alunos são solicitados a fornecer uma representação mais abstrata do que a fornecida e uma representação menos abstrata do que a fornecida.

Nada adequado

1	2	3	4	5	6	7	8	9	10

 Bastante adequado

D4b) O padrão 4 pode ser instanciado para ajudar a atingir qual resultado pretendido de aprendizagem?

Nenhum RPA1 RPA2 RPA3 RPA4 RPA5 Todos

D5a) O padrão 5 é adequado para contextualizar a abstração em modelagem e análise de software?

Padrão 5: Os alunos são solicitados a explicar algum tópico (sistema, representação...) para outra pessoa. Eles são solicitados a descrever como explicariam o tópico escolhido em dois casos: quando a pessoa para quem estão explicando é capaz de ver esse tópico e quando não pode vê-lo. Também são solicitados a explicar as considerações que orientaram a formulação de cada descrição e como essas formulações se relacionam com diferentes níveis de abstração.

Nada adequado

1	2	3	4	5	6	7	8	9	10

 Bastante adequado

D5b) O padrão 5 pode ser instanciado para ajudar a atingir qual resultado pretendido de aprendizagem?

Nenhum RPA1 RPA2 RPA3 RPA4 RPA5 Todos

D6a) O padrão 6 é adequado para contextualizar a abstração em modelagem e análise de software?

Padrão 6: Os alunos são solicitados a explicar algum tópico (sistema, representação...) com uma determinada restrição ou restrições. Os alunos também são convidados a explicar suas considerações na escolha de suas explicações.

Nada adequado

1	2	3	4	5	6	7	8	9	10

 Bastante adequado

D6b) O padrão 6 pode ser instanciado para ajudar a atingir qual resultado pretendido de aprendizagem?

Nenhum RPA1 RPA2 RPA3 RPA4 RPA5 Todos

D7a) O padrão 7 é adequado para contextualizar a abstração em modelagem e análise de software?

Padrão 7: Os alunos são solicitados a sugerir uma metáfora ou uma analogia para um determinado sistema.

	1	2	3	4	5	6	7	8	9	10	
Nada adequado											Bastante adequado

D7b) O padrão 7 pode ser instanciado para ajudar a atingir qual resultado pretendido de aprendizagem?

Nenhum RPA1 RPA2 RPA3 RPA4 RPA5 Todos

D8a) O padrão 8 é adequado para contextualizar a abstração em modelagem e análise de software?

Padrão 8: Perguntas de reflexão: este tipo de perguntas pode ser usado em muitas situações. Eles convidam os alunos a refletir sobre seus processos de pensamento e, assim, aumentar o nível de abstração de seus processos mentais.

	1	2	3	4	5	6	7	8	9	10	
Nada adequado											Bastante adequado

D8b) O padrão 8 pode ser instanciado para ajudar a atingir qual resultado pretendido de aprendizagem?

Nenhum RPA1 RPA2 RPA3 RPA4 RPA5 Todos

E – Itens de conhecimento e erros de abstração no ensino e aprendizagem de modelagem e análise de software
--

E1) A fim de atingir os resultados pretendidos de aprendizagem, qual item de conhecimento (IC) é visto como requisito, recomendação e sugestão para contextualizar a abstração no ensino de tópicos de modelagem e análise de software? As relações entre itens de conhecimento são baseadas na elaboração do plano organizacional de conteúdos da técnica OC2-RD2 (VEGA, 2018).

RPA (foco no tópico de conteúdo)	Padrão(ões) de Abstração	IC (requisito)	IC (recomendação)	IC (sugestão)

E2) Considerando que o erro faz parte do processo de aquisição de conhecimento, os erros elencados no Quadro 6 são adequados para contextualizar a abstração no ensino de modelagem e análise de software e, conseqüentemente, contribuir para se atingir os resultados pretendidos de aprendizagem? Vale ressaltar que esses erros são destacados por Halladay e Wiebel (1993).

Quadro 6 – Quatro erros comuns de abstração

- | |
|--|
| <ul style="list-style-type: none"> • Particionar elementos illogicamente para um nível de abstração; • Reduzir ou pular níveis de abstração; • Não identificar todos os elementos para um nível de abstração; • Identificar elementos inexistentes para um nível de abstração. |
|--|

Fonte: Adaptado de Halladay e Wiebel (1993).

Nada adequado	<table border="1"> <tr> <td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td> </tr> <tr> <td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td> </tr> </table>	1	2	3	4	5	6	7	8	9	10											Bastante adequado
1	2	3	4	5	6	7	8	9	10													

F – Modelo pedagógico de correlação proposto
--

F1) A existência de um modelo pedagógico que apresente planos de conteúdos temáticos e reutilizáveis de modelagem e análise de software o qual busca auxiliar o professor na elaboração de planos de disciplina e na preparação de aulas pode a vir se tornar um recurso educacional adequado para o planejamento de ensino de engenharia de software?

	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="width: 20px; text-align: center;">1</td> <td style="width: 20px; text-align: center;">2</td> <td style="width: 20px; text-align: center;">3</td> <td style="width: 20px; text-align: center;">4</td> <td style="width: 20px; text-align: center;">5</td> <td style="width: 20px; text-align: center;">6</td> <td style="width: 20px; text-align: center;">7</td> <td style="width: 20px; text-align: center;">8</td> <td style="width: 20px; text-align: center;">9</td> <td style="width: 20px; text-align: center;">10</td> </tr> <tr> <td style="width: 20px; text-align: center;"> </td> </tr> </table>	1	2	3	4	5	6	7	8	9	10											
1	2	3	4	5	6	7	8	9	10													
Nada adequado		Bastante adequado																				

F2) A existência de um modelo pedagógico que direcione atividades de ensino e aprendizagem e tarefas de avaliação para atingir resultados pretendidos de aprendizagem em relação aos tópicos de modelagem e análise de software pode a vir se tornar um recurso educacional adequado para o planejamento de ensino de engenharia de software?

	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="width: 20px; text-align: center;">1</td> <td style="width: 20px; text-align: center;">2</td> <td style="width: 20px; text-align: center;">3</td> <td style="width: 20px; text-align: center;">4</td> <td style="width: 20px; text-align: center;">5</td> <td style="width: 20px; text-align: center;">6</td> <td style="width: 20px; text-align: center;">7</td> <td style="width: 20px; text-align: center;">8</td> <td style="width: 20px; text-align: center;">9</td> <td style="width: 20px; text-align: center;">10</td> </tr> <tr> <td style="width: 20px; text-align: center;"> </td> </tr> </table>	1	2	3	4	5	6	7	8	9	10											
1	2	3	4	5	6	7	8	9	10													
Nada adequado		Bastante adequado																				

F3) A existência de um modelo pedagógico que presume a contextualização da abstração em modelagem e análise de software, oferecendo lentes de abstração distintas na abordagem dos conteúdos pode a vir se tornar um recurso educacional adequado para o planejamento de ensino de engenharia de software?

