

## CAPÍTULO 6

# METODOLOGIA DE APRENDIZAGEM BASEADA EM PROJETOS COM ÊNFASE EM COMPETÊNCIAS EM DISCIPLINAS DE ENGENHARIA DE SOFTWARE

*Cynthia Pinheiro Santiago  
José Wally Mendonça Menezes  
Francisco José Alves de Aquino*

## RESUMO

Nos últimos anos têm-se percebido uma lacuna entre o ensino nas universidades e as necessidades da indústria em relação à formação dos futuros profissionais de áreas relacionadas à Computação. Enquanto por um lado, as aulas possuem um denso conteúdo teórico-conceitual e são apresentadas de forma expositiva, por outro lado a falta de experiências realistas torna desafiador para os alunos adquirir habilidades básicas necessárias para a colaboração no desenvolvimento de projetos de *software*. Como consequência, estas áreas apresentam um percentual elevado de evasão estudantil, enquanto o déficit de profissionais em tecnologia já é expressivo e preocupante. Este capítulo tem como objetivo apresentar a metodologia de aprendizagem baseada em projetos no contexto de disciplinas de Engenharia de *Software* como forma de aproximar teoria e prática, através do engajamento do aluno em projetos realistas que contribuam para sua permanência e êxito durante a graduação e para seu futuro sucesso profissional. Para tanto, são apresentadas iniciativas bem-sucedidas de aplicação desta metodologia, considerando os processos, as etapas de desenvolvimento e os artefatos de *software* em contextos que simulam o ambiente industrial. Os resultados obtidos sugerem ser esta uma metodologia promissora - tanto do ponto de visto do ensino como da aprendizagem - para futuros engenheiros de *software*.

**Palavras-chave:** Ensino de engenharia de *software*. Aprendizagem baseada em projetos. Desenvolvimento de competências.

## INTRODUÇÃO

Nos próximos anos, há uma projeção de que o número de graduados em áreas ligadas à Computação será insuficiente para atender o mercado de Tecnologia da Informação e Comunicação (TIC)<sup>3</sup>. Além disso, ainda é possível perceber uma lacuna entre a academia e a indústria com relação à formação dos futuros profissionais de áreas correlatas à Computação: frequentemente a academia com seu ensino tradicional, com aulas expositivas e grande conteúdo teórico, não atende às necessidades da indústria, que é dinâmica e em constante transformação, onde o engenheiro de *software* deve ser capaz de desenvolver habilidades técnicas e não técnicas a fim de cumprir com suas atribuições (OGUZ; OGUZ, 2019).

Uma das formas de conseguir esse objetivo é simular entre os estudantes, ainda durante a sua formação, um ambiente industrial onde eles possam fixar os conteúdos colocando-os em prática da mesma forma como ocorre na indústria: através de projetos de *software*.

Com este fim, neste capítulo apresentamos a metodologia de aprendizagem baseada em projetos (ABPj) que é uma das mais mencionadas e aplicadas na literatura no contexto do ensino em áreas relacionadas à Computação (LIMA et al., 2019). Esta metodologia leva ainda em conta o desenvolvimento de competências técnicas (*hard skills*) e não técnicas (*soft skills*) dos estudantes, uma vez que estes relatam dificuldades com o trabalho em equipe, comunicação ou resolução de conflitos (GROENEVELD et al., 2019).

Este capítulo também traz um conjunto de relatos de experiências bem-sucedidas do uso da metodologia ABPj em disciplinas relacionadas a ES, o que comprova sua eficácia tanto em relação ao aprendizado quanto à satisfação dos estudantes, habilitando-a assim a ser aplicada amplamente em disciplinas de ES e Computação ou mesmo a ser adaptada a outras disciplinas de Engenharia, considerando as especificidades de cada caso.

---

3 <https://www.correiobraziliense.com.br/euestudante/trabalho-e-formacao/2021/05/4926392-apagao-na-area-deti-sobram-vagas-mas-falta-mao-de-obra.html>. Acesso em 05/05/2022.

O restante deste capítulo está organizado como se segue: na seção 2 é descrito o referencial teórico deste capítulo ao aprofundar sobre o ensino de Engenharia de *Software*, a metodologia ABPj e como aplicá-la no ensino de ES; na seção 3 listamos alguns trabalhos recentes que relatam experiências bem sucedidas na aplicação desta metodologia em disciplinas de Computação e Engenharia de *Software* e, por fim, na seção 4, apresentamos algumas considerações finais, bem como intenções de trabalhos futuros.

## REFERENCIAL TEÓRICO

### Ensino de Engenharia de *Software* (ES)

A ES, como área do conhecimento, trata da aplicação de abordagens sistemáticas, disciplinadas e quantificáveis para desenvolver, operar, manter e evoluir *software*. Em outras palavras, é a área da Computação que se preocupa em propor e aplicar princípios de engenharia na construção de *software* (VALENTE, 2020).

