

Monitoria em Engenharia de Software: a relevância da abstração no ensino de modelagem e análise

Endrigo Gustavo Brandão de Oliveira

Faculdade de Tecnologia da Zona Leste (Fatec Zona Leste)
endrigo.oliveira@aluno.cps.sp.gov.br

Wilson Vendramel

Faculdade de Tecnologia da Zona Leste (Fatec Zona Leste)
wilson.vendramel@cps.sp.gov.br
ORCID 0000-0002-6643-0623

Resumo

A monitoria acadêmica costuma ser tratada como reforço de conteúdo, entretanto, este relato parte de outra hipótese: a de que ela também funciona como espaço de diagnóstico das dificuldades que os estudantes enfrentam, especificamente aqui, ao modelar e analisar software cujos processos requerem bastante a habilidade cognitiva da abstração. A experiência foi desenvolvida na disciplina de Engenharia de Software da Fatec Zona Leste no curso superior de Tecnologia em Análise e Desenvolvimento de Sistemas da Articulação da Formação Profissional Média e Superior, sob orientação do Prof. Wilson Vendramel. O trabalho combinou plantões presenciais e acompanhamento *online*, com ênfase no apoio individual a cada estudante. Ao final do semestre, as dificuldades recorrentes foram reunidas em categorias e confrontadas com a literatura. Os resultados sugerem que o acompanhamento individual, orientado por um referencial pedagógico adequado, ajuda o estudante a explicitar o próprio raciocínio, bem como auxilia o docente planejar melhor a disciplina.

Palavras-Chave

Monitoria; Ensino de engenharia de software; Modelagem e análise de software; Abstração; Taxonomia SOLO.

1. Introdução

Modelar e analisar software é, antes de tudo, decidir o que manter e o que deixar de fora. Essa competência atravessa o ciclo de vida do produto de software, no entanto, não é trivial ensinar nem aprender; a abstração acaba sendo ensinada de forma indireta, por meio de outros conteúdos, embora todos eles dependam dela para explicar, modelar e resolver problemas (KRAMER, 2007). Em modelagem e análise, catálogos de erros em UML apontam a interpretação textual e a abstração como dificuldades de fundo (CHOURIO et al., 2019). A monitoria favorece a aprendizagem mediada (FRISON, 2016), mas raramente é examinada como instrumento de diagnóstico dessas dificuldades.

Abordagens que tratam a capacidade cognitiva da abstração como objeto explícito de ensino ajudam a localizar onde o raciocínio falha (VENDRAMEL, 2023), mas são pouco exploradas formalmente nos ambientes de aprendizagem. Este relato descreve a monitoria conduzida em Engenharia

de Software com três objetivos: 1) caracterizar as dificuldades de abstração observadas; 2) apresentar estratégias de mediação adotadas e 3) discutir os resultados percebidos na turma.

2. Metodologia

A experiência ocorreu na Fatec Zona Leste no curso superior de Análise e Desenvolvimento de Sistemas da Articulação da Formação Profissional Média e Superior, durante o semestre de 2026/1, com carga horária de 4 horas semanais. A disciplina em questão foi a Engenharia de Software, ministrada pelo professor orientador, com 38 estudantes matriculados. O monitor cursa o 2º ano do mesmo curso e havia feito a mesma disciplina pouco tempo antes. Essa proximidade se mostrou relevante, visto que boa parte das dúvidas trazidas pelos colegas era a mesma que ele próprio tivera.

2.1 Referencial adotado

A observação se apoiou em dois referenciais fundamentais. O primeiro é o alinhamento construtivo (BIGGS; TANG, 2011), que correlaciona resultados pretendidos de aprendizagem, atividades de ensino e tarefas de avaliação. O segundo é a taxonomia SOLO (BIGGS; COLLIS, 1982; BIGGS; TANG, 2011), cujos verbos de aprendizagem serviram para reconhecer o nível cognitivo manifestado nas respostas durante os atendimentos, em outras palavras, uniestrutural, multiestrutural, relacional e abstrato estendido. A estes, soma-se a noção de padrões de abstração no ensino de engenharia de software (HAZZAN; KRAMER, 2006; VENDRAMEL, 2023), que direcionou o foco para o estudante compreender os processos de modelagem e análise, não somente a produção de artefatos de software, comumente exigidos em sala de aula.

2.2 Procedimentos

O trabalho se organizou em quatro frentes:

1. Plantões presenciais

Encontros semanais na unidade sobre levantamento de requisitos e diagramas de casos de uso, classes e sequência.

2. Acompanhamento *online*

Atendimento remoto por mensagem e videochamada, fora do horário de aula, que ampliou o alcance da monitoria para quem trabalhava durante o dia ou não conseguia permanecer na unidade.

3. Apoio individual nas atividades práticas

Revisão dos artefatos junto de cada estudante durante as aulas de laboratório, com devolutiva imediata.

4. Material de apoio e devolutiva ao docente

Exemplos resolvidos e roteiros de verificação de diagramas, validados pelo professor, e reuniões periódicas para realimentar o planejamento das aulas.

