

PRÁTICAS ÁGEIS DA ENGENHARIA DE SOFTWARE NO ENSINO MÉDIO: UM RELATO DE EXPERIÊNCIA

AGILE SOFTWARE ENGINEERING PRACTICES IN HIGH SCHOOL: AN EXPERIENCE REPORT

PRÁCTICAS ÁGILES DE LA INGENIERÍA DE SOFTWARE EN LA EDUCACIÓN SECUNDARIA: UN RELATO DE EXPERIENCIA

Kauê Wallace Coelho Olímpio¹, Vilmar dos Santos Cavalcante Filho², Ruan Carlos Pereira Barros³, Geina de Souza Soares⁴, Ione de Castro Matos da Silva⁵, Ana Stefanne Gama Moraes⁶, Ester Figueiredo Araújo⁷, João da Mata Libório Filho⁸

DOI: 10.54899/dcs.v23i94.6509

Recibido: 07/08/2026 | Aceptado: 11/08/2026 | Publicación en línea: 25/08/2026.

RESUMO

Este artigo apresenta a aplicação de uma sequência didática para o ensino de conceitos de Engenharia de *Software* no Ensino Médio, alinhada à habilidade EM13CO02 da BNCC Computação. A proposta aborda práticas ágeis, como histórias de usuário, refinamento de requisitos, estimativas, decomposição de tarefas, *sprints* e retrospectiva, por meio de uma atividade desplugada e colaborativa. A pesquisa caracteriza-se como quali-quantitativa, com delineamento quase experimental, envolvendo 20 estudantes do 2º ano do Ensino Médio, com aplicação de pré-teste e pós-teste. Os resultados quantitativos indicaram leve aumento descritivo da média de acertos, de 86,5% para 90,0%, sem diferença estatisticamente significativa. A análise qualitativa evidenciou indícios de apropriação de práticas de especificação, decomposição, refinamento e adequação da solução ao público-alvo.

Palavras-chave: Engenharia de Software. Educação Básica. Práticas Ágeis. Sequência Didática.

¹ Graduando em Computação, Universidade do Estado do Amazonas (UEA), Centro de Estudos Superiores de Itacoatiara (CESIT), Itacoatira, Amazonas, Brasil. E-mail: kwco.lic23@uea.edu.br

² Graduando em Computação, Universidade do Estado do Amazonas (UEA), Centro de Estudos Superiores de Itacoatiara (CESIT), Itacoatira, Amazonas, Brasil. E-mail: vdscl.lic25@uea.edu.br

Orcid: <https://orcid.org/0009-0001-0196-6355>

³ Graduando em Computação, Universidade do Estado do Amazonas (UEA), Centro de Estudos Superiores de Itacoatiara (CESIT), Itacoatira, Amazonas, Brasil. E-mail: rcpb.lic25@uea.edu.br

⁴ Graduando em em Computação, Universidade do Estado do Amazonas (UEA), Centro de Estudos Superiores de Itacoatiara (CESIT), Itacoatira, Amazonas, Brasil. E-mail: gdss.lic25@uea.edu.br

⁵ Especialista em Produção e Gestão de Mídias Digitais Educacionais, Universidade do Estado do Amazonas (UEA) – Centro de Estudos Superiores de Itacoatiara (CESIT), Trainee no Programa PROTLAB (CESIT), Itacoatiara, Amazonas, Brasil. E-mail: idcm.mbt26@uea.edu.br

⁶ Mestranda em Ensino de Computação, Universidade do Estado do Amazonas (UEA) – Centro de Estudos Superiores de Itacoatiara (CESIT), Itacoatiara, Amazonas, Brasil. E-mail: asgm.com26@uea.edu.br

⁷ Doutora em Ciências da Educação, Universidade do Minho, Braga, Distrito de Braga, Portugal.

E-mail: ester.araujo.1965@educacao.am.gov.br

⁸ Doutor em Informática, Universidade Federal do Amazonas (UFAM), Manaus, Amazonas, Brasil.

E-mail: jlfilho@uea.edu.br Orcid: <https://orcid.org/0000-0002-1987-6198>

ABSTRACT

This paper presents the application of an instructional sequence for teaching introductory Software Engineering concepts in high school, aligned with skill EM13CO02 of the Brazilian BNCC Computing framework. The proposal addresses agile practices, such as user stories, requirements refinement, estimation, task decomposition, sprints, and retrospectives, through an unplugged and collaborative activity. The study follows a qualitative-quantitative approach with a quasi-experimental design, involving 20 second-year high school students and the administration of pre- and post-tests. The quantitative results showed a slight descriptive increase in the mean score, from 86.5% to 90.0%, with no statistically significant difference. The qualitative analysis revealed evidence of students' appropriation of practices related to specification, decomposition, refinement, and adaptation of the solution to the target audience.

Keywords: Software Engineering. Basic Education. Agile Practices. Instructional Sequence.