	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="width: 20px; text-align: center;">1</td> <td style="width: 20px; text-align: center;">2</td> <td style="width: 20px; text-align: center;">3</td> <td style="width: 20px; text-align: center;">4</td> <td style="width: 20px; text-align: center;">5</td> <td style="width: 20px; text-align: center;">6</td> <td style="width: 20px; text-align: center;">7</td> <td style="width: 20px; text-align: center;">8</td> <td style="width: 20px; text-align: center;">9</td> <td style="width: 20px; text-align: center;">10</td> </tr> <tr> <td style="width: 20px; text-align: center;"> </td> </tr> </table>	1	2	3	4	5	6	7	8	9	10											
1	2	3	4	5	6	7	8	9	10													
Nada adequado		Bastante adequado																				

F4) Uma fábula, comumente narrada por meio de uma história, costuma mexer com a emoção das pessoas enquanto é contada para um determinado público. A narrativa de uma história em sala de aula poderia causar emoção ao estudante, em outras palavras, provocar o desequilíbrio/equilíbrio cognitivo do sujeito que aprende, mediante o grau de aprendizagem durante a narração da história, onde essa equilíbrio oscilaria entre a presença e a ausência de conhecimento sobre o tema abordado. Tal narração pode ser traduzida como, usando aqui palavras de Becker (2012), um jogo cognitivo de assimilações e acomodações. Posto isto, a existência de um modelo pedagógico que apresente dimensões de conteúdo e de cognição pode ser um recurso educacional adequado para apoiar o planejamento de roteiros narrativos no ensino de engenharia de software?

	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="width: 20px; text-align: center;">1</td> <td style="width: 20px; text-align: center;">2</td> <td style="width: 20px; text-align: center;">3</td> <td style="width: 20px; text-align: center;">4</td> <td style="width: 20px; text-align: center;">5</td> <td style="width: 20px; text-align: center;">6</td> <td style="width: 20px; text-align: center;">7</td> <td style="width: 20px; text-align: center;">8</td> <td style="width: 20px; text-align: center;">9</td> <td style="width: 20px; text-align: center;">10</td> </tr> <tr> <td style="width: 20px; text-align: center;"> </td> </tr> </table>	1	2	3	4	5	6	7	8	9	10											
1	2	3	4	5	6	7	8	9	10													
Nada adequado		Bastante adequado																				

F5) Além do que já foi respondido, você tem outra contribuição para viabilizar a proposta do modelo pedagógico?

F6) Você tem alguma sugestão de melhoria para as perguntas apresentadas?

APÊNDICE B – Termo de Consentimento Livre e Esclarecido

TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO (TCLE)

(Parecer Consubstanciado do CEP da PUC-SP: 5.734.131)

Você está sendo convidado(a) como voluntário(a) a participar da pesquisa intitulada “Ensino de engenharia de software a partir de uma perspectiva neopiagetiana e do processo de abstração: proposição de um modelo pedagógico de correlação entre o alinhamento construtivo e a técnica OC2-RD2”.

A presente pesquisa é motivada pela investigação de um recurso educacional para apoiar o professor no planejamento de ensino de engenharia de software com a finalidade de ajudar o estudante a atingir níveis de complexidade cognitiva crescente na aquisição de conhecimento de conceitos concretos e abstratos.

O objetivo desse projeto é propor um modelo pedagógico de ensino baseado na correlação entre o alinhamento construtivo e a técnica narrativa OC2-RD2 para auxiliar o professor no planejamento de disciplinas e aulas de engenharia de software.

Para a fase de coleta de dados será utilizado um roteiro de entrevista semiestruturada, na qual sua participação consistirá em responder perguntas seguindo um roteiro de entrevista ao pesquisador do projeto. Essa entrevista somente será gravada mediante sua autorização. O tempo de duração da entrevista é de aproximadamente uma hora.

Ao participar desta pesquisa, o benefício esperado é o educador de cursos de computação ter à disposição um plano organizacional de conteúdos de modelagem e análise de software, divididos em regiões temáticas e reutilizáveis, elaborados sob a perspectiva docente, e que possam ser incorporados em um modelo pedagógico, a fim de orientar o professor, do ponto de vista de desenvolvimento cognitivo e progressão do aprendizado, no planejamento de ensino de engenharia de software.

Os riscos identificados na participação desta pesquisa são mínimos, especificamente desconforto e cansaço. Se houver desconforto diante de alguma pergunta, você pode não responder a essa indagação; havendo cansaço durante a entrevista, você pode interrompê-la ou mesmo desistir de participar.

Cabe ressaltar que você não terá nenhum custo para participar deste estudo, tampouco receberá qualquer vantagem financeira por sua participação.

O pesquisador irá tratar a sua identidade com padrões profissionais de sigilo e todos os dados coletados servirão apenas para fins de pesquisa. Seu nome ou

material que indique a sua participação não será liberado sem a sua permissão. Você não será identificado(a) em nenhuma publicação que possa resultar deste estudo.

Você será esclarecido(a) sobre a pesquisa em qualquer tempo que desejar. Você é livre para se recusar a participar, retirar seu consentimento ou interromper a participação a qualquer momento, sendo sua participação voluntária e a recusa em participar não irá acarretar qualquer penalidade.

Este documento deverá ser assinado em duas vias, de acordo com o disposto na resolução 466/2012 do Conselho Nacional de Saúde.

Ciente e conforme com o que foi anteriormente exposto, eu, _____, de forma livre e esclarecida, manifesto meu consentimento em participar desta pesquisa. Declaro que recebi cópia deste termo, e autorizo a realização da pesquisa e a divulgação dos dados obtidos neste estudo.

Data: ____/____/____.

Assinatura do Participante

Assinatura do Pesquisador

Pesquisador: Wilson Vendramel

Telefone: (11) 97514-1970 | e-mail: pwvendramel@uol.com.br

Comitê de Ética em Pesquisa da PUC-SP: Rua Ministro Godói, 969 – Perdizes – São Paulo – SP – CEP: 05015-001.

