

Proposta e Avaliação de uma Metodologia de Aprendizagem Baseada em Projetos em Disciplinas de Engenharia de Software através de uma Sequência Didática

Title: Proposal and Evaluation of a Project-Based Learning Methodology in Software Engineering Disciplines through a Didactic Sequence

Cynthia Pinheiro Santiago
Instituto Federal de Educação,
Ciência e Tecnologia do Ceará,
campus Tianguá
ORCID: [0000-0003-4013-4751](https://orcid.org/0000-0003-4013-4751)
cynthia.pinheiro@ifce.edu.br

José Wally Mendonça Menezes
Instituto Federal de Educação,
Ciência e Tecnologia do Ceará,
campus Fortaleza
ORCID: [0000-0003-2605-8633](https://orcid.org/0000-0003-2605-8633)
wally@ifce.edu.br

Francisco José Alves de Aquino
Instituto Federal de Educação,
Ciência e Tecnologia do Ceará,
campus Fortaleza
ORCID: [0000-0003-2963-3250](https://orcid.org/0000-0003-2963-3250)
fcoalves_aq@ifce.edu.br

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 em Engenharia de Computação e Ciência da Computação, mais especificamente na área de Engenharia de Software. Enquanto por um lado, as aulas tradicionais possuem um denso conteúdo teórico-conceitual e são apresentadas de forma expositiva, por outro lado a falta de experiências práticas torna desafiador para os alunos adquirir as habilidades necessárias para o desenvolvimento de projetos reais de software. Como forma de minimizar este problema, neste artigo apresentamos a aplicação da metodologia de Aprendizagem Baseada em Projetos em uma disciplina de Engenharia de Software, de forma que os estudantes, ainda durante sua formação acadêmica, possam ter uma experiência real em um projeto de desenvolvimento de software, com métodos, tecnologias e ferramentas comumente utilizados na indústria. Esta experiência, avaliada em uma turma regular de Engenharia de Software de um Bacharelado em Ciência da Computação, foi conduzida através de uma sequência didática concebida através de ciclos de Design Science Research, com objetivos educacionais bem definidos segundo a taxonomia de Bloom. Resultados preliminares mostram cenários promissores em relação ao ganho na aprendizagem, à aquisição de habilidades e à motivação dos estudantes com a metodologia proposta.

Palavras-Chave: Aprendizagem baseada em projetos; Engenharia de Software; Sequência didática; Taxonomia de Bloom.

Abstract

Recently, a gap has been noticed between teaching in universities and the needs of the industry in relation to the training of future professionals in Computer Engineering and Computer Science, more specifically in Software Engineering. While, on the one hand, traditional classes have a dense theoretical-conceptual content and are presented in an expository way, on the other hand, the lack of practical experiences makes it challenging for students to acquire the necessary skills for the development of real software projects. As a way to minimize this problem, in this article we present the application of a Project-Based Learning methodology in a Software Engineering discipline, so that students, even during their academic training, can have a real experience in a software project, with methods, technologies and tools commonly used in the industry. This experience, evaluated in a regular Software Engineering class of a Bachelor's Degree in Computer Science, was conducted through a didactic sequence conceived through Design Science Research cycles, with well-defined educational objectives according to Bloom's taxonomy. Preliminary results show promising scenarios in relation to learning gains, skills acquisition and student motivation with the proposed methodology.

Keywords: Project-based learning; Software Engineering; Didactic sequence; Bloom's Taxonomy.

Cite as: Santiago, C. P., Menezes, J. W. M. & Aquino, F. J. A. (2003). 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. DOI: 10.5753/rbie.2023.2817

1 Introdução

Atualmente, 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 diminuir esta lacuna 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*. Nesse sentido, a aprendizagem baseada em projetos (ABPj) é uma metodologia ativa que vem ganhando destaque em disciplinas de Engenharia de *Software* (ES) e que, por sua vez, promove uma aprendizagem mais significativa aos discentes. Na metodologia ABPj, os alunos recebem especificações sobre um produto e são incentivados a desenvolvê-lo, de acordo com procedimentos bem definidos, dentro do contexto de um projeto.