RESUMEN

Este artículo presenta la aplicación de una secuencia didáctica para la enseñanza de conceptos de Ingeniería de Software en la Educación Secundaria, alineada con la competencia EM13CO02 de la BNCC Computación. La propuesta aborda prácticas ágiles, como historias de usuario, refinamiento de requisitos, estimaciones, descomposición de tareas, sprints y retrospectivas, mediante una actividad desconectada y colaborativa. La investigación se caracteriza por un enfoque cuali-cuantitativo, con un diseño cuasiexperimental, en el que participaron 20 estudiantes del segundo año de la Educación Secundaria, con la aplicación de un pretest y un postest. Los resultados cuantitativos indicaron un ligero aumento descriptivo en el promedio de respuestas correctas, del 86,5 % al 90,0 %, sin diferencias estadísticamente significativas. El análisis cualitativo evidenció indicios de apropiación de prácticas de especificación, descomposición, refinamiento y adecuación de la solución al público objetivo.

Palabras clave: Ingeniería de Software. Educación Básica. Prácticas Ágiles. Secuencia Didáctica.



Esta obra está bajo una [Licencia Creative Commons Atribución- NoComercial 4.0 Internacional](https://creativecommons.org/licenses/by-nc/4.0/)

INTRODUÇÃO

A inserção da Computação na Educação Básica tem sido discutida como uma necessidade formativa diante de uma sociedade cada vez mais mediada por tecnologias digitais, dados, sistemas computacionais e processos automatizados. Nesse contexto, aprender Computação não se limita ao uso instrumental de recursos tecnológicos, mas envolve compreender fundamentos, conceitos e formas de resolução de problemas próprias da área. Wing (2006) argumenta que o

pensamento computacional constitui uma habilidade fundamental para todos, pois envolve formular problemas e organizar soluções de modo que possam ser executadas por humanos ou máquinas. No contexto brasileiro, Ribeiro *et al.* (2019) defendem que os fundamentos da Computação sejam ensinados em profundidade compatível com cada etapa escolar, articulando os eixos de Pensamento Computacional, Mundo Digital e Cultura Digital.

No Brasil, o Ministério da Educação (2022) homologou o Complemento à Base Nacional Comum Curricular referente à Computação, fortalecendo a presença da área na Educação Básica, estabelecendo competências e habilidades específicas a serem desenvolvidas ao longo da escolarização. Esse documento organiza a Computação em três eixos estruturantes: Pensamento Computacional, Mundo Digital e Cultura Digital. No Ensino Médio, essas habilidades buscam ampliar a capacidade dos estudantes de compreender, representar, construir, analisar e comunicar soluções para problemas em diferentes contextos. De acordo com o Ministério da Educação (2022), entre essas habilidades, destaca-se a EM13CO02, que propõe: “Explorar e construir a solução de problemas por meio de refinamentos, utilizando diferentes níveis de abstração, desde a especificação até a implementação”.

Apesar desse avanço normativo, a implementação da Computação na Educação Básica ainda apresenta desafios pedagógicos, especialmente quando se busca trabalhar conceitos da área sem restringi-los à programação. Em muitas situações escolares, há limitações de infraestrutura, tempo didático, formação docente e disponibilidade de recursos computacionais. Nesse cenário, atividades desplugadas, colaborativas e contextualizadas podem constituir caminhos viáveis para introduzir conceitos computacionais de forma significativa. No caso da Engenharia de *Software*, práticas como levantamento de requisitos, escrita de histórias de usuário, refinamento, estimativa, decomposição de tarefas, construção incremental e retrospectiva podem ser adaptadas para contextos escolares, permitindo que os estudantes vivenciem processos de especificação, planejamento, implementação e revisão de soluções.

A problemática que orienta esta pesquisa pode ser formulada da seguinte maneira: de que modo uma sequência didática baseada em práticas ágeis e atividades desplugadas pode contribuir para o desenvolvimento da habilidade EM13CO02 em estudantes do Ensino Médio? Essa questão considera a necessidade de investigar não apenas a viabilidade da proposta em sala de aula, mas também as evidências de aprendizagem produzidas a partir de sua aplicação prática, especialmente no que se refere à capacidade dos estudantes de planejar, refinar, construir e revisar uma solução em equipe.

Diante disso, o objetivo deste artigo é analisar a aplicação de uma sequência didática para o ensino de conceitos de Engenharia de *Software* no Ensino Médio, baseada em práticas ágeis e alinhada à habilidade EM13CO02 da BNCC Computação. Especificamente, busca-se analisar o desempenho dos estudantes em questões objetivas aplicadas antes e após a intervenção, bem como discutir evidências qualitativas relacionadas à organização do processo de desenvolvimento, à especificação e decomposição da solução, ao refinamento incremental e à adequação da solução ao público-alvo.