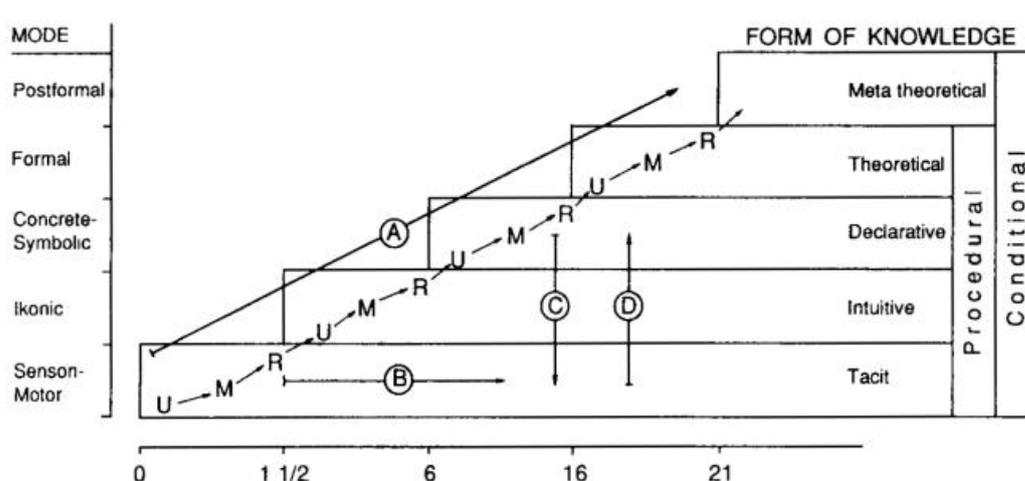
Telefone: (11) 3670-8466 | e-mail: cometica@pucsp.br

ANEXO A – Representações de Apoio para o Roteiro de Entrevista

Artefato constituído por uma coleção de representações (figuras e quadros) utilizado como apoio ao instrumento de coleta de dados da presente pesquisa.

Seção B - Desenvolvimento Cognitivo

Figura 1 – Modos, ciclos de aprendizagem e formas de conhecimento



Fonte: Biggs (1992, p. 39).

Quadro 1 – Tipos qualitativos diferentes de desempenho cognitivo

- (a) curso de desenvolvimento ótimo.
- (b) curso de aprendizagem dentro de uma modalidade (intramodal);
- (c) facilitação de cima para baixo da aprendizagem de ordem inferior (multimodal);
- (d) facilitação de baixo para cima da aprendizagem de ordem superior (multimodal).

Fonte: Adaptado de Biggs (1992, p. 38-39).

Seção C – Alinhamento Construtivo

Quadro 2 – Descrição da área de modelagem e análise de software

Modelagem e Análise podem ser consideradas conceitos centrais em qualquer disciplina de engenharia de software, porque são essenciais para documentar e avaliar decisões e alternativas de projeto.

A modelagem e análise são aplicadas primeiro à análise, especificação e validação de requisitos. Os requisitos representam as necessidades reais dos usuários, clientes e outras partes interessadas afetadas pelo sistema. A construção de requisitos inclui uma análise da viabilidade do sistema desejado, elicitación e análise das necessidades das partes interessadas, a criação de uma descrição precisa do que o sistema deve e não deve fazer, juntamente com quaisquer restrições à sua operação e implementação, e a validação desta descrição ou especificação pelas partes interessadas.

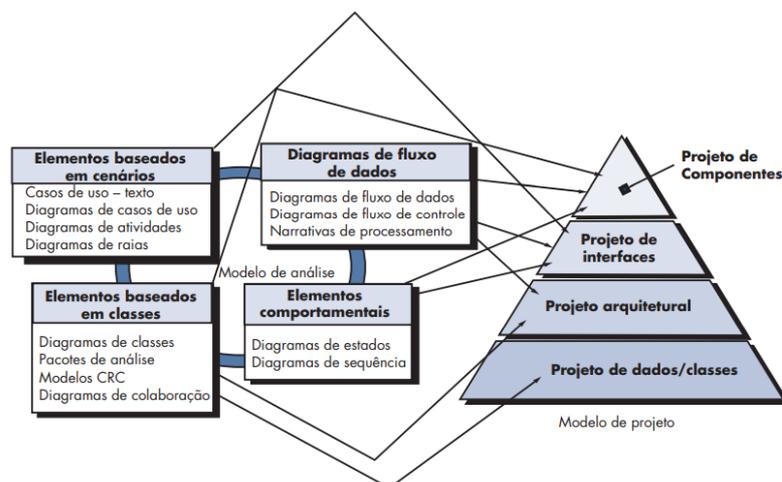
Fonte: Adaptado de ACM e IEEE (2004, p. 25) e ACM e IEEE (2014, p.31).

Quadro 3 – Unidades e tópicos da área de modelagem e análise de software

Unidades	Tópicos
Fundamentos de Modelagem (FM)	FM1: Princípios de modelagem (por exemplo, decomposição, abstração, generalização, projeção/visões, e uso de abordagens formais)
	FM2: Pré-condições, pós-condições, invariantes e design por contrato
	FM3: Introdução aos modelos matemáticos e notação formal
Tipos de Modelos (TM)	TM1: Modelagem de informações (por exemplo, modelo entidade-relacionamento e diagramas de classe)
	TM2: Modelagem comportamental (por exemplo, diagramas de estado, análise de casos de uso, diagramas de interação, modos de falha e análise de efeitos, e análise de árvore de falha)
	TM3: Modelagem arquitetural (por exemplo, padrões de arquitetura e diagramas de componente)
	TM4: Modelagem de domínio (por exemplo, abordagens de engenharia de domínio)
	TM5: Modelagem empresarial (por exemplo, processos de negócios, organizações, metas e fluxo de trabalho)
	TM6: Modelagem de sistemas embarcados (por exemplo, análise de escalonamento em tempo real e protocolos de interface)
Fundamentos de Análise (FA)	FA1: Análise da construção (por exemplo, completude, consistência, e robustez)
	FA2: Análise da corretude (por exemplo, análise estática, simulação, e verificação de modelo)
	FA3: Análise da confiabilidade (por exemplo, análise de modo de falha e árvores de falha)
	FA4: Análise formal (por exemplo, prova de teoremas)

