

APRENDIZAGEM ATIVA NO ENSINO DE ENGENHARIA DE SOFTWARE: RELATO DE EXPERIÊNCIA DA INTEGRAÇÃO ENTRE PROBLEM-BASED LEARNING (PBL) E O FRAMEWORK SCRUM NA EDUCAÇÃO SUPERIOR TECNOLÓGICA

FELIPE EDUARDO DAHER PEREIRA¹

¹Especialista em Matemática e Física pela FACUMINAS, Coordenador e Professor no Centro Universitário UNIFACIG. E-mail: felipe.pereira@sempre.UNIFACIG.edu.br

RESUMO

O ensino de Engenharia de *Software* no ensino superior tecnológico exige estratégias que aproximem os conteúdos teóricos das situações encontradas no mercado de trabalho. Nesse sentido, a utilização de metodologias ativas pode favorecer tanto a aprendizagem dos conceitos técnicos quanto o desenvolvimento de habilidades relacionadas à colaboração e à resolução de problemas. Este artigo apresenta um relato de experiência docente sobre a aplicação conjunta da Aprendizagem Baseada em Problemas (*Problem-Based Learning* - PBL) e do framework ágil Scrum no conteúdo de Engenharia e Qualidade de *Software*, ministrado no 3º período do curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas (ADS) do Centro Universitário UNIFACIG. A atividade foi desenvolvida em cinco etapas. Inicialmente, cada equipe recebeu uma situação-problema distinta, baseada em demandas do mercado de desenvolvimento de *Software*, a ser desenvolvida ao longo de aproximadamente quinze dias. Em seguida, foram organizadas as equipes e definidos os papéis do Scrum: *Product Owner*, *Scrum Master* e *Developers*. Na sequência, cada grupo elaborou o *Product Backlog*, organizando as demandas em histórias de usuário, critérios de aceite e estimativas. Posteriormente, foram simulados ciclos de *Sprints*, permitindo acompanhar as etapas de planejamento, execução e acompanhamento das atividades. Ao final, as equipes apresentaram os resultados por meio de *storytelling*, compartilhando o produto desenvolvido e as experiências vivenciadas durante o processo. Participaram 19 estudantes, organizados em seis equipes, que utilizaram as ferramentas Trello ou Jira no acompanhamento das tarefas. Durante a atividade, observou-se maior participação dos estudantes e melhor compreensão dos conceitos e artefatos relacionados à Engenharia de *Software* e ao Scrum. Também foram trabalhadas competências como comunicação, organização, negociação de prioridades e trabalho em equipe. As principais dificuldades envolveram a familiaridade inicial com as práticas ágeis e o controle do tempo das cerimônias, aspectos acompanhados e ajustados pela mediação docente. A experiência indica que a integração entre PBL e Scrum pode aproximar o ensino de Engenharia de *Software* da realidade profissional, associando conhecimentos técnicos à prática colaborativa e à resolução de problemas. Como contribuição adicional, o artigo apresenta um conjunto de instrumentos de avaliação, um questionário de autopercepção e uma rubrica de análise de artefatos, proposto para aplicações futuras da abordagem.

Palavras-chave: Metodologias ativas; Aprendizagem baseada em problemas; Scrum; Engenharia de *Software*; Ensino superior.

ACTIVE LEARNING IN SOFTWARE ENGINEERING EDUCATION: AN EXPERIENCE REPORT ON THE INTEGRATION OF PROBLEM-BASED LEARNING (PBL) AND THE SCRUM FRAMEWORK IN HIGHER TECHNOLOGICAL EDUCATION

ABSTRACT

Software Engineering education in higher education requires strategies that bring theoretical content closer to situations encountered in the professional environment. In this context, active learning methodologies can support both the understanding of technical concepts and the development of skills related to collaboration and problem-solving. This article presents a teaching experience involving the combined use of Problem-Based Learning (PBL) and the Scrum agile framework in the Software Engineering and Quality course, offered to third-semester students of the Technologist Degree in

Systems Analysis and Development (ADS) at Centro Universitário UNIFACIG. The activity was developed in five stages. First, each team was assigned a distinct problem scenario based on demands from the Software development market, to be developed over approximately fifteen days. Teams were then organized and Scrum roles were assigned: Product Owner, Scrum Master, and Developers. Next, each team developed a Product Backlog, organizing requirements into user stories, acceptance criteria, and estimates. The students then simulated Sprint cycles, following the planning, execution, and monitoring of activities. At the end, the teams presented their results through storytelling, demonstrating the product developed and sharing their experiences throughout the process. During the activity, greater student participation and a better understanding of Software Engineering and Scrum concepts and artifacts were observed. Skills such as communication, organization, priority negotiation, and teamwork were also developed. The main challenges involved students' initial familiarity with agile practices and time management during Scrum ceremonies, which were addressed through continuous teacher guidance. The experience indicates that integrating PBL and Scrum can bring Software Engineering education closer to professional practice by combining technical knowledge with collaborative work and problem-solving.