Este estudo constitui uma extensão de uma proposta pedagógica apresentada anteriormente em formato de resumo expandido nos Anais do Simpósio Brasileiro de Computação na Educação Básica (SBC-EB). Na versão anterior, a contribuição concentrou-se na concepção e descrição da sequência didática, estruturada em etapas progressivas para o ensino de conceitos de Engenharia de *Software*. A originalidade deste artigo está na aplicação empírica da proposta em uma turma do 2º ano do Ensino Médio, com 20 estudantes, permitindo analisar evidências de aprendizagem em contexto real de sala de aula. Para isso, adotou-se uma abordagem quali-quantitativa, com delineamento quase experimental de grupo único, aplicação de pré-teste e pós-teste e análise qualitativa das respostas discursivas produzidas pelos estudantes.

Os resultados quantitativos indicaram leve aumento descritivo no desempenho dos estudantes após a intervenção. A média de acertos passou de 86,5% no pré-teste para 90,0% no pós-teste, enquanto a mediana passou de 90,0% para 100%. No entanto, a comparação inferencial não indicou diferença estatisticamente significativa entre os dois momentos avaliativos, sugerindo que a intervenção contribuiu mais para a manutenção e consolidação de um desempenho já elevado do que para um aumento expressivo nos percentuais de acerto. Por outro lado, a análise qualitativa das questões discursivas evidenciou indícios de apropriação de conceitos relacionados à Engenharia de *Software*, especialmente quanto à organização do processo de desenvolvimento, à decomposição de tarefas, ao refinamento da solução e à adequação ao público-alvo.

O artigo está organizado em cinco seções. Após esta introdução, a Seção 2 apresenta o referencial teórico. A Seção 3 descreve os procedimentos metodológicos. A Seção 4 discute os resultados quantitativos e qualitativos. Por fim, a Seção 5 apresenta as considerações finais, incluindo contribuições, limitações e possibilidades de continuidade do estudo.

REFERENCIAL TEÓRICO

Esta seção apresenta os fundamentos que sustentam a intervenção, articulando Computação na Educação Básica, BNCC Computação, Engenharia de *Software*, práticas ágeis e trabalhos relacionados.

COMPUTAÇÃO, BNCC E ENGENHARIA DE *SOFTWARE* NA EDUCAÇÃO BÁSICA

Lucas (2025) e Pacheco *et al.* (2025) afirmam que a inserção da Computação na Educação Básica tem sido compreendida como parte da formação necessária aos estudantes, especialmente por favorecer habilidades relacionadas à resolução de problemas, ao pensamento computacional e à construção de soluções em diferentes contextos. De acordo com Santos *et al.* (2023), no Ensino Médio, a BNCC Computação organiza essas aprendizagens em competências associadas aos eixos estruturantes da área, contemplando a compreensão, a representação, a análise e a construção de soluções computacionais. Entre essas habilidades, a EM13CO02 destaca a exploração e construção de soluções por meio de refinamentos e diferentes níveis de abstração, desde a especificação até a implementação.

Nesse contexto, a Engenharia de *Software* apresenta-se como uma possibilidade para o desenvolvimento da habilidade EM13CO02. Sommerville (2011) descreve a Engenharia de *Software* como uma disciplina que envolve processos como compreensão de problemas, levantamento de requisitos, planejamento, decomposição, implementação, testes e manutenção de soluções. Dessa forma, Santiago *et al.* (2023), Tonhão *et al.* (2021), Alves e Paulinho (2026), Rodrigues e Ribeiro (2024) destacam que, por sua natureza prática, a área pode ser trabalhada por meio de atividades colaborativas e contextualizadas, favorecendo a participação ativa dos estudantes, a organização do trabalho em equipe e a resolução de problemas. Assim, mais do que restringir o ensino de Computação à programação, a Engenharia de *Software* permite explorar processos de especificação, refinamento e construção de soluções em contextos acessíveis à Educação Básica.

PRÁTICAS ÁGEIS E APRENDIZAGEM COLABORATIVA

Kuz (2021) afirma que as práticas ágeis foram consolidadas como movimento em 2001, com a publicação do Manifesto Ágil, que sistematizou valores e princípios já presentes em diferentes métodos de desenvolvimento de *software*. Diante disso, Chagas *et al.* (2021) reforçam que essas abordagens valorizam colaboração, comunicação, adaptação a mudanças e entrega incremental. No contexto educacional, Kuz (2021) destaca que sua adaptação pode favorecer o trabalho colaborativo, a participação ativa dos estudantes e a reflexão sobre o processo de aprendizagem.

Neste trabalho, foram utilizadas práticas inspiradas em métodos ágeis para organizar a resolução do problema proposto. As histórias de usuário apoiaram a descrição da solução a partir da perspectiva do público-alvo, conforme Neo *et al.* (2024); o *Example Mapping* foi empregado para refinar requisitos e discutir critérios de aceitação, segundo Wynne (2024); o *Planning Poker* auxiliou a estimativa colaborativa do esforço das tarefas, de acordo com Naik e Divya (2022); e os eventos inspirados no *Scrum* organizaram o trabalho em ciclos iterativos de planejamento, execução, revisão e retrospectiva, conforme de Castro Lima *et al.* (2025). Dessa forma, as práticas ágeis foram adaptadas como estratégias pedagógicas para favorecer a colaboração e o refinamento progressivo da solução.