A ES, como disciplina, tem seu currículo baseado no livro SWEBOOK (*Software Engineering Body of Knowledge* ou Corpo de Conhecimento em Engenharia de *Software*, em português), composto por quinze áreas de conhecimento, juntamente com sete disciplinas relacionadas (BOURQUE; FAIRLEY, 2004). O SWEBOOK tem como objetivo direcionar os alunos quanto às habilidades necessárias para realizar a transição para o mercado de trabalho e se adequar à indústria (DEVADIGA, 2017). A dimensão das habilidades contida no SWEBOOK representa as capacidades dos alunos e de engenheiros de *software*, que são adquiridas tanto por educação formal como por experiência, dependendo do tipo e do contexto (OGUZ; OGUZ, 2019).

Normalmente o conteúdo do SWEBOOK é apresentado nas disciplinas de ES com um denso conteúdo teórico-conceitual e ensinado de forma tradicional, com aulas expositivas e leituras complementares (CUNHA et al., 2018).

A natureza fundamentalmente prática da ES, na maioria das vezes, entra em confronto com a maneira predominantemente teórica com que se dá o ensino nestas disciplinas. O ensino focado apenas na parte teórica pode ocasionar a desmotivação dos estudantes, visto que, eles podem não conseguir entender como os problemas surgem ou quais suas principais causas e consequências em um ambiente real (TONHÃO et al., 2021).

Em outras situações, o curso de ES pode contemplar a competência prática, mediante o desenvolvimento de um projeto final (CUNHA et al., 2018). No entanto, os estudantes percebem que os projetos da vida real são diferentes daqueles que realizaram durante sua educação. Essa situação cria uma lacuna entre a academia e a indústria onde, entre os possíveis motivos, estão: (i) a profissão de engenheiro de *software* é capaz de reagir rapidamente a novas plataformas e tendências que exigem a aquisição de novas habilidades, enquanto que a academia não é; (ii) a academia não é rápida o suficiente para incorporar as mudanças na profissão em seu currículo; (iii) é um desafio criar experiências realistas no ensino de engenharia e (iv) a falta de experiências realistas torna mais desafiador para os alunos adquirir habilidades básicas que são necessárias para a colaboração no desenvolvimento de projetos de *software* em grande escala (OGUZ; OGUZ, 2019).

Integrar teoria e prática é um grande desafio para as universidades, sendo que esse processo exige tempo e recursos. Seriam necessários também professores com experiência em desenvolvimento de *software* e, em alguns casos, clientes reais para atuarem em contrapartida nos projetos (CUNHA et al., 2018).

Por outro lado, as diretrizes curriculares do ACM/IEEE enfatizam a necessidade de proporcionar aos alunos experiências práticas suficientes para o desenvolvimento das competências esperadas em profissionais de ES (ARDIS et al., 2015). No Brasil, baseados nas Diretrizes Curriculares Nacionais<sup>4</sup>, os referenciais de formação para as disciplinas de graduação na área da Computação, incluindo ES, apresentam uma proposta baseada em competências, em vez de conteúdo.

Nesse contexto, *soft skills*, como liderança, trabalho em equipe, tomada de decisão, negociação e autorreflexão, são competências importantes para a prática de ES, uma vez que o desenvolvimento de *software* também envolve diversos aspectos humanos e sociais (Figura 6.1). No entanto, o desenvolvimento dessas capacidades transversais é geralmente pouco apoiado em programas de graduação em Ciência da Computação ou ES (SOUZA et al., 2019).

---

4 Disponível em [https://www.in.gov.br/materia/-/asset\\_publisher/Kujrw0TZC2Mb/content/id/22073129/do12016-11-17-resolucao-n-5-de-16-de-novembro-de-2016-22073052](https://www.in.gov.br/materia/-/asset_publisher/Kujrw0TZC2Mb/content/id/22073129/do12016-11-17-resolucao-n-5-de-16-de-novembro-de-2016-22073052). Acesso em: 16/04/2022.

Figura 6.1 - *Soft Skills* em Engenharia de *Software*.



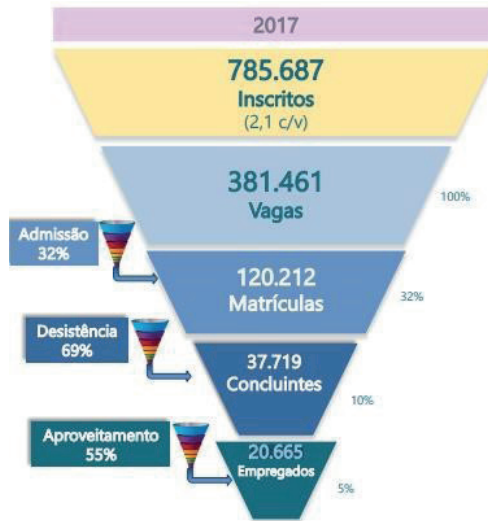
Fonte: Traduzido de <https://careerkarma.com/careers/software-engineer/>.