Embora atualmente existam muitas propostas de utilização de ABPj em disciplinas de ES, até o momento, nenhuma delas considerou o uso de uma Sequência Didática (SD), para estruturar as intervenções em sala de aula, juntamente com a taxonomia de Bloom para a definição dos objetivos educacionais, que é a proposta deste trabalho.

Portanto, neste artigo temos como objetivo apresentar a aplicação da ABPj em uma disciplina de ES através de uma SD – um artefato, desenvolvido através do método *Design Science Research* - que se baseia nos objetivos educacionais da taxonomia de Bloom. Outro objetivo é compreender a percepção dos alunos sobre o uso desta metodologia em sala de aula, através de uma pesquisa do tipo *survey*. Resultados preliminares mostram cenários promissores em relação ao ganho na aprendizagem e à motivação dos alunos com a metodologia proposta.

O restante deste artigo está organizado como se segue. Na Seção 2, é descrito o referencial teórico com o objetivo de aprofundar sobre o ensino de ES, a metodologia ABPj e a taxonomia de Bloom; na Seção 3, listamos alguns trabalhos relacionados a este artigo, com a justificativa para a proposição da presente metodologia; na Seção 4, apresentamos o desenho da pesquisa, com os seus objetivos, o método utilizado para a elaboração da SD, assim como a definição do instrumento de coleta de dados utilizado; na Seção 5, detalhamos as etapas da aplicação da metodologia ABPj através da SD proposta; na Seção 6, analisamos e discutimos os resultados obtidos e, por fim, na Seção 7, apresentamos algumas considerações finais, assim como as intenções de trabalhos futuros.

2 Referencial Teórico

2.1 Ensino de Engenharia de *Software* (ES)

A sociedade passa por um avanço crescente no surgimento de novas tecnologias. Este fato torna necessário que novas estratégias de ensino sejam incorporadas à sala de aula, objetivando a busca e aquisição de conhecimentos (Alves & Matos, 2017). Além disso, a formação qualificada e a capacitação de profissionais são pontos cada vez mais necessários na sociedade em que vivemos: formar bons profissionais faz parte do compromisso das Instituições de Ensino Superior com a sociedade (Prikladnicki, Albuquerque, von Wangenheim, & Cabral, 2009).

Especificamente no ensino de Engenharia de *Software* (ES), a qualidade dos profissionais está diretamente relacionada à qualidade da educação, uma vez que esta pode contribuir significativamente para a melhoria do estado da arte do desenvolvimento de *software* e auxiliar na solução de alguns problemas tradicionais e crises relacionadas com as práticas da indústria (Prikladnicki et al, 2009).

Por definição, a ES é uma área que se dedica à aplicação de teoria, conhecimento e prática para o desenvolvimento efetivo e eficiente de sistemas de *software* que satisfaçam os requisitos dos usuários (ACM/IEEE, 2014). A ES também 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).

Como disciplina, a ES tem seu currículo baseado no livro SWEBOK (*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 SWEBOK 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 SWEBOK representa as capacidades dos alunos e de engenheiros de *software*, que são adquiridas tanto por educação formal como por experiência (Oguz & Oguz, 2019). Normalmente, o conteúdo do SWEBOK é apresentado nas disciplinas de ES com um denso conteúdo teórico-conceitual e ensinado de forma tradicional, com aulas expositivas e leituras complementares (Cunha, Marques, Lemos, Câmara, & Vasconcellos, 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, Medeiros, & Prates, 2021). Portanto, é importante que o ensino seja prazeroso e englobe situações reais e dinâmicas para envolver o discente (Hoed, 2016).

O uso de metodologias ativas (MA) no ensino de ES envolve cenários que oferecem aos alunos uma aprendizagem mais significativa, aproximando a academia e a indústria a fim de estimular competências transversais e habilidades analíticas por meio de uma formação humanística, crítica e reflexiva. Ao serem aplicadas de modo assertivo no ensino de ES, as MA apresentam indicativos de que as habilidades cognitivas, interpessoais e intrapessoais são obtidas entre os discentes, além de permitir que estes retenham mais conhecimentos, surgindo como um recurso pedagógico inovador (Lima & Santos, 2021).