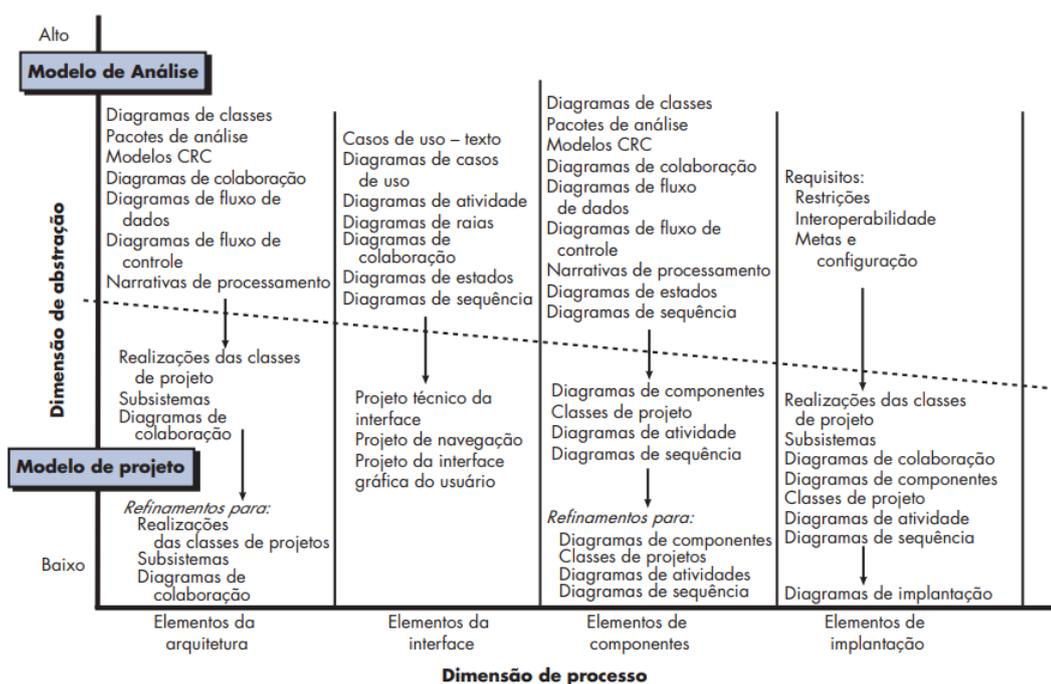
Fonte: Adaptado de ACM e IEEE (2014, p. 31).

Figura 2 – Transição do modelo de análise para o modelo de projeto



Fonte: Pressman (2011, p. 208).

Figura 3 – Dimensão de abstração e de processo do modelo de análise para o modelo de projeto



Fonte: Pressman (2011, p. 221).

Quadro 4 – Verbos de conhecimento declarativo e funcional por nível SOLO

Nível	Conhecimento Declarativo	Conhecimento Funcional
Uniestructural	Memorizar, identificar, recitar	Contar, combinar, ordenar
Multiestructural	Descrever, classificar	Calcular, ilustrar
Relacional	Comparar e contrastar, explicar, argumentar, analisar	Aplicar, construir, traduzir, resolver problemas próximos, prever dentro do mesmo domínio
Abstrato Estendido	Teorizar, hipotetizar, generalizar	Refletir e melhorar, inventar, criar, resolver problemas invisíveis, extrapolar para domínios desconhecidos

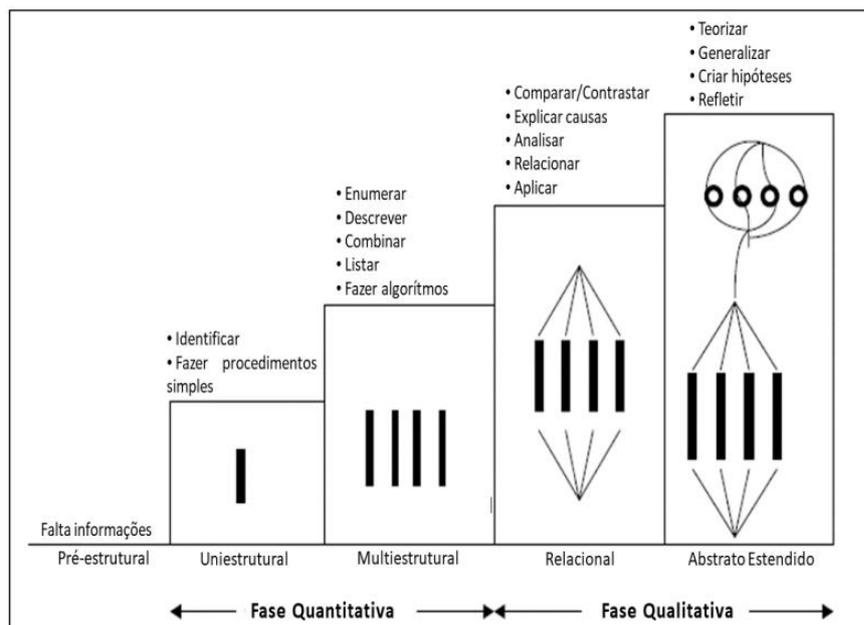
Fonte: Adaptado de Biggs e Tang (2011, p. 124).

Quadro 5 – Modos e níveis da taxonomia SOLO

Modo	Nível Estrutural
Próximo	5. Abstrato Estendido. O estudante agora generaliza a estrutura para incorporar recursos novos e mais abstratos, representando um modo de operação novo e superior.
Alvo	4. Relacional. O estudante agora integra as partes umas às outras, de modo que o todo tenha uma estrutura e um significado coerentes. 3. Multiestructural. O estudante aprende recursos cada vez mais relevantes ou corretos, mas não os integra. 2. Uniestructural. O estudante se concentra no domínio relevante e pega um aspecto com o qual trabalhar.
Anterior	1. Pré-estrutural. Diante de uma tarefa, o estudante está distraído ou enganado por um aspecto irrelevante pertencente a um estágio ou modo anterior.

Fonte: Adaptado de Biggs (1992, p. 38).

Figura 4 – Uma hierarquia dos verbos da taxonomia SOLO



Fonte: Biggs (2003) *apud* Rizzo e Poletti (2021, p. 1094).

Quadro 6 – Exemplos de RPAs para um curso (disciplina) de Sistemas de Informação

RPA1	Descrever os conceitos básicos de sistemas de informação, sua composição, configuração e arquitetura, incluindo a Internet e as tecnologias baseadas na web em particular.
RPA2	Explicar os aspectos sociais, econômicos, regulatórios, políticos e principalmente éticos no desenvolvimento, implementação e uso de sistemas de informação em ambientes de negócios internacionais.
RPA3	Aplicar os conhecimentos gerais e metodologias de sistemas de informação, incluindo o uso de hardware e software, para conceber e avaliar soluções eficazes para problemas de negócios internacionais, atendendo às necessidades de informação.
RPA4	Projetar e desenvolver construções e modelos particulares para apoiar vários níveis de atividades de negócios internacionais usando diferentes ferramentas como Microsoft FrontPage, Microsoft Access e Microsoft Excel.
RPA5	Trabalhar de forma produtiva em equipe e, em particular, comunicar e apresentar informações de forma eficaz em formatos escritos e eletrônicos em um ambiente colaborativo.