Keywords: Active learning methodologies; Problem-based learning; Scrum; Software engineering; Higher education.

1 INTRODUÇÃO

A formação de profissionais na área de Computação e Tecnologia da Informação passa por mudanças constantes, acompanhando a rápida evolução das ferramentas, práticas e formas de trabalho utilizadas no desenvolvimento de *Software*. Esse cenário também traz novos desafios para o ensino superior, especialmente na preparação de estudantes para situações que vão além do domínio de linguagens de programação e conceitos técnicos. No ambiente profissional, é necessário saber analisar problemas, trabalhar em equipe, lidar com mudanças de requisitos, tomar decisões e buscar soluções que atendam às necessidades dos usuários e das organizações.

Nesse contexto, as metodologias de aprendizagem ativa têm ganhado espaço no ensino superior por favorecerem uma participação mais efetiva dos estudantes no processo de aprendizagem (Berbel, 2011), sendo apontadas na literatura recente como uma via para reduzir os desafios historicamente associados ao ensino de Engenharia de *Software* (Lima *et al.*, 2020). Diferentemente de uma dinâmica baseada apenas na exposição de conteúdos, essas abordagens permitem que o estudante tenha contato com situações que exigem investigação, tomada de decisão e aplicação dos conhecimentos trabalhados em sala de aula. Essa perspectiva é especialmente relevante nos cursos de graduação tecnológica, como o Superior de Tecnologia em Análise e Desenvolvimento de Sistemas (ADS), nos quais a formação busca estabelecer uma relação próxima entre os conhecimentos acadêmicos e as demandas da atuação profissional.

Entre as metodologias que podem contribuir para esse processo está a Aprendizagem Baseada em Problemas (*Problem-Based Learning* - PBL) (Barrows, 1986). A proposta parte de situações-problema que desafiam os estudantes a buscar informações, discutir possibilidades e

construir soluções de forma colaborativa. Na área de Engenharia de *Software*, essa abordagem pode ser associada a práticas utilizadas no desenvolvimento de projetos, criando oportunidades para que os estudantes compreendam os conteúdos a partir de sua aplicação em situações próximas às encontradas no mercado.

Nesse sentido, o framework Scrum apresenta características que podem complementar a dinâmica proposta pela PBL. Por meio de seus papéis, eventos e artefatos, o Scrum organiza o trabalho em ciclos curtos e favorece o acompanhamento contínuo das atividades, a colaboração entre os integrantes e a revisão das entregas (Schwaber; Sutherland, 2020). Embora tenha sido concebido para o desenvolvimento de produtos, seus princípios e práticas podem ser utilizados em atividades educacionais que envolvam projetos, resolução de problemas e trabalho em equipe. Experiências anteriores já relataram a integração entre métodos ágeis e a Aprendizagem Baseada em Problemas no ensino de Engenharia de *Software*, evidenciando ganhos na aproximação entre os conteúdos teóricos e a prática profissional (Meyreles; Bonifácio, 2015).

A aproximação entre PBL e Scrum, portanto, permite trabalhar dois aspectos importantes na formação em Engenharia de *Software*: a resolução de problemas e a organização do trabalho necessário para construir uma solução. Enquanto a situação-problema orienta a investigação e a aprendizagem, a dinâmica do Scrum oferece uma estrutura para que os estudantes planejem, executem e acompanhem suas atividades. Essa combinação também cria oportunidades para o desenvolvimento de competências como comunicação, colaboração, organização e negociação de prioridades.