Embora existam muitos estudos que enfatizam a importância das habilidades sociais, elas também são uma das causas da lacuna entre a academia e a indústria, uma vez que os engenheiros de *software* em início de carreira relatam que não se sentem preparados para a comunicação e o trabalho em equipe, onde as *soft skills* são mais necessárias (OGUZ; OGUZ, 2019). Em muitos casos as empresas de *software* têm que complementar os conhecimentos dos recém-formados com treinamentos e prover meios de desenvolver tanto *hard* como *soft skills*, relacionadas ao desenvolvimento de sistemas de *software* (MEIRELES; BONIFÁCIO, 2015).

Outro problema a ser enfrentado é a grande evasão que existe em cursos relacionados à Computação, como ES. Em um estudo realizado por Baggi e Lopes (2011), sobre a evasão no ensino superior de uma forma geral, verificou-se que as causas para a evasão são muito diversas e dependem de fatores sociais, culturais, políticos e econômicos em que a instituição se encontra. Uma causa possível seria a má qualidade do ensino ofertado pela instituição, levando à desmotivação e perda do aluno. Isso requer uma maior reflexão a respeito, permitindo assim elaborar propostas mais eficazes de combate à evasão. Sendo assim a evasão pode estar, em muitos casos, associada diretamente a fatores institucionais (HOED, 2016). Em se tratando de cursos relacionados à Computação a situação não é diferente.

Segundo o Relatório Brasscom<sup>5</sup>, sobre Formação Educacional e Empregabilidade em Tecnologias da Informação e Comunicação (TIC), há uma demanda prevista de 420 mil profissionais entre 2018-2024, o que quer dizer que serão necessários 70 mil profissionais ao ano até 2024. No entanto, a oferta é de apenas 46 mil alunos formados com perfil tecnológico no ensino superior por ano, o que incorre em um grande déficit de profissionais. Segundo dados de 2017 do presente relatório, foram ofertadas 381.461 vagas no ensino superior para a formação presencial em TIC, para as quais houveram 785.687 inscritos, com a realização de 120.212 matrículas. No entanto, dos alunos matriculados, apenas 37.719 concluíram o curso e, destes, apenas 20.665 empregaram-se, conforme pode ser visto no resumo da Figura 6.2.

**Figura 6.2** - Inscritos x Vagas x Matrículas x Concluintes x Empregados em 2017 para formados com perfil tecnológico no ensino superior.



Fonte: Relatório Brasscom, 2023.

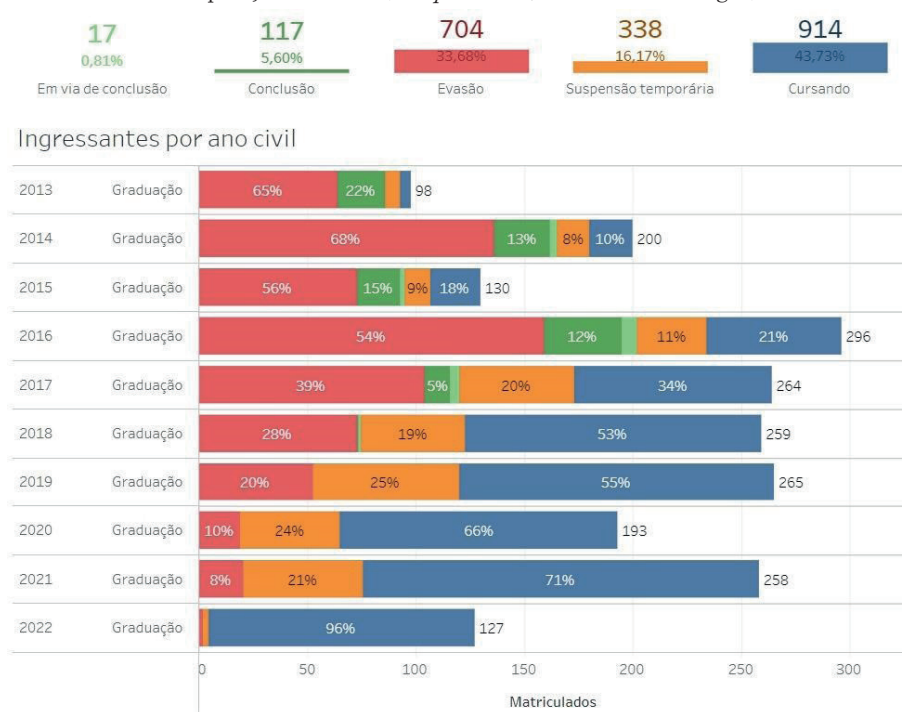
No Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE), situação semelhante ocorre nos cursos de Bacharelado em Ciência da Computação e Engenharia de Computação. De 2013 até 2022, segundo o portal IFCE em Números, para os cursos de Bacharelado em Ciência da

5 Disponível em <https://brasscom.org.br/wp-content/uploads/2021/10/BRI2-2019-010-P02-Formacao-Educacional-e-Empregabilidade-em-TIC-v83.pdf>. Acesso em 16/04/2022

Computação, de um total de 2090 estudantes matriculados, 704 (33,68%) encontravam-se em situação de evasão e apenas 117 (5,6%) haviam concluído o curso (Figura 6.3). De forma similar, para o curso de Engenharia de Computação, de um total de 841 estudantes matriculados, 273 (32,43%) encontravam-se em situação de evasão e somente 70 (8,32%) haviam concluído o curso (Figura 6.4).