Fonte: Adaptado de Biggs e Tang (2011, p. 337).

Quadro 7 – Técnicas de ensino e aprendizagem

<ul style="list-style-type: none"> • Aprendizagem baseada em investigação; • Aprendizagem baseada em problema; • Aprendizagem baseada em projeto; • Estudo de Caso; • Gamificação; • Outra.

Fonte: Elaboração do autor com adaptações de Biggs e Tang (2011).

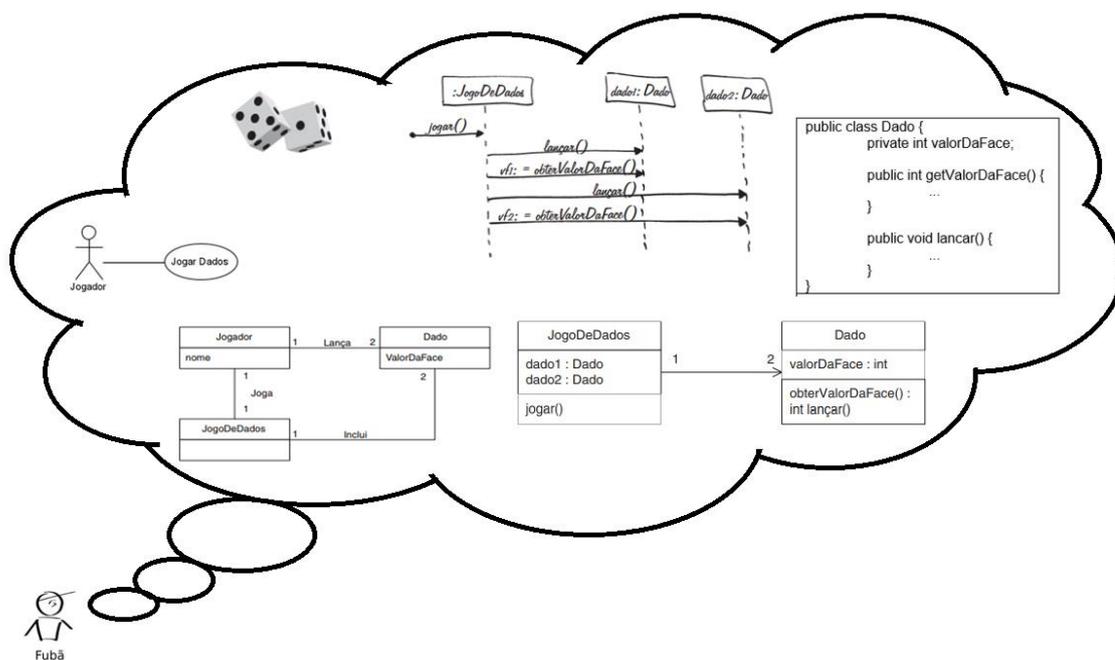
Quadro 8 – Tarefas de avaliação

- Teste de múltipla escolha;
- Avaliação escrita com questões abertas;
- Avaliação de resolução de problemas;
- Mapa conceitual;
- Portfólio de aprendizagem;
- Diário de bordo;
- Avaliação por pares;
- Autoavaliação;
- Projetos individuais;
- Projetos em grupo;
- Apresentação oral (projeto, solução de problema, experiência);
- A própria técnica de ensino e aprendizagem escolhida na questão anterior;
- Outra.

Fonte: Elaboração do autor com adaptações de Biggs e Tang (2011).

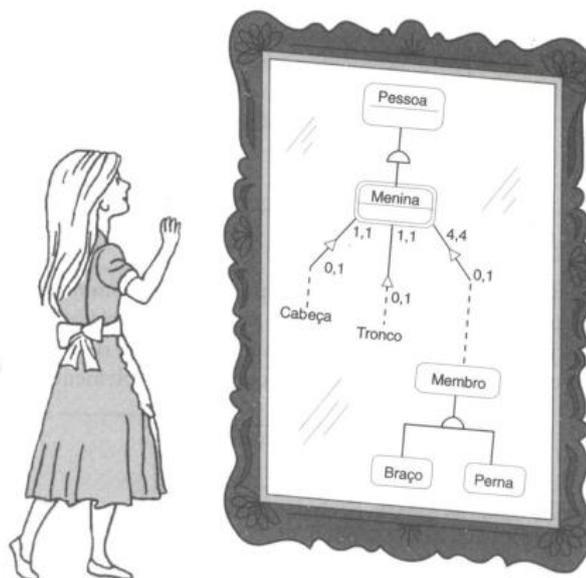
Seção D – Padrões de Abstração

Figura 5 – Modelagem parcial de abstrações do personagem Fubã



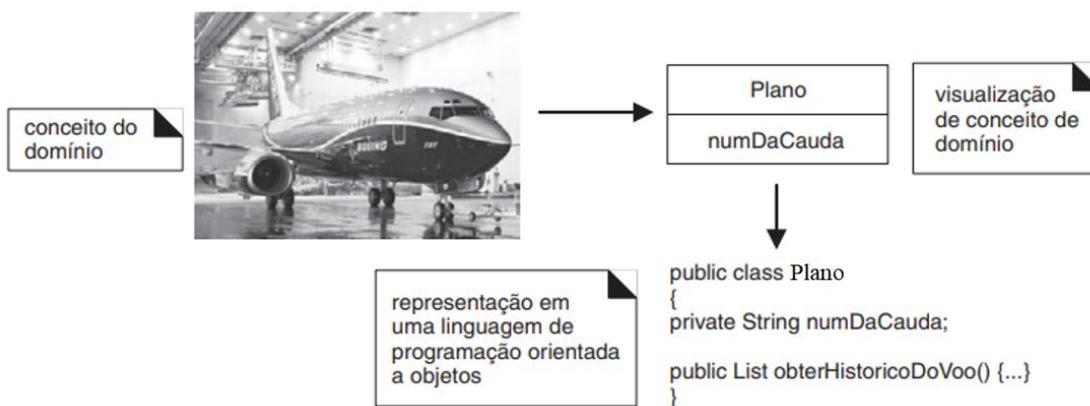
Fonte: Elaboração do autor com adaptações de Larman (2007, p. 36-38).