Diante desse contexto, este artigo tem como objetivo descrever e analisar a experiência de aplicação integrada da Aprendizagem Baseada em Problemas e do framework Scrum no conteúdo de Engenharia e Qualidade de *Software*, ministrado para uma turma do 3º período do curso de Análise e Desenvolvimento de Sistemas do Centro Universitário UNIFACIG. O trabalho apresenta as etapas utilizadas na atividade, desde a definição da situação-problema e organização das equipes até a execução das *Sprints* e a apresentação final por meio de *storytelling*. Também são discutidas as percepções observadas durante a experiência, considerando a participação dos estudantes, a compreensão dos conceitos de Engenharia de *Software* e o desenvolvimento de competências relacionadas ao trabalho em equipe, além das dificuldades encontradas durante a condução da atividade.

2 MATERIAIS E MÉTODOS

Este trabalho consiste em um relato de experiência de natureza qualitativa e descritiva, elaborado a partir da observação direta do docente responsável pelo conteúdo de Engenharia e

Qualidade de *Software*, ministrado no 3º período do curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas (ADS) do Centro Universitário UNIFACIG, no primeiro semestre letivo do ano de 2026 (2026/1), com uma turma composta por 19 estudantes.

Tabela 1 – Caracterização da turma e organização da atividade

Aspecto	Descrição
Conteúdo	Engenharia e Qualidade de <i>Software</i>
Curso e período	Superior de Tecnologia em Análise e Desenvolvimento de Sistemas (ADS) — 3º período
Instituição e semestre	Centro Universitário UNIFACIG — 2026/1
Participantes	19 estudantes
Organização das equipes	Seis equipes de até quatro integrantes: quatro com quatro estudantes, uma dupla e um estudante em condução individual
Abordagens integradas	Aprendizagem Baseada em Problemas (PBL) e adaptação pedagógica do <i>framework Scrum</i>
Ferramentas de apoio	Trello ou Jira, à escolha de cada equipe

Fonte: elaborado pelo autor (2026).

A atividade foi planejada e conduzida em cinco etapas sequenciais. Na primeira etapa, os estudantes receberam uma situação-problema baseada em demandas reais do mercado de desenvolvimento de *Software*, seguindo os princípios da Aprendizagem Baseada em Problemas (BARROWS, 1986). Na segunda etapa, as equipes foram organizadas e os papéis do Scrum foram distribuídos entre os integrantes: *Product Owner*, *Scrum Master* e *Developers*. A turma foi dividida em seis equipes de até quatro integrantes, sendo quatro equipes compostas por quatro estudantes, uma dupla e um estudante que conduziu a atividade individualmente. Nesses dois últimos casos, os papéis do Scrum foram acumulados pelos participantes, condição cujos desdobramentos são retomados na seção de resultados. Na terceira etapa, cada equipe elaborou o *Product Backlog* da situação-problema recebida, organizando as demandas em histórias de usuário, critérios de aceite e estimativas de esforço. Na quarta etapa, foram simulados ciclos de *Sprints*, com sessões de planejamento, execução e revisão das atividades, mediadas pelo docente. As equipes foram orientadas a organizar a simulação do desenvolvimento ao longo de aproximadamente quinze dias, cabendo a cada uma definir a quantidade e a duração de suas próprias *Sprints* conforme a estratégia adotada para conduzir o projeto. A proposta era que os ciclos simulassem progressivamente a construção da solução, contemplando planejamento, execução, revisão e ajustes até a apresentação final. Cabe destacar que se tratou de uma

adaptação pedagógica de elementos do Scrum ao contexto de sala de aula, e não da aplicação integral do framework tal como praticado em ambiente profissional. Na quinta e última etapa, cada equipe apresentou seus resultados por meio de *storytelling*, compartilhando o produto desenvolvido e as experiências vivenciadas ao longo do processo. Para apoio à execução das etapas, foi indicado às equipes o uso das ferramentas de gestão ágil Trello e Jira, à escolha de cada grupo, empregadas na criação, no detalhamento e na movimentação dos cards correspondentes às histórias de usuário e às tarefas de cada *Sprint*, bem como no apoio à organização da apresentação final.

Cada equipe recebeu uma situação-problema distinta, formulada a partir de contextos organizacionais de funcionamento conhecido pelos estudantes, de modo que a investigação pudesse concentrar-se na modelagem da solução e não na compreensão do domínio. As situações atribuídas e os produtos correspondentes são apresentados na Tabela 2.