O apoio individual foi a frente mais eficiente. Em uma explicação dirigida à turma inteira, é preciso supor um ponto de partida comum a todos. No atendimento um a um, o monitor descobre onde o raciocínio de fato se interrompe, e dois estudantes que erram o mesmo diagrama raramente cometem o erro pelo mesmo motivo. Por isso, em vez de apontar o erro, o monitor passou a perguntar como o estudante havia chegado até ali. Cabe destacar que corrigir o artefato resolve a entrega da semana; explicitar o raciocínio que o produziu tende a resolver as próximas.

2.3 Sistematização das observações

Ao final do semestre, as dificuldades ao longo dos atendimentos foram agrupadas por similaridade em categorias, tomando como referência o catálogo de erros de Chourio et al. (2019); as de menor incidência formaram um conjunto residual. O desempenho da turma nas atividades de modelagem e análise de software foi obtido junto ao professor responsável, de forma agregada e sem identificação individual. Trata-se de relato descritivo, com análise qualitativa das dificuldades e tratamento quantitativo descritivo das incidências e das notas, sem grupo de comparação nem pretensão de inferência causal.

3. Resultados e discussão

Foram realizados 32 atendimentos ao longo do semestre entre plantões presenciais e acompanhamento remoto, envolvendo aproximadamente 30 estudantes distintos. A procura não se distribuiu de forma uniforme: concentrou-se nas semanas que antecedem as entregas e, em menor grau, logo após a introdução de cada novo tópico de conteúdo em aula. Como parte dos encontros tratou de mais de uma dúvida, os 32 atendimentos produziram 35 ocorrências, distribuídas na Tabela 1 por categoria.

Categoria de dificuldade	Ocorrências	Participação (%)
Omissão de notação visual relevante	9	25,7
Erros de abstração na regra de negócio	8	22,9
Confusão entre notações visuais	7	20,0
Confusão na relação entre diagramas	5	14,3
Outras ocorrências	6	17,1
Total	35	100,0

Tabela 1 - Incidência das dificuldades observadas nos atendimentos, por categoria

Fonte O autor, 2026

As quatro categorias recorrentes responderam por 82,9% das ocorrências. O conjunto residual reúne dúvidas isoladas, que não se repetiram o suficiente para formar categoria própria. Chama atenção que nenhuma das quatro decorre do desconhecimento de notação UML: em todos os casos o estudante dominava a notação visual, mas hesitava sobre o que representava com ela. A dificuldade notória é a de abstração e, consequentemente, de modelagem (lógica) e análise, não de diagramação propriamente dita.

3.1 Resultados

O Quadro 1 relaciona cada categoria à forma como se manifestou nos artefatos e à estratégia adotada durante o atendimento.

Dificuldade identificada	Manifestação no artefato	Estratégia de mediação
Omissão de notação visual relevante	Relacionamentos de classe sem cardinalidade nas extremidades.	Ler o relacionamento em linguagem natural nos dois sentidos.
Erros de abstração na regra de negócio	Regras de negócio ausentes do modelo ou dispersas em classes de fronteira.	Separar o que o sistema decide do que ele apenas exibe e alocar a regra na classe responsável.
Confusão entre notações visuais	Telas e componentes do sistema representados como atores.	Retomar a fronteira do sistema e perguntar quem permanece fora dela.
Confusão na relação entre diagramas	Elementos presentes em um diagrama sem correspondência nos demais.	Percorrer o mesmo elemento nos diferentes diagramas e verificar a correspondência.

Quadro 1 - Dificuldades recorrentes e estratégias de mediação adotadas
Fonte O autor, 2026

Mediante a consolidação e o entendimento das principais dúvidas obtidas no processo da monitoria, a abordagem individual, somada ao alinhamento com o professor, acompanhou o percurso retratado na Figura 1, que apresenta a distribuição das notas obtidas pelos 38 estudantes da turma nas avaliações de engenharia de software, cada qual composta por uma prova (60%) e um trabalho (40%), considerando que a segunda avaliação acumulou conteúdos e exigiu maior conhecimento em modelagem e análise de software.