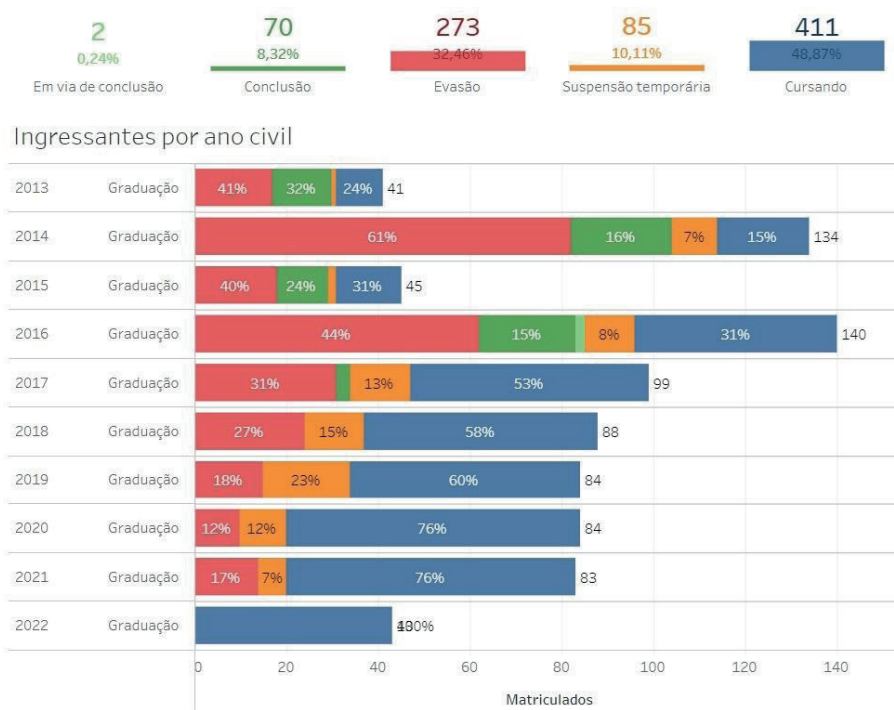
Como agravante para esta situação de evasão, podemos ter problemas ligados à didática no ensino em sala de aula, o que dificulta a aprendizagem. Segundo Borges (2000), o modo tradicional de ensino de disciplinas que envolvem algoritmos e programação, por exemplo, não consegue facilmente motivar os alunos a se interessar pelo assunto. Entre outras razões, isso ocorre, porque não é clara para os estudantes a importância de certos conteúdos para sua formação. É importante que o ensino seja prazeroso e englobe situações reais e dinâmicas para envolver o discente (HOED, 2016).

**Figura 6.3** - Situação de matrículas dos cursos de Ciência da Computação no IFCE (Campi Aracati, Maracanaú e Tianguá).



Fonte: IFCE em Números, 2022.

**Figura 6.4** - Situação de matrículas do curso de Engenharia de Computação no IFCE (*Campus Fortaleza*)



Fonte: IFCE em Números, 2022.

Segundo Oguz e Oguz (2019), o objetivo da academia não deveria ser a formação profissional propriamente dita, mas um meio-termo que equilibre os princípios fundamentais com os aspectos práticos da profissão. Para tanto, as universidades podem considerar, entre outras, as seguintes opções para diminuir a lacuna entre a academia e a indústria: (i) os professores devem se envolver em projetos reais da indústria para acompanhar as novas metodologias da profissão; (ii) os profissionais da indústria podem ser convidados para realizar exposições e apresentações para o corpo discente em disciplinas de ES; (iii) o realismo nos projetos do curso pode ser obtido convidando clientes reais para expor suas necessidades, em colaboração direta com a indústria; (iv) os professores devem apresentar projetos interessantes para atrair a atenção dos alunos, de modo que eles fiquem motivados a trabalhar; (v) a academia deve revisar o programa do curso com práticas técnicas e sociotécnicas, para englobar não



apenas as *hard skills*, mas também as *soft skills* e (vi) os currículos dos programas de ES devem considerar diferentes abordagens de ensino e aprendizagem como, por exemplo, a aprendizagem baseada em projetos.

## **Aprendizagem Baseada em Projetos (ABPj)**

O desenvolvimento da metodologia ABPj teve suas origens em 1900, quando o filósofo americano John Dewey (1859 – 1952) comprovou o “aprender mediante o fazer”, valorizando, questionando e contextualizando a capacidade de pensar dos alunos numa forma gradativa de aquisição de um conhecimento relativo para resolver situações reais em projetos referentes aos conteúdos na área de estudos, que tinha como meta o desenvolvimento destes no aspecto físico, emocional e intelectual, por meio de métodos experimentais (MASSON *et al.*, 2012).