As MA representam uma abordagem apoiada na interação de conhecimentos, análises e decisões, onde o docente age como mediador e o discente como gestor do conhecimento. Estas estratégias estimulam o uso de funções mentais de pensar, raciocinar, refletir, entender, dentre outras. As mesmas também provocam a aprendizagem ativa, a qual é considerada como uma aprendizagem a ser seguida no século XXI, além de desenvolver habilidades que são exigidas pela indústria.

De acordo com Lima, Alencar e Santos (2020), entre as MA mais utilizadas em ES estão: Aprendizagem Baseada em Design; Mapas Mentais e Conceituais; Aprendizagem entre Pares; Aprendizagem Baseada em Equipes; Gamificação; Estudos de Casos; Sala de Aula Invertida;

Discussões em Classe; Aprendizagem Baseada em Problemas e Aprendizagem Baseada em Projetos, sendo este último o método de ensino mais utilizado em ES (Anicic, & Stapic, 2022).

2.2 Aprendizagem Baseada em Projetos (ABPj)

O desenvolvimento da Aprendizagem Baseada em Projetos (ABPj) teve suas origens em 1900, quando o filósofo americano John Dewey comprovou o “aprender mediante o fazer”, valorizando, questionando e contextualizando a capacidade de pensar dos alunos. Isto era feito numa forma gradativa de aquisição de conhecimento, através da execução de projetos para resolução de problemas reais referentes aos conteúdos na área de estudos. Esta prática tinha como meta o desenvolvimento dos estudantes no aspecto físico, emocional e intelectual, por meio de métodos experimentais (Masson, Miranda, Munhoz, & Castanheira, 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, Sena, Araújo, & Akamatsu, 2010). Este é um dos métodos educacionais mais amplamente utilizados em cursos de Ciência da Computação, Sistemas de Informação e Engenharia (Souza, Moreira, & Figueiredo, 2019).

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 e 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 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 conseqüentemente 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 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.

O próprio currículo da ACM/IEEE (ACM/IEEE, 2014) considera que *capstone projects* são elementos essenciais em um programa de graduação em ES. Segundo Prikladnicki et al (2009), um *capstone project* é um esforço em grupo em que alunos executam um projeto do início ao fim. As principais características destes projetos são: (i) o projeto deve abranger um período letivo completo, dando aos alunos tempo adequado para refletir sobre as experiências e tentar novas soluções se necessário; (ii) sempre que possível, deve ser realizado como um projeto de grupo; (iii) o projeto deve ter alguma forma de implementação como sua entrega final para que os alunos possam experimentar um amplo conjunto de atividades de desenvolvimento de *software* e avaliar adequadamente essas experiências e (iv) a avaliação dos resultados do projeto deve ir além da implementação do conceito, avaliando a eficácia com que as práticas e processos de ES foram empregados, incluindo a qualidade da reflexão do aluno sobre a experiência (ACM/IEEE, 2014).

No entanto, não é todo projeto que pode ser considerado um projeto segundo a ABPj. A fim de compreender as distinções entre tarefas de projetos tradicionais e a abordagem de ensino na ABPj, Bender (2015) listou os aspectos comuns ou características essenciais que devem constar nos projetos que seguem a ABPj, representados na Figura 1:

- **Âncora:** introdução e informações básicas sobre o projeto, para preparar o terreno e gerar o interesse dos alunos;
- **Questão motriz:** pergunta que deve chamar a atenção dos alunos, bem como focar seus esforços;
- **Trabalho cooperativo em equipe:** crucial para as experiências de ABPj, como forma de tornar as experiências de aprendizagem mais autênticas.
- **Feedback e revisão:** a assistência estruturada deve ser rotineiramente proporcionada e pode ser baseada nas avaliações do professor ou dos colegas;
- **Investigação e inovação:** dentro da questão motriz, o grupo precisará gerar questões adicionais focadas mais especificamente nas tarefas do projeto;
- **Oportunidade para reflexão:** criar oportunidades para a reflexão dos alunos dentro de vários projetos é um aspecto a ser enfatizado;
- **Diretrizes para a conclusão:** para estruturar a atividade, podem-se utilizar diretrizes para a conclusão do projeto e para a geração de artefatos ou, alternativamente, o próprio grupo pode desenvolver linhas de tempo e metas específicas para a conclusão de partes do projeto;
- **Voz e escolha do aluno:** os alunos devem ter voz em relação a alguns aspectos de como o projeto pode ser realizado, além de serem encorajados a fazer escolhas ao longo de sua execução;
- **Resultados apresentados publicamente:** os projetos de ABPj pretendem ser exemplos autênticos dos tipos de problemas que os alunos enfrentam no mundo real, de modo que é fundamental algum tipo de apresentação pública dos resultados do projeto.



Figura 1: Características essenciais em projetos que seguem a ABPj (Bender, 2015)

Nesse contexto, destaca-se a importância da delimitação dos objetivos de aprendizagem, que é produto da reflexão sobre as habilidades, competências e conhecimentos que se espera que o estudante tenha obtido ao final da prática educacional. Um dos instrumentos existentes que pode vir a facilitar esse processo é a taxonomia proposta por Bloom (1956), que tem explicitamente como objetivo ajudar no planejamento, organização e controle dos objetivos de aprendizagem (Carvalho, Junqueira, & Furlanetto, 2017).

2.3 Taxonomia de Bloom

Por bastante tempo, acreditou-se que a razão pela qual alguns discentes obtinham desempenho melhor do que outros estava relacionada a variáveis existentes fora do ambiente educacional e que, nas mesmas condições de aprendizagem, todos aprenderiam com a mesma competência e profundidade (Ferraz & Belhot, 2010). Entretanto, Bloom, Madaus e Hastings (1971) descobriram que nas mesmas condições de ensino todos os alunos aprendiam, mas havia diferenças em relação ao nível de profundidade e abstração do conhecimento adquirido.

O objetivo do processo de aprendizagem é obter um maior nível de profundidade em relação ao conhecimento. Bloom (1956) entende que a simples acumulação de conhecimento é distinta do aumento da quantidade de informação lembrada pelo estudante. Em vez disso, ele sugere que a aprendizagem envolve diferentes níveis de conhecimento pelos quais o aluno deve transitar (Ferraz & Belhot, 2010). Para demonstrar isso, Bloom desenvolveu uma taxonomia, com o objetivo de proporcionar uma ferramenta coerente com as características dos processos mentais superiores (Ferraz & Belhot, 2010).

Originalmente, a taxonomia de Bloom (ou taxonomia de objetos educacionais), tem 11 níveis, que são categorizados em três diferentes domínios. O domínio cognitivo, no entanto, foi o que mais recebeu atenção de Bloom e, conseqüentemente, tem sido o mais aplicado pelos educadores (Britto & Usman, 2015). Neste domínio, as competências estão relacionadas à forma como as pessoas recordam o conhecimento, como compreendem e pensam criticamente sobre um determinado tópico. Também envolve a aquisição de um novo conhecimento, do desenvolvimento intelectual, de habilidades e de atitudes, bem como inclui o reconhecimento de fatos específicos, procedimentos padrão e conceitos que estimulam o desenvolvimento intelectual constantemente (Bloom, 1956; Ferraz & Belhot, 2010; Britto & Usman, 2015).

Os níveis do domínio cognitivo são os seguintes: o primeiro e mais básico nível é o “Conhecimento”, ou seja, dados ou informações que se podem recordar. Esse conhecimento deve ser “compreendido”, o que implica em atingir o segundo nível que é a “Compreensão”, onde ocorre o entendimento do significado e da interpretação de instruções e problemas envolvendo a informação. O terceiro nível de conhecimento é a “Aplicação”, que envolve o uso de um conceito em uma nova situação ou o uso espontâneo de uma abstração. O quarto nível é a “Análise”, ou seja, a separação de conceitos ou habilidades em partes na tentativa de entender sua estrutura organizacional. Além de ter habilidades analíticas, os alunos devem ser capazes de se envolver em uma “Síntese”, que corresponde ao quinto nível, construindo uma estrutura ou padrão a partir de diversos elementos. O sexto e último nível de conhecimento é a “Avaliação”, relacionada à capacidade dos alunos de fazer julgamentos de valor sobre ideias ou materiais. A cada um dos níveis foi associado um conjunto de ações (ou verbos) que auxiliam na classificação de uma questão em um dos níveis da taxonomia (Fu, Wu, & Ho, 2009).