Figura 6 – O espelho da orientação a objetos



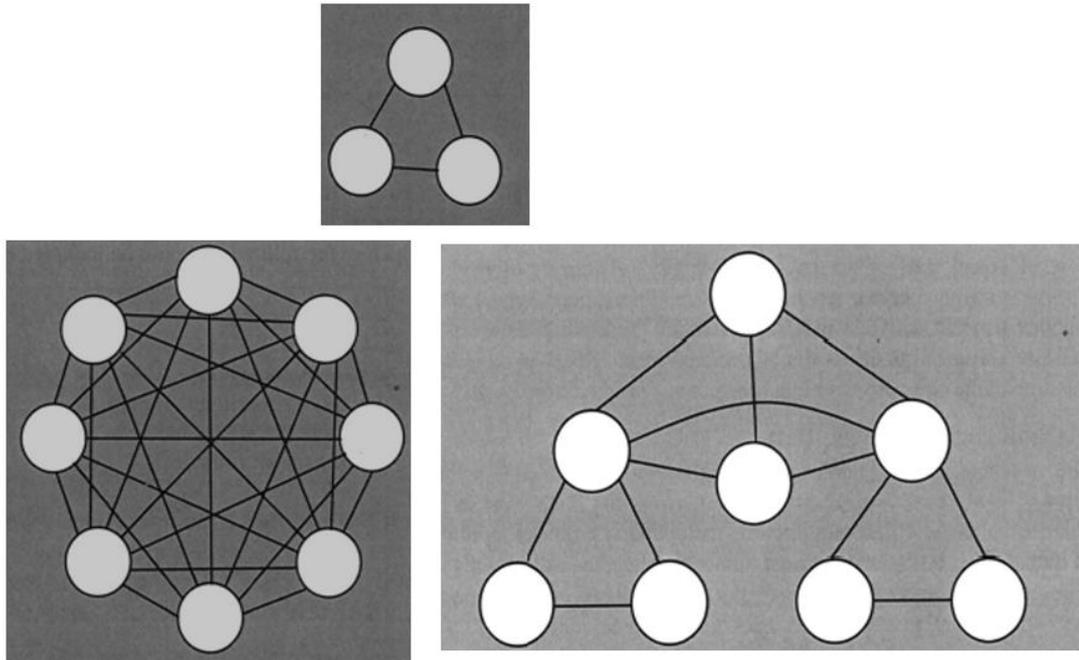
Fonte: Yourdon e Argila (1999, p. 9).

Figura 7 – A orientação a objetos enfatiza a representação de objetos



Fonte: Adaptado de Larman (2007, p. 35).

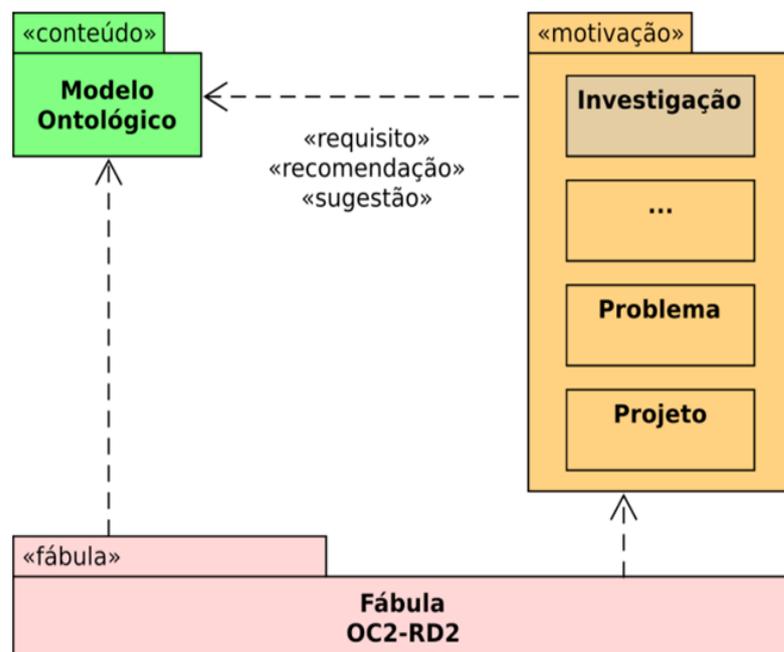
Figura 8 – Usando abstração para desenvolvimento de software: relacionamentos entre objetos



Fonte: Adaptado de Halladay e Wiebel (1993, p. 35-37).

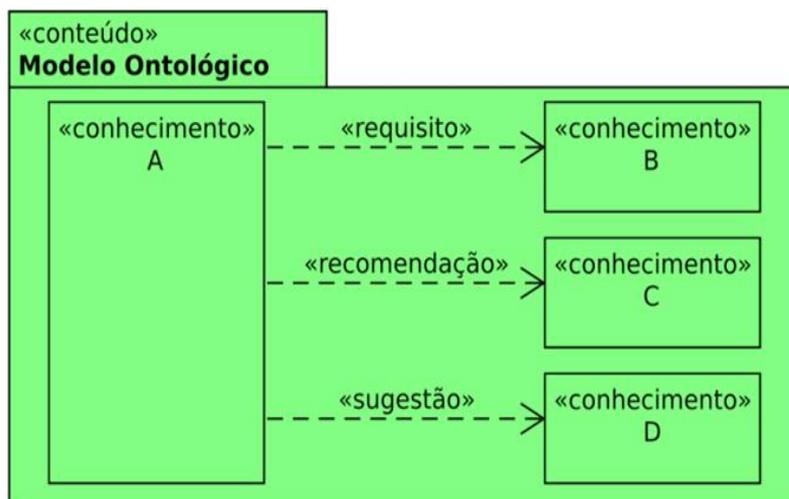
Seção E – Itens de Conhecimento e Erros de Abstração

Figura 9 – Planos organizacionais OC2-RD2



Fonte: Vega (2018, p. 17).

Figura 10 – Tipos de dependência entre itens de conhecimento



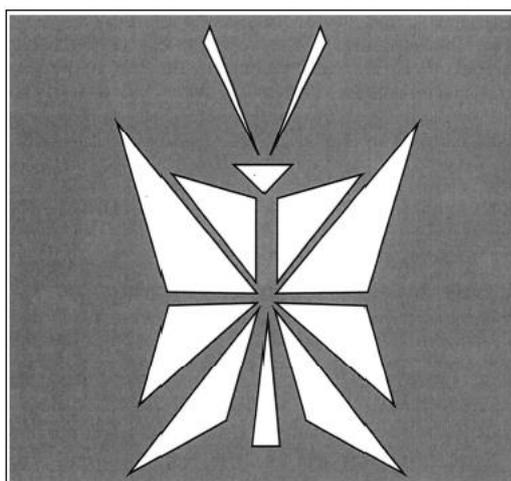
Fonte: Vega (2018, p. 18).