TRABALHOS RELACIONADOS

Monteiro *et al.* (2024) apresentam uma abordagem para o planejamento de aulas interdisciplinares com Pensamento Computacional na Educação Básica, enfatizando práticas colaborativas, contextualizadas e articuladas a diferentes áreas do conhecimento. Embora o estudo também esteja situado na Educação Básica, seu foco recai sobre o planejamento interdisciplinar com Pensamento Computacional, enquanto o presente trabalho investiga a aplicação e avaliação de uma sequência didática voltada a conceitos de Engenharia de *Software*, como especificação, refinamento, decomposição e revisão de soluções.

Alves e Paulino (2026) analisam metodologias ativas e ágeis em projetos integradores no Ensino Médio Integrado à Educação Profissional, em um curso técnico de Desenvolvimento de Sistemas. A experiência utiliza práticas como *Scrum*, *Kanban*, *sprints* e aprendizagem colaborativa para articular teoria e prática no desenvolvimento de soluções computacionais.

Diferentemente desse contexto técnico e voltado ao desenvolvimento de sistemas, a proposta aqui apresentada foi aplicada em uma turma regular do Ensino Médio, por meio de uma atividade desplugada e sem exigir conhecimento prévio de programação.

Souza *et al.* (2023) relatam uma experiência de ensino de Engenharia de Requisitos no Ensino Superior com apoio da ferramenta colaborativa REMO, envolvendo histórias de usuário, critérios de aceite, *sprints*, priorização e gerenciamento de requisitos. Apesar da proximidade com práticas ágeis e requisitos de *software*, o estudo diferencia-se do presente trabalho por estar situado na formação universitária e por utilizar uma ferramenta digital específica. Assim, o diferencial deste artigo está na adaptação e aplicação empírica desses conceitos no Ensino Médio regular, em uma abordagem desplugada, colaborativa, alinhada à habilidade EM13CO02 da BNCC Computação e acompanhada de análise quali-quantitativa da aprendizagem.

METODOLOGIA

Esta pesquisa caracteriza-se como um estudo de abordagem quali-quantitativa, com delineamento quase experimental de grupo único, baseado na aplicação de pré-teste e pós-teste. A escolha por esse delineamento justifica-se pelo fato de a intervenção ter sido realizada em uma turma previamente constituída, sem distribuição aleatória dos participantes e sem grupo controle. Participaram da atividade 20 estudantes do 2º ano do Ensino Médio de uma escola pública, os quais vivenciaram uma sequência didática voltada à introdução de conceitos de Engenharia de *Software* por meio de práticas ágeis e atividades desplugadas.

A intervenção foi desenvolvida ao longo de duas aulas, com duração de três horas cada, totalizando seis horas de atividades. A sequência didática foi organizada em etapas progressivas, articuladas a diferentes níveis de abstração no processo de construção de uma solução. A atividade consistiu na produção colaborativa de cartazes sobre agropecuária sustentável, compreendidos como artefatos concretos para a aplicação de práticas inspiradas na Engenharia de *Software* e em métodos ágeis. Os estudantes foram conduzidos desde a compreensão do problema e definição da visão da solução até sua construção, revisão e socialização. A Tabela 1 sintetiza a intervenção.

Tabela 1

Caracterização da intervenção pedagógica.

Etapas	Conceitos de ES	Atividade realizada
Apresentação do problema	Compreensão do problema e visão da solução	Discussão sobre o tema agropecuária sustentável, público-alvo e finalidade comunicacional da solução.
Histórias de usuário	Especificação inicial de requisitos	Formulação de histórias de usuário em linguagem simples, indicando o que a solução deveria comunicar e para quem seria produzida.
Example Mapping	Refinamento de requisitos e critérios de aceitação	Discussão de exemplos, regras, dúvidas e critérios para avaliar a clareza, adequação e coerência da solução.
Planning Poker	Estimativa colaborativa de esforço	Estimativa do esforço necessário para pesquisa, escrita, organização visual, ilustrações e revisão.
Decomposição de tarefas	Divisão do trabalho em ações executáveis	Transformação das histórias e requisitos em tarefas menores, com definição de responsabilidades no grupo.
Sprints	Construção incremental da solução	Produção dos cartazes em ciclos curtos, com ajustes sucessivos a partir dos critérios definidos.
Revisão e retrospectiva	Validação, melhoria e reflexão sobre o processo	Análise dos produtos, socialização dos resultados e discussão sobre planejamento, construção e revisão da solução.