Nesta versão original da taxonomia de Bloom, os níveis são organizados por graus de dificuldade, com cada nível seguinte sendo atingível apenas quando os níveis mais baixos e mais fáceis forem dominados (Fu et al, 2009), conforme pode ser visto na Figura 2.

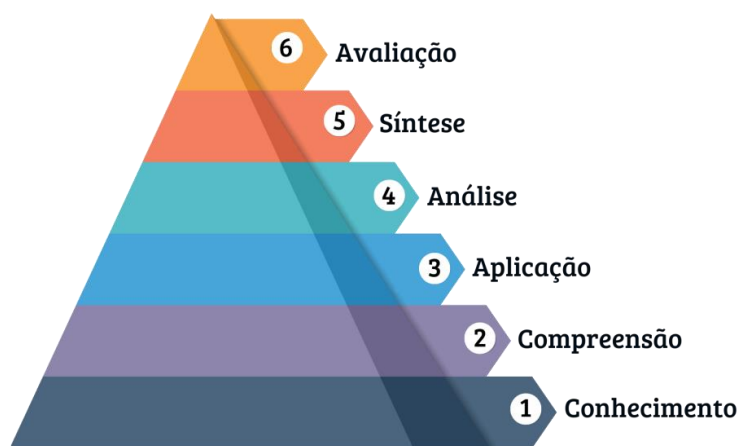


Figura 2: Categorias da Taxonomia de Bloom

Krathwohl (2002) percebeu que mudanças na taxonomia original seriam necessárias em relação aos verbos de cada nível e sua associação direta com o objetivo cognitivo, avaliação do objetivo e desenvolvimento de competências. Ao analisar a relação direta entre verbo e substantivo, chegou à conclusão de que verbos e substantivos deveriam pertencer a dimensões separadas, nas quais os substantivos formariam a base para a dimensão Conhecimento (o quê) e os verbos seriam utilizados para a dimensão relacionada aos aspectos cognitivos (como). Essa separação de substantivos e verbos, conhecimento e aspectos cognitivos, deu um caráter bidimensional à taxonomia original (Ferraz & Belhot, 2010). As partes da estrutura bidimensional foi nominada como Dimensão Conhecimento e Dimensão Processo Cognitivo (Figura 3).

A dimensão Processo Cognitivo abrange os cinco níveis da taxonomia original, porém estes foram renomeados; em alguns casos, apenas para suas formas verbais. A categoria Conhecimento tornou-se Lembrar; Compreensão tornou-se Entender; Síntese tornou-se Criar (e foi promovida para a categoria mais alta da hierarquia); Aplicação, Análise e Avaliação tornaram-se respectivamente Aplicar, Analisar e Avaliar (Galhardi & Azevedo, 2013). Os verbos associados a cada um dos níveis da taxonomia de Bloom revisada por Krathwohl (2002) são apresentados no Quadro 1.