Tabela 2 – Situações-problema atribuídas às equipes

Equipe	Situação-problema	Produto a ser desenvolvido
1	Gestão de pedidos de uma lanchonete	Sistema de pedidos
2	Agendamento de consultas em uma clínica	Sistema de agendamento
3	Controle de empréstimos de uma biblioteca	Sistema de biblioteca
4	Gestão de serviços de uma oficina mecânica	Sistema de oficina
5 (dupla)	Gestão de chamados de suporte técnico	Sistema de chamados
6 (individual)	Gestão de reservas de salas e recursos	Sistema de reservas

Fonte: elaborado pelo autor (2026).

A coleta de informações sobre a experiência ocorreu por meio de observação direta em sala de aula pelo docente responsável, com registro em diário de campo das interações, dúvidas e dificuldades apresentadas pelas equipes ao longo das *Sprints*, complementada pela análise dos artefatos produzidos pelos estudantes (*Product Backlogs*, incrementos e apresentações finais). Somaram-se a esses registros as devolutivas orais realizadas em sala: ao final da apresentação de cada equipe, o docente forneceu *feedback* individualizado e em tempo real, prática que caracterizou a condução guiada da atividade e que também gerou material de observação sobre as reações e as dúvidas dos estudantes. Adicionalmente, ao final da atividade foi aplicado um questionário de percepção à turma, com perguntas fechadas sobre a experiência com a integração entre PBL e *Scrum*, respondido pela ampla maioria dos estudantes. Registra-se, como limitação deste relato, que as respostas não foram sistematicamente tabuladas nem preservadas para análise posterior; por essa razão, o instrumento é aqui tratado apenas como

fonte complementar de percepção, sem apresentação de dados quantitativos. A análise das informações coletadas foi realizada de forma qualitativa, a partir de quatro categorias definidas a posteriori: participação e engajamento; compreensão dos conceitos e artefatos de Engenharia de *Software* e do *Scrum*; desenvolvimento de competências relacionadas ao trabalho em equipe; e dificuldades enfrentadas durante a condução da atividade.

A combinação dessas fontes buscou reduzir a dependência de uma única perspectiva de análise. Em vez de sustentar os achados apenas na impressão do docente, procurou-se confrontá-la com o que os estudantes efetivamente produziram e com o que apresentaram publicamente ao final da atividade, conforme sintetizado na Tabela 3.

Tabela 3 – Fontes de evidência utilizadas na experiência

Momento	Fonte de evidência	O que permitiu observar
Durante as <i>Sprints</i>	Observação direta com registro em diário de campo	Participação, dúvidas conceituais, negociação de prioridades e necessidade de mediação docente
Durante e após as <i>Sprints</i>	Artefatos produzidos pelas equipes (Product Backlogs e incrementos)	Qualidade e evolução das histórias de usuário, dos critérios de aceite e da priorização
Encerramento	Apresentações finais por storytelling e devolutivas individualizadas em sala	Clareza na comunicação e justificativa das decisões tomadas por cada equipe
Encerramento	Questionário de percepção com perguntas fechadas	Percepção declarada pelos estudantes; respostas não preservadas, utilizadas apenas de forma qualitativa

Fonte: elaborado pelo autor (2026).

3 RESULTADOS E DISCUSSÃO

Ao longo da atividade, observou-se, na percepção do docente responsável, maior participação dos estudantes do que a habitualmente verificada nas aulas expositivas do mesmo conteúdo, o que é consistente com a literatura sobre metodologias ativas, segundo a qual o envolvimento do estudante em situações que exigem investigação e tomada de decisão favorece a autonomia e o comprometimento com a própria aprendizagem (Berbel, 2011). Foi possível observar, por exemplo, que os estudantes passaram a se manifestar com mais frequência durante as cerimônias de *Sprint Planning* e *Sprint Review*, propondo ajustes nas histórias de usuário e questionando critérios de aceite definidos pelos colegas que assumiam o papel de *Product*

Owner. As discussões em equipe também se tornaram mais frequentes fora do horário de aula, evidenciadas pelas atualizações incrementais registradas nos *Product Backlogs* entre uma *Sprint* e outra, o que sugere maior engajamento dos estudantes com o andamento do projeto.