Durante a intervenção, os estudantes trabalharam em grupos, simulando práticas colaborativas de desenvolvimento de soluções. A Figura 1 apresenta registros da aplicação da atividade, evidenciando a realização das *sprints* e a organização dos grupos em sala de aula.

Figura 1

Aplicação da intervenção pedagógica em sala de aula.



Para avaliar possíveis mudanças no desempenho dos estudantes, foi aplicado o mesmo instrumento antes e após a intervenção. O questionário foi composto por dez questões objetivas e duas questões abertas. As questões objetivas abordaram situações relacionadas ao planejamento de uma solução em equipe, definição de público-alvo, seleção e priorização de informações,

divisão de tarefas, revisão do produto e incorporação de melhorias a partir de *feedback*. As questões abertas buscaram captar, de forma qualitativa, como os estudantes compreendiam a organização inicial da solução e quais estratégias propunham para refiná-la quando sua mensagem não estivesse suficientemente compreensível.

Os dados quantitativos foram organizados em planilha eletrônica e analisados com auxílio do *software* Jamovi. Foram calculadas estatísticas descritivas, incluindo média, mediana, desvio-padrão, valores mínimo e máximo e intervalos de confiança. Em seguida, realizou-se a comparação entre pré-teste e pós-teste por meio do teste *t* para amostras emparelhadas, complementado pelo teste de Wilcoxon. A normalidade da diferença entre as medidas foi verificada pelo teste de Shapiro-Wilk, e a magnitude da diferença foi interpretada por meio do tamanho de efeito.

As questões abertas, nomeadas P1 e P2, foram analisadas por meio de análise de conteúdo categorial, considerando uma categoria predominante para cada resposta. Na P1, foram utilizadas as categorias: resposta insuficiente, organização genérica, distribuição de responsabilidades, planejamento da solução e especificação e decomposição da solução. Na P2, foram adotadas as categorias: resposta insuficiente, melhoria genérica da solução, ajuste de clareza comunicacional, refinamento com recursos de apoio e validação e adequação ao usuário. As categorias foram interpretadas de forma descritiva, priorizando a frequência das respostas e os deslocamentos qualitativos observados entre pré-teste e pós-teste.

Os materiais complementares desta pesquisa estão disponíveis no repositório Zenodo (DOI: <https://doi.org/10.5281/zenodo.20477319>) e incluem as planilhas com os dados tabulados, o relatório estatístico gerado no Jamovi, o questionário aplicado e exemplos dos artefatos produzidos. A pesquisa foi autorizada pela Coordenadoria Regional de Educação, pela direção da escola e pelo professor regente da turma. Os estudantes responderam ao termo de assentimento e foram informados sobre os objetivos da atividade. Como limitação ética e metodológica, registra-se que a pesquisa não foi submetida à apreciação de Comitê de Ética em Pesquisa, devendo seus resultados ser interpretados como evidências de uma intervenção pedagógica aplicada em contexto escolar, com autorização institucional.

RESULTADOS E DISCUSSÕES

Esta seção apresenta os resultados do pré-teste e do pós-teste aplicados aos estudantes participantes da sequência didática. O instrumento avaliativo foi composto por dez questões objetivas e duas questões discursivas, relacionadas a conceitos introdutórios de Engenharia de Software, como planejamento, definição de público-alvo, priorização de informações, divisão de tarefas, revisão e melhoria de soluções.

ANÁLISE DAS QUESTÕES OBJETIVAS

A Tabela 2 apresenta as estatísticas descritivas dos percentuais de acerto no pré-teste, no pós-teste e do ganho obtido pelos estudantes. Observa-se aumento da média de acertos de 86,5% para 90,0%, com ganho médio de 3,5 pontos percentuais. A mediana passou de 90,0% para 100,0%, indicando maior concentração de estudantes em faixas elevadas de desempenho no pós-teste. No entanto, a mediana do ganho foi igual a 0,0, sugerindo estabilidade no desempenho de parte da turma.

Tabela 2

Estatística descritiva dos resultados do pré-teste, pós-teste e ganho.

Indicador	Pré-teste (%)	Pós-teste (%)	Ganho (%)
N	20	20	20
Média	86,5	90,0	3,5
IC 95% - limite inferior	79,5	83,4	-5,89
IC 95% - limite superior	93,5	96,6	12,9
Mediana	90,0	100,0	0,0
Desvio-padrão	15,0	14,1	20,1
Mínimo	40,0	50,0	-30,0
Máximo	100,0	100,0	100,0

Antes da comparação inferencial, a normalidade da diferença entre pós-teste e pré-teste foi verificada pelo teste de Shapiro-Wilk, que não indicou violação do pressuposto ($W = 0,920$, $p = 0,098$). Assim, utilizou-se o teste t para amostras emparelhadas, complementado pelo teste de Wilcoxon. Os resultados são apresentados na Tabela 3.