A ABPj é, portanto, uma proposta de ensino-aprendizagem que se concentra na concepção central e nos princípios de uma tarefa, envolvendo o aluno na investigação de soluções para os problemas e em outros objetivos significativos, permitindo assim ao estudante trabalhar de forma autônoma na construção do seu próprio conhecimento (TOYOHARA *et al.*, 2010). De acordo com Barbosa e Moura (2013), os projetos partem de um problema, uma necessidade, uma oportunidade ou interesses de uma pessoa, um grupo de pessoas ou uma organização. Entre as principais características dessa metodologia estão: o aluno é o centro do processo; desenvolve-se em grupos tutoriais; caracteriza-se por ser um processo ativo, cooperativo, integrado, interdisciplinar e orientado para a aprendizagem do aluno (MASSON *et al.*, 2012).

Sendo assim, três importantes critérios promovem um aprendizado mais eficaz com ABPj: (i) o aprendizado acontece em um ambiente onde os estudantes estão imersos na prática, em atividades em que recebem *feedback* de seus colegas estudantes e professores; (ii) os estudantes recebem guias e suporte de seus pares, de maneira a promover um ensino multidirecional envolvendo outros estudantes, professores e monitores, diferentemente do ensino convencional, normalmente unidirecional (do professor para o estudante); (iii) o aprendizado é funcional, a partir de problemas reais (CUNHA *et al.*, 2018).

Na ABPj, os alunos geralmente recebem especificações sobre um produto e são incentivados a desenvolvê-lo de acordo com procedimentos bem definidos.

Na medida em que vão desenvolvendo esse produto, podem se deparar com situações onde terão que resolver problemas que requerem um raciocínio sobre possíveis soluções, trazendo consequentemente momentos de aprendizagem a partir da resolução de problemas dentro do projeto (TONHÃO *et al.*, 2021).

A ABPj tem como uma das suas maiores vantagens a de criar ambientes de aprendizado empolgantes, reais e adaptados, estimulando a motivação e engajamento dos estudantes, características que são dificilmente encontradas no ambiente tradicional de ensino. Além disso, ela pode ser importante na exploração das competências individuais e do trabalho em equipe, e pode permitir ao estudante o desenvolvimento de habilidades de gerenciamento de projetos e resolução de conflitos (TONHÃO *et al.*, 2021).

Segundo Uzun, Pugliesi e Roland (2018), ainda existem outras vantagens como: os alunos são motivados pelo dinamismo ao se envolverem com o projeto, pois interagem com a realidade e despertam a curiosidade para complementar as informações básicas obtidas para o desenvolvimento do projeto; os alunos conseguem desenvolver habilidades de relacionar diversas disciplinas/conteúdos objetivando encontrar a solução do projeto; desenvolve o pensamento crítico, pois é necessário que o aluno reflita, elabore e organize os conhecimentos adquiridos para desenvolver o projeto; impulsiona a interação e as habilidades interpessoais, já que os alunos precisam conviver e trabalhar com os colegas.

Por outro lado, um dos problemas da ABPj é que, quando se desenvolve um projeto, os estudantes e o professor podem perder o foco dos objetivos de aprendizagem estabelecidos. Assim, o desenvolvimento do projeto pode não demonstrar os conteúdos curriculares propostos para a aprendizagem. Mesmo que o protagonismo dos discentes nessas situações de aprendizado seja mais acentuado, é imprescindível acompanhar cada etapa do projeto, cabendo ao professor oferecer-lhes ajuda e orientação durante todo esse período (SALES *et al.*, 2020).

Outras desvantagens são: a insegurança inicial dos alunos, por ser um método diferente de ensino-aprendizagem; o tempo maior que é necessário para a aplicação de ABPj em relação ao ensino tradicional, pois a construção do conhecimento é um caminho que demanda mais tempo; a inadequação do currículo, já que os conteúdos necessários para resolver o problema são ministrados nas disciplinas de forma distinta, o que dificulta para os alunos chegar

ao resultado final; a forma de avaliação, que requer maior critério e discernimento e, por fim, a falta de preparo do próprio professor, que pode prejudicar a aplicabilidade do método (UZUN; PUGLIESI; ROLAND, 2018).

## **Aprendizagem Baseada em Projetos no Ensino de Engenharia de Software**

A ABPj, de uma forma geral, é definida pela utilização de projetos realistas, baseados em uma questão, tarefa ou problema motivador e envolvente, com o objetivo de ensinar conteúdos acadêmicos aos alunos no contexto do trabalho cooperativo para a resolução de problemas (BENDER, 2015).

A motivação da ABPj é que o grupo seja capaz de criar um possível projeto para construir, investigar ou explicar um problema. Esse esforço para organizar projetos em torno de situações reais torna o estudante protagonista de sua aprendizagem através de uma experiência de educação autêntica, com tarefas que são solicitados a concretizar no mundo a sua volta (ANTUNES *et al.*, 2019).