Quadro 9 – Erros comuns de abstração

- a) Particionar elementos illogicamente para um nível de abstração (Elemento Detalhe);
- b) Reduzir ou pular níveis de abstração (Elemento Estranho);
- c) Não identificar todos os elementos para um nível de abstração (Elemento Omisso);
- d) Identificar elementos inexistentes para um nível de abstração (Elemento Alienígena).

Fonte: Adaptado de Halladay e Wiebel (1993) e Vega (2019).

Figura 11 – Uma imagem abstrata



Fonte: Halladay e Wiebel (1993, p. 33).

ANEXO B – Parecer de Mérito Acadêmico



Pontifícia Universidade Católica de São Paulo
Programa de Estudos Pós-Graduados em Tecnologias da
Inteligência e Design Digital

PUC-SP

São Paulo, 16 de agosto de 2022

Ao Comitê de Ética em Pesquisa da PUC/SP

A pesquisa intitulada *Ensino de engenharia de software a partir de uma perspectiva neopiagetiana e do processo de abstração: proposição de um modelo pedagógico de correlação entre o alinhamento construtivo e a técnica OC2-RD2*, de autoria de **Wilson Vendramel**, doutorando do PEPG em Tecnologias da Inteligência e Design Digital desta Universidade, sob a orientação do Prof. Dr. Italo Vega Santiago, tem como objetivo propor um modelo pedagógico de ensino baseado na correlação entre o alinhamento construtivo e a técnica narrativa OC2-RD2 para auxiliar o professor no planejamento de disciplinas e aulas de engenharia de software.

O modelo proposto se baseia em uma concepção neopiagetiana, organizando as fases do alinhamento construtivo, sustentado pela taxonomia SOLO, e as conectando aos planos da técnica OC2-RD2, a fim de nortear o docente, sob o ponto de vista cognitivo, a planejar cursos e aulas de acordo com os resultados de aprendizagem pretendidos. Esse modelo também visa oferecer formas e conteúdos que permitam a contextualização da abstração no ensino de engenharia de software, aspirando assim o desenvolvimento do pensamento abstrato e da capacidade de abstração, a qual é vista como uma habilidade-chave para os estudantes de cursos de computação.

As referências estudadas e as justificativas apresentadas evidenciam a pertinência e oportunidade deste estudo, o qual busca investigar a seguinte questão de pesquisa. Como apoiar o professor no planejamento de ensino de engenharia de software com a finalidade de ajudar o estudante a atingir níveis de complexidade cognitiva crescente na aquisição de conhecimento de conceitos concretos e abstratos?

O estudo é de abordagem qualitativa e propósito exploratório, a qual visa aplicar a entrevista semiestruturada como procedimento de pesquisa. O roteiro de entrevista foi elaborado a fim de se coletar dados junto a uma amostra de professores de instituições de ensino distintas, docentes estes atuantes no ensino de engenharia de software. Após a fase de coleta, os dados provenientes do conhecimento e experiência dos docentes serão analisados e interpretados para se projetar um conjunto de planos de conteúdos temáticos e reutilizáveis, assim como alinhados construtivamente aos resultados de aprendizagem pretendidos, para apoiar a preparação de cursos e aulas de engenharia de software em outros tempos e espaços.

O Termo de Consentimento Livre e Esclarecido apresenta os objetivos da pesquisa, garantindo o direito de resposta às dúvidas, de não haver quaisquer sanções ou prejuízos pela não participação ou desistência da pesquisa, como também a inexistência de qualquer incentivo ou ônus financeiro aos participantes, atendendo assim a Resolução CNS 466/12.

Prof. Dra. Ana Maria Di Grado Hessel
Vice-Coordenadora do Programa de Estudos
Pós-Graduados em Tecnologias da Inteligência e Design Digital

ANEXO C – Plano da Disciplina “A”

O plano da disciplina “A”, referente ao componente curricular de Engenharia de Software I, foi extraído do Projeto Pedagógico do Curso Superior de Tecnologia em Desenvolvimento de Software Multiplataforma, oferecido pela Faculdade de Tecnologia da Zona Leste (Fatec Zona Leste).

Quadro 1 – Componente Curricular de Engenharia de Software I

Objetivos	Identificar as características de Sistemas de Informação, seus tipos, viabilidade técnica, características de custo, valor e qualidade da informação.
	Explicar as características de um sistema, seus componentes e relacionamentos.
	Compreender o ciclo de vida utilizando concepções do modelo cascata.
	Utilizar conceitos da UML na análise de requisitos e na elaboração de diagramas focando na modelagem de sistemas
Ementa	Introdução à Análise de Sistemas. Modelos de Ciclo de Vida de Software. Modelos de Processos de Desenvolvimento de Software (Modelo em Cascata, Espiral e Prototipagem). Definição e classificação de Requisitos de Software (funcionais e não funcionais). Técnicas de Levantamento de Requisitos. Modelo de Negócios aplicado ao levantamento de Requisitos (Canvas). Estudo de Viabilidade. Técnicas de documentação. Metodologias para desenvolvimento de sistemas.
Bibliografia Básica	BEZERRA, Eduardo. Princípios de Análise e Projeto de Sistemas com UML. 3 ed. Rio de Janeiro: Elsevier, 2015. PRESSMAN, Roger; MAXIM, Bruce. Engenharia de Software. 8 ed. São Paulo: McGraw Hill Brasil, 2016. SOMMERVILLE, Ian. Engenharia De Software. 10 ed. São Paulo: Pearson Brasil, 2019.
Bibliografia Complementar	LARMAN, Craig. Utilizando UML e padrões. 3 ed. Porto Alegre: Bookman, 2007. REZENDE, Denis Alcides. Engenharia de software e sistemas de informação. 3 ed. Rio de Janeiro: Brasport, 2005. WASLAWICK Raul. Análise e Projeto de Sistemas de Informação Orientados a Objetos. 2 ed. Rio de Janeiro: Elsevier, 2010.

Fonte: Adaptado do PPC de DSM da Fatec Zona Leste (2021, p. 22-23).

ANEXO D – Plano da Disciplina “B”

O plano da disciplina “B”, referente ao componente curricular de Engenharia de Software II, foi extraído do Projeto Pedagógico do Curso Superior de Tecnologia em Desenvolvimento de Software Multiplataforma, oferecido pela Faculdade de Tecnologia da Zona Leste (Fatec Zona Leste).