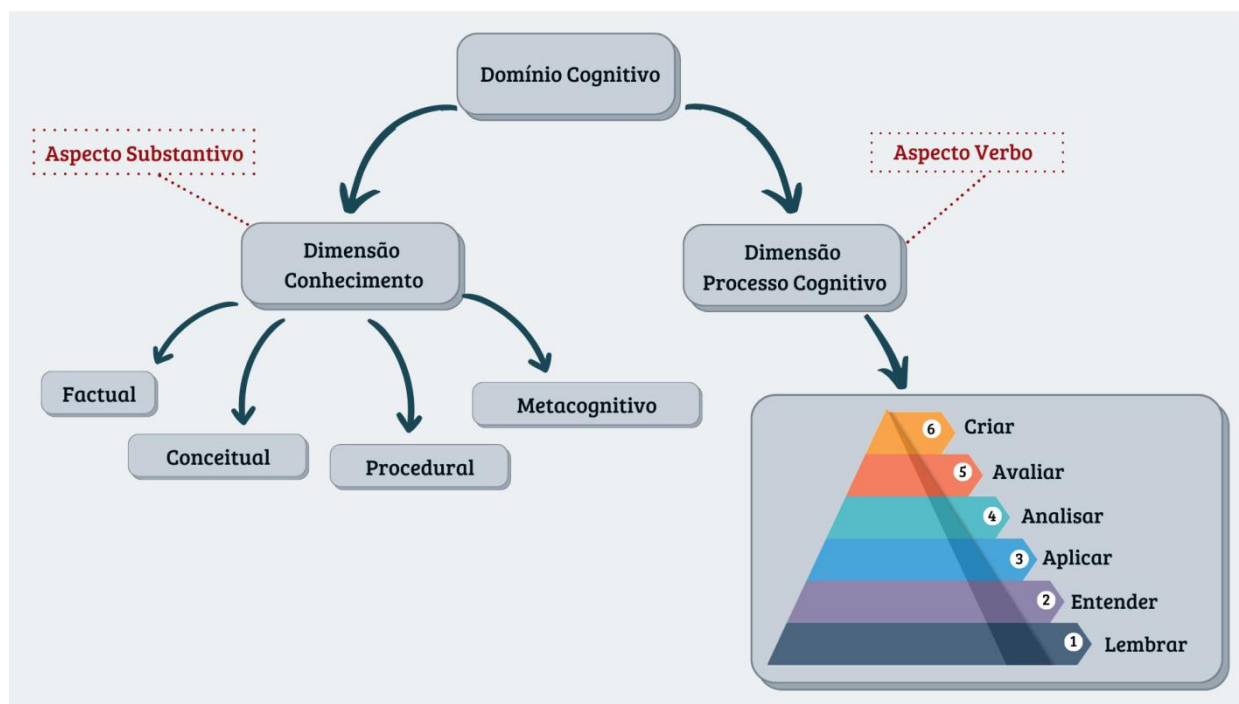


Figura 3 - Dimensões do Domínio Cognitivo da Taxonomia de Bloom revisada por Krathwohl (2002)

Quadro 1: Níveis da taxonomia de Bloom revisada por Krathwohl (2002) e seus respectivos verbos

| Categoria | Verbos |
|--------------------|---|
| 1. Lembrar | Enumerar, definir, descrever, identificar, denominar, listar, nomear, combinar, realçar, apontar, relembra, recordar, relacionar, reproduzir, solucionar, declarar, distinguir, rotular, memorizar, ordenar e reconhecer. |
| 2. Entender | Alterar, construir, converter, decodificar, defender, definir, descrever, distinguir, discriminar, estimar, explicar, generalizar, dar exemplos, ilustrar, inferir, reformular, prever, reescrever, resolver, resumir, classificar, discutir, identificar, interpretar, reconhecer, redefinir, selecionar, situar e traduzir. |
| 3. Aplicar | Aplicar, alterar, programar, demonstrar, desenvolver, descobrir, dramatizar, empregar, ilustrar, interpretar, manipular, modificar, operacionalizar, organizar, prever, preparar, produzir, relatar, resolver, transferir, usar, construir, esboçar, escolher, escrever, operar e praticar. |
| 4. Analisar | Analisar, reduzir, classificar, comparar, contrastar, determinar, deduzir, diagramar, distinguir, diferenciar, identificar, ilustrar, apontar, inferir, relacionar, selecionar, separar, subdividir, calcular, discriminar, examinar, experimentar, testar, esquematizar e questionar. |
| 6. Avaliar | Avaliar, averiguar, escolher, comparar, concluir, contrastar, criticar, decidir, defender, discriminar, explicar, interpretar, justificar, relatar, resolver, resumir, apoiar, validar, escrever um <i>review</i> sobre, detectar, estimar, julgar e selecionar. |
| 5. Criar | Categorizar, combinar, compilar, compor, conceber, construir, criar, desenhar, elaborar, estabelecer, explicar, formular, generalizar, inventar, modificar, organizar, originar, planejar, propor, reorganizar, relacionar, revisar, reescrever, resumir, sistematizar, escrever, desenvolver, estruturar, montar e projetar. |