Em relação à compreensão dos conceitos e artefatos de Engenharia de *Software* e do *Scrum*, notou-se que a vivência prática dos papéis, eventos e artefatos previstos no *framework* (Schwaber; Sutherland, 2020) contribuiu para que os estudantes relacionassem os conteúdos teóricos às situações enfrentadas durante a construção do *Product Backlog* e a execução das *Sprints*. Esse resultado é semelhante ao relatado por Meireles e Bonifácio (2015), que também observaram ganhos na compreensão de conceitos de Engenharia de *Software* ao associar métodos ágeis à Aprendizagem Baseada em Problemas. Um exemplo concreto dessa evolução foi a dificuldade inicial dos estudantes em diferenciar uma história de usuário de uma tarefa técnica, frequentemente descrevendo requisitos de forma genérica e sem critérios de aceite mensuráveis. Ao longo das *Sprints*, observou-se uma evolução na elaboração desses artefatos: os *Product Backlogs* revisados nas etapas finais passaram a apresentar histórias redigidas no formato: "Como [usuário], quero [funcionalidade], para [benefício]", acompanhadas de critérios de aceite mais objetivos. Ressalta-se, contudo, que essa evolução foi identificada a partir da análise dos artefatos produzidos, não tendo sido aplicada avaliação diagnóstica pré e pós-atividade que permitisse mensurar o ganho de aprendizagem de forma sistemática.

As principais dificuldades identificadas envolveram a familiaridade inicial dos estudantes com as práticas ágeis, sobretudo a distinção entre os papéis do *Scrum* e a elaboração de histórias de usuário com critérios de aceite bem definidos e o controle do tempo das cerimônias, especialmente nas primeiras *Sprints*. Essas dificuldades foram acompanhadas e ajustadas por meio da mediação docente, que atuou reforçando os prazos estabelecidos e esclarecendo dúvidas conceituais ao longo do processo. Na prática, essa mediação ocorreu por meio de breves reuniões de *feedback* ao final de cada *Sprint*, nas quais o docente comentava os *Product Backlogs* e os incrementos entregues por cada equipe, apontando ajustes necessários antes da *Sprint* seguinte. Além disso, durante as cerimônias de planejamento o docente atuou como facilitador do tempo, sinalizando quando as discussões se estendiam além do previsto e auxiliando as equipes a priorizar os pontos mais relevantes, o que contribuiu para que o controle do tempo melhorasse progressivamente ao longo das *Sprints*. Ao final da apresentação de cada equipe, o docente forneceu ainda devolutivas individualizadas em tempo real, diante da turma, comentando as escolhas feitas por cada grupo. Essa condução guiada, com acompanhamento contínuo e retorno imediato, mostrou-se um elemento central da experiência, uma vez que

permitiu corrigir desvios conceituais no momento em que se manifestavam e tornou o processo de avaliação parte da própria aprendizagem.

Além da compreensão técnica, a atividade favoreceu o desenvolvimento de competências como comunicação, organização, negociação de prioridades e trabalho em equipe, aspectos apontados na literatura como centrais na formação em Engenharia de *Software* e frequentemente pouco contemplados no ensino tradicional (Lima *et al.*, 2020; Santiago; Menezes; Aquino, 2023). Esses resultados sugerem que a integração entre as duas abordagens pode contribuir para uma formação mais próxima da realidade profissional em Engenharia de *Software*.

Registra-se ainda o caso de um estudante que conduziu a atividade individualmente. Diante da impossibilidade de distribuir os papéis entre integrantes, optou por simular as diferentes perspectivas previstas no *framework*, conduzindo as cerimônias a partir de cada uma delas e apresentando ao final um trabalho completo, com engajamento expressivo. A situação sugere que a flexibilidade quanto ao formato de participação pode ampliar o alcance da proposta, acomodando diferentes perfis e formas de trabalho.

4 INSTRUMENTOS DE ACOMPANHAMENTO E AVALIAÇÃO DA EXPERIÊNCIA

Durante a experiência, foram utilizados diferentes instrumentos e procedimentos de acompanhamento com o objetivo de observar a participação dos estudantes, acompanhar a evolução dos artefatos produzidos e identificar dificuldades relacionadas à compreensão e à aplicação dos conceitos de Engenharia de *Software* e *Scrum*. Esses procedimentos compreenderam a observação direta das atividades, registros realizados pelo docente, análise dos artefatos desenvolvidos pelas equipes, acompanhamento da evolução das atividades ao longo das *Sprints*, apresentações e momentos de *feedback*, além de um questionário de percepção aplicado ao final da experiência.