Tabela 3*Comparação entre pré-teste e pós-teste.*

Teste	Estatística	gl	<i>p</i>	Efeito	IC 95% do efeito
<i>t</i> pareado	0,780	19	0,445	$d = 0,174$	[-0,270; 0,614]
Wilcoxon	69,0	-	0,623	$r_{rb} = 0,150$	-

Os testes inferenciais não indicaram diferença estatisticamente significativa entre pré-teste e pós-teste, tanto pelo teste *t* pareado, $t(19) = 0,780$, $p = 0,445$, quanto pelo teste de Wilcoxon, $W = 69,0$, $p = 0,623$. Os tamanhos de efeito também foram pequenos ($d = 0,174$; $r_{rb} = 0,150$), indicando que a diferença observada foi limitada. Assim, embora tenha ocorrido aumento descritivo da média e da mediana, os dados quantitativos não permitem afirmar que houve melhora estatisticamente significativa no desempenho geral da turma. Uma possível explicação é que o instrumento objetivo apresentava situações generalizadas de tomada de decisão, nas quais os estudantes poderiam inferir a alternativa mais adequada mesmo sem conhecer formalmente práticas ágeis ou conceitos específicos de Engenharia de *Software*. Além disso, o desempenho inicial elevado reduziu a margem para crescimento no pós-teste, podendo ter produzido efeito teto e limitado a sensibilidade do instrumento para captar avanços decorrentes da intervenção.

ANÁLISE QUALITATIVA DAS QUESTÕES DISCURSIVAS

A análise das questões discursivas permitiu observar como os estudantes descreveram o processo de construção e melhoria de uma solução antes e após a intervenção. A Tabela 4 apresenta a distribuição das respostas por categoria, considerando a P1 como relacionada à organização, especificação e planejamento inicial da solução, e a P2 como relacionada ao refinamento, validação e adequação da solução ao público-alvo. Para fins de apresentação, optou-se por uma tabela-síntese, enquanto as respostas anonimizadas e suas respectivas classificações encontram-se disponíveis nos materiais complementares.

Tabela 4

Distribuição das categorias nas respostas abertas com foco em Engenharia de Software.

Categoria	Pré-teste	Pós-teste
P1 -- Organização, especificação e planejamento da solução		
Resposta insuficiente	3 (15%)	2 (10%)
Organização genérica	4 (20%)	3 (15%)
Distribuição de responsabilidades	2 (10%)	4 (20%)
Planejamento da solução	9 (45%)	6 (30%)
Especificação e decomposição da solução	2 (10%)	5 (25%)
P2 -- Refinamento, validação e adequação da solução		
Resposta insuficiente	2 (10%)	2 (10%)
Melhoria genérica da solução	4 (20%)	4 (20%)
Ajuste de clareza comunicacional	7 (35%)	5 (25%)
Refinamento com recursos de apoio	7 (35%)	7 (35%)
Validação e adequação ao usuário	0 (0%)	2 (10%)

Na P1, a categoria mais frequente no pré-teste foi “planejamento da solução”, com 9 estudantes (45%), reunindo respostas que mencionavam pesquisa, seleção de informações, organização de tópicos ou estruturação prévia do trabalho. No pós-teste, essa categoria passou para 6 estudantes (30%), enquanto aumentaram as respostas classificadas como “distribuição de responsabilidades”, de 2 (10%) para 4 estudantes (20%), e “especificação e decomposição da solução”, de 2 (10%) para 5 estudantes (25%). Considerando a ordenação das categorias do menor para o maior nível de elaboração, 7 estudantes avançaram para categorias mais elaboradas, 10 permaneceram na mesma categoria e 3 foram classificados em categorias inferiores. Esses dados sugerem melhora qualitativa na forma como parte da turma passou a descrever a passagem de uma ideia inicial para ações concretas de implementação.

Na P2, as categorias “ajuste de clareza comunicacional” e “refinamento com recursos de apoio” concentraram a maior parte das respostas. No pré-teste, ambas reuniram 7 estudantes cada (35%); no pós-teste, “refinamento com recursos de apoio” permaneceu com 7 estudantes (35%), enquanto “ajuste de clareza comunicacional” passou para 5 estudantes (25%). O principal avanço foi o surgimento da categoria “validação e adequação ao usuário”, ausente no pré-teste e identificada em 2 estudantes (10%) no pós-teste. Essa categoria indica respostas mais próximas de práticas de Engenharia de *Software*, pois envolve revisar a solução, testar sua compreensão, reorganizar informações e considerar a perspectiva do público-alvo. Na movimentação ordinal, 6 estudantes apresentaram respostas mais elaboradas, 9 permaneceram na mesma categoria e 5 foram classificados em categorias inferiores.