A taxonomia de Bloom é provavelmente a taxonomia educacional mais utilizada para especificar os objetivos de aprendizagem em Ciência da Computação (Masapanta-Carrión & Velázquez-Iturbide, 2017), sendo recomendada, inclusive, no currículo do ACM/IEEE (ACM/IEEE, 2014). Segundo Britto e Usman (2015), a taxonomia de Bloom tem sido aplicada

também à ES que, por sua vez, possui particularidades que tornam o processo de ensino e aprendizagem ainda mais desafiador. No mapeamento sistemático realizado por estes autores, observou-se que, na maioria dos trabalhos revisados, foram utilizados todos os níveis cognitivos da taxonomia, tanto para o *design* quanto para a avaliação de disciplinas de ES.

2.4 Sequência Didática

Sequências didáticas (SD) são “um conjunto de atividades ordenadas, estruturadas e articuladas para a realização de certos objetivos educacionais, que têm princípio e fim conhecidos tanto pelos professores como pelos alunos” (Zabala, 2015, p.24).

Dolz, Noverraz e Schneuwly (2004) definem uma SD como um método de organização linear do conteúdo temático, com síntese do objetivo prático, destinada a orientar as intervenções dos professores com o intuito de tornar mais eficiente o processo de aprendizagem. A partir da elaboração de uma SD, professores e alunos seguem um roteiro bem definido, que contemple objetivos, procedimentos e formas de avaliação claras e que lhes permita gerenciar melhor seu aprendizado sobre os conteúdos abordados (Henrique, Farias, Cunha, & Scaico, 2013).

Segundo Dolz et al (2004), a estrutura de base de uma SD contém inicialmente uma seção de abertura, com a apresentação da situação de estudo na qual é descrita de maneira detalhada a tarefa que os alunos deverão realizar. Em seguida, deve haver uma produção inicial ou diagnóstica, a partir da qual o professor avalia as capacidades já adquiridas e ajusta as atividades e os exercícios previstos na sequência às possibilidades e dificuldades reais da turma. Após esta etapa, o trabalho se concentra nos módulos, constituídos de várias atividades ou exercícios sistemáticos e progressivos. A produção final, segundo os autores, é o momento de os alunos porem em prática os conhecimentos adquiridos e de o professor avaliar os progressos efetivados, servindo esse momento, também, para uma avaliação do tipo somativa (Araújo, 2013). Esta representação do processo envolvido na sequência didática está descrita na Figura 4.

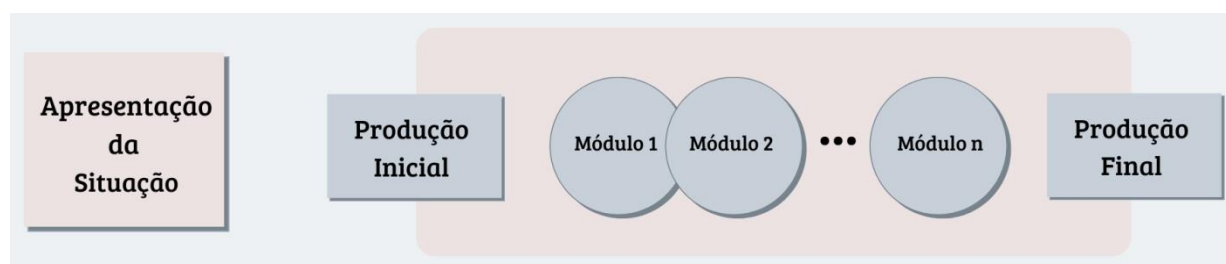


Figura 4: Esquema de uma SD, proposta por Dolz et al (2004)

A produção desse artefato conduz o professor a um planejamento de ensino que busque minimizar apontamentos considerados por ele como dificuldades de ensino, como é o caso da má gestão do tempo das aulas, o que pode gerar o não cumprimento de todo conteúdo programado. Além disso, a partir da elaboração de uma SD, professor e alunos seguem um roteiro de aprendizado bem definido, que contemple objetivos, procedimentos e formas de avaliação claras e que lhes permita gerenciar melhor seu aprendizado sobre os conteúdos abordados (Henrique et al, 2013).