A utilização desses diferentes procedimentos permitiu acompanhar o desenvolvimento da atividade sob diferentes perspectivas. A observação docente possibilitou registrar aspectos relacionados à participação, à comunicação, à negociação de prioridades, à organização das equipes e às dificuldades encontradas durante a execução das atividades. Paralelamente, os artefatos produzidos pelas equipes, como histórias de usuário, critérios de aceitação e *Product Backlogs*, possibilitaram observar a evolução da compreensão dos conceitos trabalhados.

O questionário de percepção foi utilizado como instrumento complementar para identificar a percepção dos estudantes acerca da experiência e dos conceitos trabalhados. Embora sua aplicação tenha ocorrido ao final da atividade, as respostas não foram

sistematicamente tabuladas e preservadas em formato que permitisse uma análise quantitativa ou uma comparação estatística entre diferentes momentos. Dessa forma, seus resultados são considerados de caráter complementar e não constituem evidência quantitativa de aprendizagem ou de ganho de desempenho.

Para futuras aplicações, recomenda-se o aperfeiçoamento desse procedimento por meio da aplicação do mesmo questionário em dois momentos, antes e depois da experiência. O instrumento poderá ser composto por dez afirmações respondidas em escala Likert de cinco pontos, variando de “discordo totalmente” a “concordo totalmente”. As dimensões contempladas, apresentadas no Quadro 1, foram definidas a partir dos conceitos centrais trabalhados na atividade e das dificuldades observadas durante as *Sprints*, como a distinção entre os papéis do Scrum, a elaboração de histórias de usuário e a definição de critérios de aceitação verificáveis.

Quadro 1 – Itens propostos para o questionário de autopercepção (pré e pós-atividade)

Item	Afirmação
P1	Consigo explicar o que é o <i>framework Scrum</i> e sua finalidade no desenvolvimento de <i>Software</i> .
P2	Consigo diferenciar as responsabilidades do <i>Product Owner</i> , do <i>Scrum Master</i> e dos <i>Developers</i> .
P3	Consigo elaborar uma estória de usuário considerando usuário, funcionalidade e benefício.
P4	Consigo elaborar critérios de aceite claros e verificáveis para uma estória de usuário.
P5	Consigo organizar requisitos em um <i>Product Backlog</i> estruturado e priorizado.
P6	Compreendo como uma <i>Sprint</i> é planejada, executada e revisada.
P7	Sinto-me capaz de colaborar com colegas na resolução de problemas de desenvolvimento de <i>Software</i> .
P8	Sinto-me capaz de discutir e negociar prioridades entre funcionalidades.
P9	Consigo comunicar e justificar as decisões tomadas pela equipe.
P10	Consigo relacionar conceitos de Engenharia de <i>Software</i> e <i>Scrum</i> a uma situação prática.

Fonte: elaborado pelo autore (2026).

Além do questionário, foi estruturada uma rubrica de avaliação dos artefatos produzidos pelas equipes, apresentada no Quadro 2. A rubrica contempla sete critérios: histórias de usuário, critérios de aceitação, *Product Backlog*, aplicação do Scrum, participação da equipe, evolução entre *Sprints* e apresentação final. Cada critério pode ser avaliado em quatro níveis de

desempenho, permitindo complementar a percepção declarada pelos estudantes com evidências observáveis em suas produções.

Quadro 2 – Rubrica proposta para avaliação dos artefatos

Critério	1 – Insuficiente	2 – Básico	3 – Adequado	4 – Excelente
Estórias de usuário	Incompletas ou inadequadas	Parcialmente estruturadas	Adequadas	Claras e orientadas ao valor
Crítérios de aceite	Ausentes	Pouco objetivos	Claros	Claros, verificáveis e completos
<i>Product Backlog</i>	Desorganizado	Parcialmente estruturado	Organizado	Priorizado e consistente
Aplicação do <i>Scrum</i>	Pouca compreensão	Aplicação parcial	Aplicação adequada	Aplicação consistente
Participação da equipe	Baixa	Irregular	Boa	Colaborativa e equilibrada
Evolução entre <i>Sprints</i>	Sem evolução	Pequena evolução	Evolução adequada	Evolução evidente e justificada
Apresentação final	Pouco clara	Compreensão parcial	Clara	Clara, estruturada e fundamentada

Fonte: elaborado pelo autor (2026).

Na experiência relatada, esses instrumentos e procedimentos foram empregados em caráter predominantemente formativo e qualitativo, acompanhando a dinâmica da simulação ao longo do período destinado à atividade. A mediação docente ocorreu durante as *Sprints*, com orientações, questionamentos e *feedbacks* sobre os artefatos e as decisões das equipes. Dessa forma, o acompanhamento não se restringiu ao produto final, mas considerou também o processo de construção, revisão e aperfeiçoamento das soluções.