Em conjunto, os resultados quantitativos e qualitativos indicam que a intervenção contribuiu mais para a consolidação e aplicação prática de conceitos do que para um aumento

estatisticamente significativo nos percentuais de acerto. Enquanto os dados objetivos apontam manutenção de um desempenho já elevado, a análise discursiva evidencia indícios de apropriação de práticas relacionadas à Engenharia de *Software*, especialmente quanto à organização do processo de desenvolvimento, decomposição de tarefas, refinamento incremental e adequação da solução ao público-alvo. Assim, o cartaz deve ser compreendido não como foco central da avaliação, mas como artefato concreto para vivenciar o percurso entre especificação, construção, revisão e melhoria de uma solução.

CONCLUSÃO

Este artigo apresentou a aplicação de uma sequência didática para o ensino de conceitos introdutórios de Engenharia de *Software* no Ensino Médio, fundamentada em práticas ágeis, atividades desplugadas e alinhada à habilidade EM13CO02 da BNCC Computação. A proposta partiu da compreensão de que a Computação na Educação Básica não deve se restringir ao uso instrumental de tecnologias ou à programação, mas pode envolver processos de resolução de problemas, representação, especificação, refinamento e construção de soluções, conforme discutido por Lucas (2025), Pacheco *et al.* (2025) e Santos *et al.* (2023). Nesse sentido, os resultados reforçam a possibilidade de explorar a Engenharia de *Software* como objeto de ensino na Educação Básica, aproximando os estudantes de práticas como planejamento, decomposição, revisão e melhoria de soluções, em consonância com a compreensão da área como processo sistemático de desenvolvimento, conforme descrito por Sommerville (2011), e com abordagens práticas e colaborativas de ensino, destacadas por Santiago *et al.* (2023) e Tonhão *et al.* (2021).

Os resultados quantitativos indicaram leve melhora descritiva no desempenho dos estudantes, com aumento da média de acertos de 86,5% para 90,0% e da mediana de 90,0% para 100,0%. Entretanto, essa diferença não foi estatisticamente significativa, conforme indicado pelo teste *t* pareado, $t(19) = 0,780$, $p = 0,445$, e pelo teste de Wilcoxon, $W = 69,0$, $p = 0,623$. Esse resultado não permite afirmar que houve melhora significativa no desempenho objetivo geral da turma. Uma possível explicação é que o instrumento avaliativo apresentava situações relativamente generalizadas, nas quais os estudantes podiam inferir a alternativa mais adequada mesmo sem domínio formal de práticas ágeis ou de conceitos específicos de Engenharia de *Software*. Além disso, o desempenho inicial elevado pode ter produzido efeito teto, reduzindo a sensibilidade do teste para captar avanços decorrentes da intervenção.

Por outro lado, a análise qualitativa das questões discursivas evidenciou indícios de apropriação de conceitos trabalhados na sequência didática. Na questão relacionada à organização inicial da solução, observou-se aumento nas respostas associadas à distribuição de responsabilidades e à especificação e decomposição da solução. Na questão voltada ao refinamento, surgiu no pós-teste a categoria validação e adequação ao usuário, ausente no pré-teste. Esses achados confirmam parcialmente a literatura que aponta o potencial de práticas ágeis para favorecer colaboração, comunicação, adaptação e reflexão sobre o processo de aprendizagem, segundo Kuz (2021) e Chagas *et al.* (2021). Também dialogam com Alves e Paulino (2026), ao evidenciar a contribuição de metodologias ativas e ágeis para organização e colaboração, e com Souza *et al.* (2023), ao aproximar estudantes de práticas relacionadas a requisitos, histórias de usuário, critérios e refinamento. Diferentemente desses estudos, porém, a presente proposta foi aplicada em uma turma regular do Ensino Médio, em formato desplugado e sem exigir conhecimentos prévios de programação.

Assim, conclui-se que a sequência didática mostrou-se adequada como experiência introdutória para aproximar estudantes do Ensino Médio de práticas de Engenharia de *Software*. A principal contribuição observada não foi um ganho estatisticamente significativo nos percentuais de acerto, mas a possibilidade de vivenciar, em contexto escolar, um processo de construção de solução envolvendo compreensão do problema, planejamento, especificação, decomposição de tarefas, refinamento incremental, revisão e adequação ao público-alvo. Nesse aspecto, o estudo amplia discussões anteriores sobre Pensamento Computacional e interdisciplinaridade na Educação Básica, ao focalizar especificamente práticas de Engenharia de *Software* associadas à habilidade EM13CO02 da BNCC Computação.

Como limitações, destacam-se o número reduzido de participantes, o delineamento de grupo único sem grupo controle, a curta duração da intervenção e a possível baixa sensibilidade do instrumento objetivo. Para estudos futuros, recomenda-se refinar o instrumento de avaliação, incorporando questões mais sensíveis a aspectos técnicos das metodologias ágeis e da Engenharia de *Software*, como histórias de usuário, critérios de aceitação, priorização, estimativa, decomposição e retrospectiva. Também se sugere aplicar a sequência didática em um período maior, permitindo explorar cada etapa com mais detalhamento e aprofundamento, além de incorporar outras fontes de evidência, como observação das interações em grupo, análise dos artefatos produzidos, rubricas avaliativas e registros reflexivos dos estudantes.