Quadro 2 – Componente Curricular de Engenharia de Software II

Objetivos	Conhecer e aplicar padrões ao processo de software.
	Mapear modelos de representação.
	Empregar os diversos tipos de Arquitetura de Software.
	Conhecer a aplicar padrões de documentação, integração de sistemas e manutenção.
Ementa	Modelos de Processo de Desenvolvimento de Software (Espiral e Prototipagem). Modelagem de Sistemas utilizando (UML). Modelagem de Processos do Negócio (BPMN). Técnicas de documentação e Definition of Done (DoD). Padrões de Projeto e Projeto de Arquitetura de Software. Conceitos de Testes de Software. Implantação de Software. Evolução funcional e Manutenção corretiva de Software.
Bibliografia Básica	PRESSMAN, Roger; MAXIM, Bruce. Engenharia de Software. 8 ed. São Paulo: McGraw Hill Brasil, 2016.
	SOMMERVILLE, Ian. Engenharia De Software. 10 ed. São Paulo: Pearson Brasil, 2019.
	WAZLAWICK, R. S. Engenharia de Software: conceitos e práticas. Rio de Janeiro: Elsevier, 2013.
Bibliografia Complementar	LENGHOLM JR. Hélio. Engenharia de Software na Prática. São Paulo: Novatec, 2010.
	GUEDES, G. T. A. UML 2 - uma abordagem prática. São Paulo: Novatec, 2011.
	HIRAMA, K. Engenharia de Software: qualidade e produtividade com tecnologia. Rio de Janeiro: Elsevier, 2011.
	PAULA FILHO, W. P. Engenharia de software. Rio de Janeiro: LTC, 2010.
	PETERS, James. Engenharia de software: Teoria e prática. Rio de Janeiro: Campus, 2001.
	SBROCCO, J. H. T. C.; MACEDO, P. C. Metodologias Ágeis: Engenharia de

	<p>Software sob medida. São Paulo: Érica, 2012.</p> <p>TSUI, F.; KARAM, O. Fundamentos de Engenharia de Software. São Paulo: LTC, 2013.</p>
--	---

Fonte: Adaptado do PPC de DSM da Fatec Zona Leste (2021, p. 29-30).

ANEXO E – Plano da Disciplina “C”

O plano da disciplina “C”, referente ao componente curricular de Análise Orientada a Objetos, foi extraído do Projeto Pedagógico do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, oferecido pelo Instituto Federal de São Paulo – Câmpus Bragança Paulista.

Quadro 3 – Componente Curricular de Análise Orientada a Objetos

Objetivos	Conhecer e aplicar técnicas para levantamento e especificação de requisitos baseadas em casos de uso.
	Conhecer e aplicar técnicas para modelagem estrutural e comportamental de sistemas orientados a objeto.
	Identificar adequadamente aspectos de qualidade da modelagem.
	Utilizar corretamente a notação de modelagem orientada a objetos.
	Compreender a influência da orientação a objeto no processo de desenvolvimento de software.
Ementa	O componente curricular aborda a análise e modelagem de sistemas seguindo o paradigma da orientação a objetos, utilizando os padrões de notação de modelagem orientada a objetos.
Bibliografia Básica	<p>GUEDES, G. T. A. UML 2: uma abordagem prática. 2.ed. São Paulo: Novatec, 2011.</p> <p>LARMAN, C. Utilizando UML e padrões: uma introdução à análise e projeto orientado a objetos e ao desenvolvimento iterativo. 3.ed. Porto Alegre: Bookman, 2007.</p> <p>WAZLAWICK, R. S. Análise e projeto de sistemas de informação orientados a objetos. 2.ed. Rio de Janeiro: Campus Elsevier, 2011.</p>
Bibliografia Complementar	<p>BEZERRA, E. Princípios de análise e projeto de sistemas com UML. 3.ed. Rio de Janeiro: Campus Elsevier, 2015.</p> <p>BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. UML: guia do usuário. 2.ed. Rio de Janeiro: Campus Elsevier, 2006.</p> <p>FURGERI, S. Modelagem de sistemas orientados a objetos: ensino didático. São Paulo: Érica, 2013.</p> <p>GOES, W. M. Aprenda UML por meio de estudos de caso. São Paulo: Novatec,</p>

	2014. SILVA, R. P. UML2 em modelagem orientada a objetos. Florianópolis: Visual Books, 2007.
--	---

Fonte: Adaptado do PPC de ADS do Instituto Federal de São Paulo (2020, p. 104-105).

ANEXO F – Plano da Disciplina “D”

O plano da disciplina “D”, referente ao componente curricular de Arquitetura de Software, foi extraído do Projeto Pedagógico do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, oferecido pelo Instituto Federal de São Paulo – Câmpus Bragança Paulista.

Quadro 4 – Componente Curricular de Arquitetura de Software

Objetivos	Empregar métodos e técnicas de análise e projeto no processo de desenvolvimento de sistemas de software orientado a objetos.
	Representar a arquitetura de software utilizando notações de modelagem.
Ementa	O componente curricular apresenta conceitos, evolução e importância da arquitetura de software. O componente curricular trabalha com análise e projeto no processo de desenvolvimento, padrões de arquitetura, padrões de distribuição, camadas no desenvolvimento de software, tipos de arquitetura de software, mapeamento de modelos, integração do sistema e estratégias de manutenção de software.
Bibliografia Básica	BEZERRA, Eduardo. Princípios de análise e projeto de sistemas com UML. 3.ed. Rio de Janeiro: Elsevier, 2015. ENGHOLM JÚNIOR, Hélio. Análise e design orientados a objetos. São Paulo: Novatec, 2013. SILVEIRA, Paulo et al. Introdução à arquitetura e design de software: uma visão sobre a plataforma Java. Rio de Janeiro: Campus, 2012..
Bibliografia Complementar	ERL, Thomas. SOA: princípios de design de serviços. São Paulo: Pearson Education do Brasil, 2009. GAMMA, Erich. et al. Padrões de projeto: soluções reutilizáveis de software orientado a objetos. Porto Alegre: Bookman, 2000. SOMMERVILLE, Ian. Engenharia de software. 10.ed. São Paulo: Pearson Education do Brasil, TERUEL, Evandro Carlos. Arquitetura de sistemas para web com Java utilizando design patterns e frameworks. Rio de Janeiro: Ciência Moderna, 2012. WAZLAWICK, Raul Sidnei. Análise e projeto de sistemas de informação orientados a objetos. 2.ed. Rio de Janeiro: Elsevier, 2011.

Fonte: Adaptado do PPC de ADS do Instituto Federal de São Paulo (2020, p. 127-128).