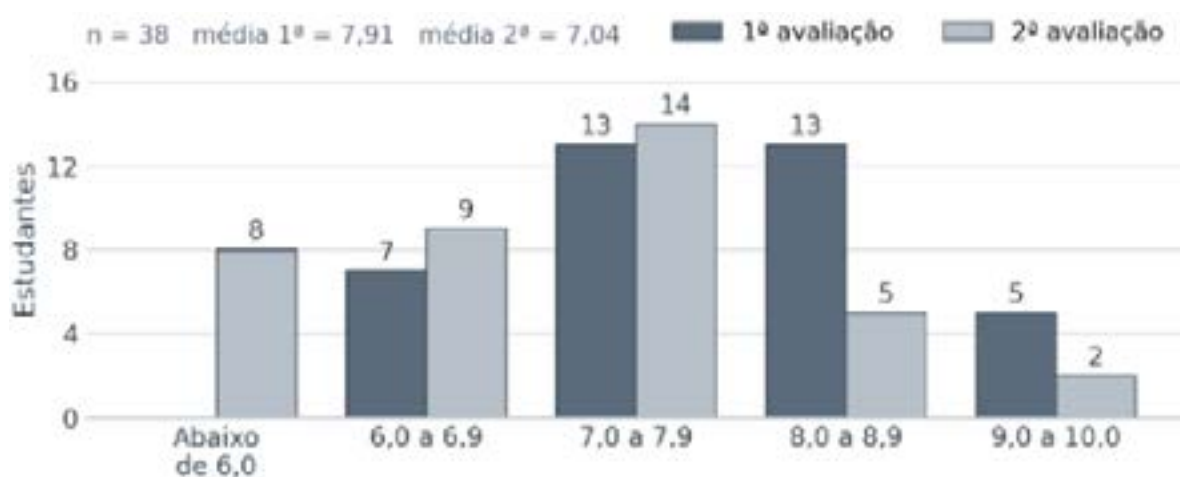


Figura 1 - Distribuição das notas dos estudantes na disciplina de Engenharia de Software
Fonte O autor, 2026

Na primeira avaliação, a distribuição se concentra nas faixas intermediárias e superiores, com média de 7,91 e uma taxa de 100% dos alunos com nota maior que 6,0. Na segunda, a média caiu para 7,04 e oito estudantes ficaram abaixo de 6,0, entretanto, 78,9% dos estudantes continuaram

com uma nota maior que 6,0. Mesmo havendo uma redução parcial de desempenho, oito estudantes (21,1%) seguiram a direção oposta, visto que elevaram a nota. Cabe salientar que houve mais conteúdos e maior complexidade de resolução de problemas na segunda avaliação.

3.2 Discussão

As categorias observadas convergem com o catálogo de Chourio et al. (2019), elaborado a partir da experiência com turmas de cursos da área da computação, o que indica que não se trata de um problema desta turma em particular. Tais autores também identificam a interpretação de texto e a abstração como habilidades de fundo para projetar e representar modelos, e propõem tratar o erro como instrumento mediador de ensino, que foi precisamente o uso dado a ele nos atendimentos.

Tratar a abstração como objeto explícito de observação ajudou justamente nesse ponto. Em vez de atribuir a dificuldade genericamente à falta de estudo, foi possível localizar em qual passagem entre níveis ela se instalava. O quadro 1 coincide com o que Kramer (2007) relata de sua experiência docente: estudantes capazes de raciocinar abstratamente que, ainda assim, não conseguem aplicar a abstração na construção de modelos, porque não compreendem o propósito e nível de abstração de cada modelo. A monitoria mostrou-se, nesse sentido, um espaço promissor para esse tipo de leitura, fora da aula habitual.

Há limitações evidentes. Um semestre, um monitor, uma observação sujeita à subjetividade de quem acompanha. Não houve instrumento padronizado de pré e pós-teste, de modo que os ganhos relatados pelos estudantes são autopercebidos. A categorização foi conduzida por uma única pessoa, sem verificação de concordância entre avaliadores.

4. Considerações finais

A monitoria descrita neste relato permitiu identificar e categorizar dificuldades recorrentes de abstração na construção de artefatos provenientes da modelagem e análise de software, associando a cada categoria uma estratégia de mediação aplicada no acompanhamento individual e apoiada em referencial pedagógico.

O que a experiência sugere é que a contribuição do monitor não se esgota no atendimento pontual à dúvida. Ao reconhecer as dificuldades que se repetem e devolvê-las ao docente, ele produz devolutivas significativas para melhorar o planejamento da disciplina. Para o próprio monitor, a exigência de reconstruir o raciocínio do colega, em vez de simplesmente apresentar a resposta correta, consolidou conceitos até então estudados de forma predominantemente operacional.

Como continuidade, pretende-se aplicar instrumentos padronizados de avaliação cognitiva, calibrados segundo a taxonomia SOLO, para verificar os efeitos das estratégias aqui descritas sobre o desempenho em modelagem e análise de software, objetivando os níveis cognitivos de aprendizado profundo, isto é, relacional e abstrato estendido.

5. Referências

BIGGS, John; COLLIS, Kevin. **Evaluating the quality of learning: the SOLO taxonomy**. New York: Academic Press, 1982.

BIGGS, John; TANG, Catherine. **Teaching for quality learning at university**. 4. ed. Maidenhead: Open University Press, 2011.

CHOURIO, Patricia; AZEVEDO, Romualdo; CASTRO, Alberto; GADELHA, Bruno. Most common errors in software modeling using UML. In: SIMPÓSIO BRASILEIRO DE ENGENHARIA DE SOFTWARE, 33., 2019, Salvador. **Anais [...]**. New York: ACM, 2019. p. 244-253.

FRISON, Lourdes Maria Bragagnolo. Monitoria: uma modalidade de ensino que potencializa a aprendizagem colaborativa e autorregulada. **Pro-Posições**, Campinas, v. 27, n. 1, p. 133-153, 2016.

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

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

VENDRAMEL, Wilson. **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. 2023. Tese (Doutorado em Tecnologias da Inteligência e Design Digital) – Pontifícia Universidade Católica de São Paulo, São Paulo, 2023.