AGRADECIMENTOS

Os autores agradecem à Universidade do Estado do Amazonas (UEA) pelo apoio institucional, por meio do Programa de Gratificação de Produtividade Acadêmica (GPA), Portaria nº 1177/2025-GR/UEA. A pesquisa contou com a atuação de bolsistas vinculados aos programas de Iniciação Científica e Tecnológica, apoiados pela FAPEAM e pelo CNPq, no âmbito do Edital nº 031/2025-GR/UEA, e ao Programa de Apoio ao Desenvolvimento da Extensão Universitária (PADEX), conforme edital nº 15/2026.

REFERÊNCIAS

- Alves, M. V. P. and Paulino, O. F. (2026). Metodologias ativas e ágeis em projetos integradores: Experiência com discentes da educação profissional. *Revista Brasileira de Informática na Educação*, 34:497–515.
- Brasil. Ministério da Educação (2022). Computação: Complemento à base nacional comum curricular. Ministério da Educação. Acesso em: 21 maio 2026
- Chagas, R. A., Ribeiro, N. M., and Costa, T. D. (2021). Trilhas de aprendizagem, metodologias ativas e ágeis para o autodesenvolvimento. *Olhares & Trilhas*, 23(3):1215–1234.
- de Castro Lima, A., Jucá, S. C. S., da Silva, S. A., Lemos, P. B. S., and Lima, C. R. C. (2025). Ensino ágil na engenharia: Revisão sobre o uso do scrum no desenvolvimento de competências do século xxi. *Revista de Ensino de Engenharia*, 44:394–412.
- Kuz, A. (2021). Scrum: A new framework applied to education. *Revista Eduweb*, 15(3):10–17.
- Lucas, N. C. (2025). Da teoria à prática: propostas pedagógicas para a implementação da bncc computação no currículo escolar.
- Monteiro, L., Falcão, T., and Rodrigues, R. (2024). Uma abordagem para planejamento de aulas interdisciplinares com pensamento computacional para educação básica. In *Anais do IV Simpósio Brasileiro de Educação em Computação*, pages 168–176, Porto Alegre, RS, Brasil. SBC.
- Naik, N. G. and Divya, T. (2022). Planning poker tool for story point estimation. *RV Journal of Science Technology Engineering Arts and Management*, 3:82–92.
- Neo, G. d. S., Moura, J. A. B., Almeida, H., Neo, A. V. B. d. S., and Freitas Júnior, O.d. G. (2024). User story tutor (ust) to support agile software developers. *arXiv preprint arXiv:2406.16259*.
- Pacheco, A. M. B. *et al.* (2025). Desafios e possibilidades para a inserção do ensino da computação na educação básica.

- Ribeiro, L., Foss, L., Cavalheiro, S. A. d. C., Bordini, A., Frango, I., Reiser, R., Marques, M., Costa, E., Fortes, S., and Jaques, P. (2019). Diretrizes da sociedade brasileira de computação para o ensino de computação na educação básica. Technical report, Sociedade Brasileira de Computação, Porto Alegre.
- Rodrigues, G. and Ribeiro, M. (2024). Aprendizagem baseada em projeto: Um relato de experiência na disciplina de engenharia de software. In *Anais do Seminário de Ensino, Pesquisa e Extensão da UEG*. Universidade Estadual de Goiás.
- Santiago, C. P., Mendonça Menezes, J. W., and Alves de Aquino, F. J. (2023). Proposta e avaliação de uma metodologia de aprendizagem baseada em projetos em disciplinas de engenharia de software através de uma sequência didática. *Revista Brasileira de Informática na Educação*, 31:31–59.
- Santos, A. C., Nascimento, I., and Oliveira, W. (2023). Da bncc à bncc computação: Histórico, afinidades e desafios na implementação de um currículo Único. In *Anais Estendidos do III Simpósio Brasileiro de Educação em Computação*, pages 52–53, Porto Alegre, RS, Brasil. SBC.
- Sommerville, I. (2011). *Engenharia de Software*. Pearson Prentice Hall, São Paulo, 9 edition.
- Souza, A., Souza, F., Ortoncelli, A., and Costa, L. (2023). Ensino de engenharia de requisitos com apoio de uma ferramenta baseada em metodologia ágeis: Um relato de experiência. In *Anais do XXIX Workshop de Informática na Escola*, pages 682–691, Porto Alegre, RS, Brasil. SBC.
- Tonhão, S., Medeiros, A., and Prates, J. (2021). Uma abordagem prática apoiada pela aprendizagem baseada em projetos e gamificação para o ensino de engenharia de software. In *Anais do Simpósio Brasileiro de Educação em Computação*, pages 143–151, Porto Alegre, RS, Brasil. SBC.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3):33–35.
- Wynne, M. (2024). Example mapping: Slice any story into testable examples. Acesso em: 29 maio 2026.