Para aplicações futuras, recomenda-se sistematizar esses procedimentos por meio de uma estratégia de triangulação de dados, articulando questionário pré e pós-atividade, rubrica de avaliação dos artefatos e registro estruturado das observações realizadas durante as *Sprints*. Essa combinação permitiria comparar a percepção dos estudantes, o desempenho demonstrado nas produções e os comportamentos observados durante a atividade, ampliando a base empírica para avaliar os efeitos pedagógicos da integração entre PBL e *Scrum*.

5 CONCLUSÃO

A experiência relatada indica que a integração entre a Aprendizagem Baseada em Problemas e o *framework Scrum* pode aproximar o ensino de Engenharia de *Software* da realidade profissional, associando conhecimentos técnicos à prática colaborativa e à resolução de problemas. Observou-se maior participação dos estudantes, melhor compreensão dos conceitos e artefatos relacionados ao conteúdo, e o desenvolvimento de competências como comunicação, organização, negociação de prioridades e trabalho em equipe.

Como limitação, destaca-se que a experiência foi conduzida em uma única turma e em um contexto institucional específico, o que impede generalizações amplas sobre os resultados. Soma-se a isso a composição desigual das equipes, uma vez que duas delas contaram com apenas dois e um integrante, respectivamente, o que torna heterogêneas as condições de vivência da dinâmica colaborativa entre os grupos, somando-se a isso o fato de cada equipe ter definido autonomamente a quantidade e a duração de suas *Sprints*, o que limita a comparabilidade direta entre os percursos percorridos, bem como o fato de as respostas ao questionário de percepção não terem sido sistematicamente registradas, o que impossibilitou seu tratamento quantitativo. Sugere-se, para trabalhos futuros, a aplicação da proposta em outras turmas e instituições, associada à utilização dos instrumentos apresentados na seção anterior, de modo a permitir uma análise mais aprofundada dos ganhos pedagógicos da integração entre PBL e *Scrum*.

Nesta experiência, o uso de ferramentas de gestão ágil (*Trello e Jira*) já foi incorporado à condução das etapas, apoiando a criação, o detalhamento e a movimentação dos cards referentes às histórias de usuário e às tarefas de cada *Sprint*. Para iterações futuras do conteúdo e para instituições que pretendam replicar esta abordagem, recomenda-se a realização de uma oficina prévia de nivelamento sobre métricas de estimativa ágil, bem como a exploração mais aprofundada de recursos avançados dessas ferramentas, como quadros *Kanban*, *burndown charts* e relatórios automatizados, de modo a aproximar ainda mais o ecossistema acadêmico das plataformas utilizadas profissionalmente. Conclui-se que o investimento docente em metodologias ativas integradas pode constituir uma estratégia relevante para a formação de profissionais de computação críticos e autônomos, mais bem preparados para os desafios contemporâneos da sociedade digital.

REFERÊNCIAS

- BARROWS, H. S. A taxonomy of problem-based learning methods. **Medical Education**, v. 20, n. 6, p. 481-486, 1986.
- BERBEL, N. A. N. As metodologias ativas e a promoção da autonomia de estudantes. **Semina: Ciências Sociais e Humanas**, Londrina, v. 32, n. 1, p. 25-40, 2011.
- LIMA, J. V. V.; SILVA, C. A. D.; ALENCAR, F. M. R. de; SANTOS, W. B. Metodologias ativas como forma de reduzir os desafios do ensino em Engenharia de *Software*: diagnóstico de um survey. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 31., 2020. **Anais [...]** Porto Alegre: SBC, 2020. p. 172-181.
- MEIRELES, M. A. C.; BONIFÁCIO, B. A. Uso de métodos ágeis e aprendizagem baseada em problema no ensino de engenharia de *Software*: um relato de experiência. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 26., 2015, Maceió. **Anais [...]** Maceió: CEIE/SBC, 2015. p. 180-189.
- SANTIAGO, C. P.; MENEZES, J. W. M.; AQUINO, F. J. A. de. 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**, v. 31, p. 31-59, 2023.
- SCHWABER, K.; SUTHERLAND, J. **The Scrum Guide: the definitive guide to Scrum, the rules of the game**. Scrum.org; Scrum Inc., 2020.