A ABPj é uma abordagem que vem ganhando destaque no ensino de ES, visto que possui a capacidade de proporcionar experiências práticas aos estudantes. A utilização de exemplos reais, de planos e ferramentas baseados na indústria, assim como a possibilidade de progressão dos alunos no ciclo de vida do *software*, são considerados importantes aspectos da ABPj em ES, pois aproximam a aprendizagem daquilo que a indústria exige (TONHÃO *et al.*, 2021). Nesse contexto, os estudantes precisam saber não só como codificar adequadamente, mas também precisam saber manejar-se em áreas de ES como Requisitos, Arquitetura e Testes. Além disso, eles devem ser capazes de compreender e usar ferramentas, bibliotecas e *frameworks* - os elementos integrantes do desenvolvimento de *software* moderno (GUPTA; NGUYEN-DUC, 2021).

Com relação às *soft skills*, o elemento mais comum do ABPj em ES é o trabalho em equipe: os alunos devem ser capazes de compreender a importância de um bom trabalho em equipe e a dinâmica de equipe. Espera-se que os alunos enfrentem situações como lidar com clientes difíceis, coordenar esforços, distribuição de tarefas e responsabilidades e resolução coletiva de problemas. Os alunos obterão, assim, habilidades relacionadas à comunicação,

gerenciamento de tarefas, tomada de decisão coletiva, retrospectiva de equipe e liderança (GUPTA; NGUYEN-DUC, 2021).

## Propostas de uso de ABPJ em ES

Na presente seção, serão abordados diferentes trabalhos que contemplam os temas “Ensino de ES” e “ABPJ no ensino de ES”, que são estreitamente relacionados ao tema deste trabalho. O objetivo desta seção é oferecer um embasamento teórico na forma de uma revisão informal de literatura dos tópicos tratados neste capítulo.

Segundo Sales *et al.* (2020), na literatura atual não existe algo que forneça uma orientação sistemática para operacionalizar a ABPJ em nível de graduação, permitindo experimentações de conteúdos com viabilidade comprovada. No entanto, recentes trabalhos foram bem-sucedidos no sentido de integrar ABPJ e ES.

Por exemplo, a abordagem proposta por Tonhão *et al.* (2021) visa a aplicação da ABPJ juntamente com técnicas de gamificação. Nesta abordagem, o projeto, o escopo e as áreas de ES que os estudantes vão trabalhar são definidos no início da disciplina, assim como um cronograma de execução é elaborado pelo professor, com ordem e tempo de aplicação para as atividades. Na sequência, as equipes são divididas e, no decorrer do projeto, pontos e medalhas são atribuídos aos grupos de acordo com o cumprimento do cronograma.

Serrano *et al.* (2021) relatam a experiência sobre o uso da ABPJ na disciplina de Requisitos de *Software* (RS), uma das áreas de ES segundo o SWEBOK. A abordagem orientou-se por módulos, onde cada módulo tinha a perspectiva voltada para um tópico de relevância da área de RS, como técnicas de elicitação e de priorização de requisitos, por exemplo. Concluiu-se que esta abordagem foi generalista o suficiente para atender disciplinas com perfis similares a RS: essencialmente teórica, onde se torna difícil despertar o interesse dos discentes e desenvolver competências profissionais desejadas no mercado, tais como proatividade, senso crítico e harmonia no trabalho coletivo.

Uzun, Pugliesi e Roland (2018) utilizaram ABPJ envolvendo as disciplinas de Estatística Aplicada, Interação Humano Computador, ES e Linguagem de Programação. Foi proposto aos alunos o desenvolvimento de um *software* de Análise Estatística, com levantamento de requisitos, documentação e projeto

do *software*. Todas estas atividades utilizavam técnicas e métodos de ES, bem como prototipação da interface com conceitos, padrões e ferramentas de Interação Humano Computador. Como ponto forte da aplicação deste método esteve a satisfação dos alunos, já que 93,8% deles disseram que gostaram de ser desafiados com projetos inovadores que vão proporcionar aprendizagem em diferentes esferas. Por meio da elaboração do projeto solicitado, os alunos tiveram a oportunidade de construir e reconstruir seu conhecimento, houve mais interação com os colegas e professor, conseguiram relacionar diferentes disciplinas, desenvolveram habilidades comunicativas; se comprometeram com as disciplinas e com a aproximação da teoria com a prática.

Vázquez-Ingelmo *et al.* (2019) relatam a experiência de três anos aplicando ABPj em uma disciplina de ES. Neste caso, no início da disciplina, enuncia-se o contexto do projeto e uma série de objetivos gerais que o sistema a ser modelado deve atender. São os próprios alunos que devem definir a especificação dos requisitos e desenvolver o modelo do sistema de acordo com os seus próprios critérios, valorizando-se a originalidade das soluções propostas e a correta execução dos processos de engenharia. O projeto segue uma estrutura de entregas incrementais baseadas em marcos, de forma que estes sejam entregues ao longo do curso. Na primeira etapa, os alunos devem especificar os requisitos do sistema. Na segunda, o modelo de domínio e um breve relatório técnico, e na terceira e última etapa, o modelo de análise completo. Os professores desempenham o papel de clientes, fornecendo *feedbacks* semanais a cada grupo. Por outro lado, os alunos podem corrigir problemas detectados pelos clientes, os quais terão impacto direto na nota obtida na etapa correspondente.

Gupta e Nguyen-Duc (2021) contam com a participação de clientes reais nos projetos que são desenvolvidos nas disciplinas de ES. Neste caso, os clientes participam nos cursos acadêmicos para obter algumas vantagens como ter acesso a alunos para fins de recrutamento e obtenção de benefícios diretos ou indiretos dos resultados do projeto. Os alunos se envolvem em todas as três fases do projeto: planejamento, execução e encerramento. Na fase de planejamento, os alunos conhecem os membros da equipe, clientes e suas necessidades. Geralmente, os alunos decidem voluntariamente suas funções e áreas de responsabilidade. Eles também fazem um plano de projeto preliminar e configuram o ambiente de trabalho. Na fase de execução, os projetos costumam ser divididos em módulos com entregas frequentes às partes interessadas.

Os projetos dos alunos devem demonstrar execução de atividades de ES (ou seja, elicitação de requisitos, *design* de sistema, codificação e testes). Na fase de encerramento, os resultados do projeto são demonstrados e apresentados aos clientes.

Todos os estudos acima avaliam os artefatos de *software* produzidos pelos estudantes quanto à qualidade e ao tempo de entrega, como fatores para cálculo de notas.

## CONSIDERAÇÕES FINAIS

Neste trabalho, procuramos apresentar um panorama atual do ensino em ES bem como a metodologia ABPj e listamos, a título de revisão informal da literatura, relatos de experiências bem-sucedidas quanto à aplicação deste método em disciplinas de cursos de Computação e Engenharia de *Software*. Com base nos resultados apresentados, consideramos ser este ser um método promissor tanto do ponto de vista do ensino como da aprendizagem de futuros engenheiros de *software*, já que simula ainda na academia a prática profissional da indústria.

Como trabalhos futuros, para melhor avaliar a hipótese mencionada no parágrafo anterior, em um primeiro momento, será feita uma revisão sistemática da literatura a respeito do uso de ABPj em disciplinas relacionadas à ES e, em um momento posterior, será conduzida uma pesquisa experimental envolvendo a aplicação deste método em turmas regulares das referidas disciplinas, obtendo-se dessa forma dados quantitativos e qualitativos com o fim de avaliar a eficácia do método neste contexto.

## REFERÊNCIAS

ANTUNES, Jeferson *et al.* Metodologias ativas na educação: problemas, projetos e cooperação na realidade educativa. **Informática na Educação: teoria & prática**, v. 22, n. 1, 30 maio 2019. Universidade Federal do Rio Grande do Sul. <http://dx.doi.org/10.22456/19821654.88792>.

ARDIS, Mark *et al.* SE 2014: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering. **Computer**, v. 48, n. 11, p. 106-109, 23 fev. 2015. Institute of Electrical and Electronics Engineers (IEEE). <http://dx.doi.org/10.1109/mc.2015.345>.

BAGGI, Cristiane Aparecida dos Santos; LOPES, Doraci Alves. Evasão e avaliação institucional no ensino superior: uma discussão bibliográfica. **Avaliação: Revista da Avaliação da Educação Superior**, Campinas, v. 2, n. 16, p. 355-374, jul. 2011.

BARBOSA, Eduardo Fernandes; MOURA, Dácio Guimarães de. Metodologias ativas de aprendizagem na educação profissional e tecnológica. **Boletim Técnico do SENAC**, v. 39, n. 2, p. 48-67, nov. 2013.

BENDER, Willian N. **Aprendizagem Baseada em Projetos: Educação Diferenciada para o Século XXI**. Penso Editora, 2015.

BORGES, Marcos Augusto F. Avaliação de uma Metodologia Alternativa para a Aprendizagem de Programação. **VIII Workshop de Educação em Computação**, Curitiba, n. 8, p. 15, jul. 2000.

BOURQUE, Pierre; FAIRLEY, R. E. **SWEBOK : Guide to the Software Engineering Body of Knowledge**. Los Alamitos, CA: IEEE Computer Society, 2004.

CUNHA, José Adson O. G. da *et al.* Software engineering education in Brazil. **Proceedings of the XXXII Brazilian Symposium on Software Engineering - SBES '18**, p. 348-356, set. 2018. ACM Press. <http://dx.doi.org/10.1145/3266237.3266259>.

DEVADIGA, Nitish M.. Software Engineering Education: converging with the startup industry. **2017 IEEE 30th Conference on Software Engineering Education and Training (CSEE&T)**, p. 192-196, nov. 2017. IEEE. <http://dx.doi.org/10.1109/cseet.2017.38>. GROENEVELD, Wouter *et al.* Software Engineering Education Beyond the Technical. **Acro PHD Seminars**, Diepenbeek, Belgium, maio 2019.

GUPTA, Varun; NGUYEN-DUC, Anh. **Real-World Software Projects for Computer Science and Engineering Students**. Boca Raton, FL: CRC Press, 2021.

HOED, Raphael Magalhães. **Análise da evasão em cursos superiores: o caso da evasão em cursos superiores da área de Computação**. 2016. 188 f. Dissertação (Mestrado) - Mestrado Profissional em Computação Aplicada, Universidade de Brasília, Brasília, 2016.

LIMA, José Vinícius *et al.* As Metodologias Ativas e o Ensino em Engenharia de Software: uma revisão sistemática da literatura. **Anais do XXV Workshop de Informática na Escola (WIE 2019)**, v. 25, n. 1, p. 1014-1023, 11 nov. 2019. Sociedade Brasileira de Computação (SBC). <http://dx.doi.org/10.5753/cbie.wie.2019.1014>.



MASSON, Terezinha Jocelen *et al.* Metodologia de ensino: aprendizagem baseada em projetos (PBL). **Anais do XI Congresso Brasileiro de Educação em Engenharia (COBENGE)**, Belém, p. 13, set. 2012.

MEIRELES, Maria Alcimar Costa; BONIFÁCIO, Bruno Araújo. Uso de Métodos Ágeis e Aprendizagem Baseada em Problema no Ensino de Engenharia de Software: um relato de experiência. **Anais do XXVI Simpósio Brasileiro de Informática na Educação (SBIE 2015)**, v. 26, n. 1, p. 180, 26 out. 2015. Sociedade Brasileira de Computação - SBC. <http://dx.doi.org/10.5753/cbie.sbie.2015.180>.

OGUZ, Damla; OGUZ, Kaya. Perspectives on the Gap Between the Software Industry and the Software Engineering Education. **IEEE Access**, v. 7, p. 117527-117543, 04 set. 2019. Institute of Electrical and Electronics Engineers (IEEE). <http://dx.doi.org/10.1109/access.2019.2936660>.

SALES, André Barros de *et al.* Aprendizagem Baseada em Projetos na Disciplina de Interação Humano-Computador. **Revista Ibérica de Sistemas e Tecnologias de Informação**, n. 37, p. 49-64, jun. 2020.

SERRANO, Milene *et al.* DESENVOLVIMENTO DE COMPETÊNCIAS PROFISSIONAIS: relato da experiência utilizando aprendizagem baseada em projetos na disciplina de requisitos de software. **Revista de Ensino de Engenharia**, v. 40, n. 1, p. 76-81, 2021. Revista de Ensino em Engenharia. <http://dx.doi.org/10.37702/ree2236-0158.v40p76-81.2021>.

SOUZA, Maurício *et al.* Students Perception on the use of Project-Based Learning in Software Engineering Education. **Proceedings of the XXXIII Brazilian Symposium on Software Engineering**, p. 537-546, 23 set. 2019. ACM. <http://dx.doi.org/10.1145/3350768.3352457>.

TONHÃO, Simone de França *et al.* Uma abordagem prática apoiada pela aprendizagem baseada em projetos e gamificação para o ensino de Engenharia de Software. **Anais do I Simpósio Brasileiro de Educação em Computação (EDUCOMP 2021)**, p. 143-151, 26 abr. 2021. Sociedade Brasileira de Computação. <http://dx.doi.org/10.5753/educomp.2021.14480>.

TOYOHARA, Doroti Quiomi Kanashiro *et al.* Aprendizagem Baseada em Projetos – uma nova Estratégia de Ensino para o Desenvolvimento de Projetos. **PBL–Congresso Internacional**, São Paulo, fev. 2010.

UZUN, Maria Luísa Cervi; PUGLIESI, Jaqueline Brigladori; ROLAND, Carlos Eduardo de França. Aprendizagem baseada em projetos na perspectiva dos alunos. **Revista Profissão Docente**, [S.L.], v. 18, n. 39, p. 403-414, 21 dez. 2018. Revista Profissão Docente. <http://dx.doi.org/10.31496/rpd.v18i39.1212>.



VALENTE, Marco Tulio. **Engenharia de Software Moderna: Princípios e Práticas para Desenvolvimento de Software com Produtividade**. 2020. 408 p. Disponível em: <https://engsoftmoderna.info/>. Acesso em: 25 maio 2021.

VÁZQUEZ-INGELMO, Andrea *et al.* Resultados preliminares tras tres años aplicando aprendizaje basado en proyectos en ingeniería del software. **Aprendizaje, Innovación y Cooperación Como Impulsores del Cambio Metodológico**, p. 692-697. 2019. Servicio de Publicaciones Universidad. <http://dx.doi.org/10.26754/cinaic.2019.0141>.