Oliveira e Lavor (2022) destacam a importância de a SD ser elaborada na perspectiva de realizações de seções ou conjunto de aulas que propiciem a investigação do processo de ensino e aprendizagem. Isso permite ao professor fazer um paralelo com a elaboração de um plano de aula, porém, diferencia-se por ser mais amplo e por se utilizar vários métodos. Além disso, o

docente deve agir como mediador, possibilitando aos discentes uma interatividade significativa. Nesse sentido, as situações didáticas além de subsidiarem investigações de ensino, podem possibilitar que pesquisas apontem o caminho para se chegar ao pleno desenvolvimento dos estudantes.

Dolz et al (2004) ressaltam ainda que, na medida do possível, as SD devem ser realizadas no âmbito de um projeto de classe, elaborado durante a apresentação da situação, pois este torna as atividades de aprendizagem significativas e pertinentes. Esse projeto pode ser parcialmente fictício, à medida que a motivação pode nascer mais diretamente do desejo de progredir, de adquirir novas capacidades.

3 Trabalhos Relacionados

Nesta seção discutimos alguns trabalhos relacionados que promovem a discussão e análise de experiências no uso de ABPj em cursos relacionados à ES. A maioria dos estudos concentra-se em descrever uma proposta de configuração de ABPj e, em seguida, avaliar essas abordagens.

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 e o 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, Serrano e Sales (2021) relatam a experiência sobre o uso da ABPj na disciplina de Requisitos de *Software*. 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. 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. Aos alunos foi proposto 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 proporcionam aprendizagem em diferentes esferas.

Vázquez-Ingelmo, García-Holgado, García-Peñalvo e Rodríguez-Conde (2019) relatam a experiência de três anos aplicando ABPj em uma disciplina de ES. No início da disciplina, é definido o contexto do projeto e uma série de objetivos gerais que o sistema a ser modelado deve atender. Os alunos 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, entregues ao longo do curso. Os professores desempenham o papel de clientes, fornecendo *feedbacks* semanais a cada grupo.

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 na disciplina

para obter vantagens como ter acesso a alunos para fins de recrutamento e benefícios diretos ou indiretos dos resultados do projeto. Os alunos se envolvem em todas as fases do projeto: planejamento, execução e encerramento. Na fase de execução, os projetos costumam ser divididos em módulos com entregas frequentes às partes interessadas. Os projetos dos alunos devem demonstrar a execução de atividades de ES (ou seja, elicitação de requisitos, design de sistema, codificação e teste).

O trabalho de Alves e Matos (2017) tem como objetivo apresentar a concepção e a análise da aplicação de uma SD desenvolvida para auxiliar no ensino de conceitos elementares da unidade curricular Engenharia de Software de um curso técnico integrado em Informática. A SD proporcionou ao professor a organização, a transparência e a democratização no processo de ensino dos conteúdos da unidade curricular e, aos alunos, a clareza de como seriam avaliados e classificados durante a sequência de atividades que teve como objetivo relacionar teoria e prática.

Embora estes recentes trabalhos incluam tópicos como uso de ABPj, ensino de ES ou áreas afins, aplicação de SD e/ou utilização de *capstone projects* (Quadro 2) segundo Sales, Serrano e Serrano (2020), na literatura atual não existem trabalhos que forneçam uma orientação sistemática para operacionalizar a ABPj, através de uma SD, em nível de graduação.

Quadro 2: Trabalhos relacionados

| | Uso de ABPj | Ensino de ES | Aplicação de SD | <i>Capstone project</i> |
|--|-------------|--------------|-----------------|-------------------------|
| Serrano, Serrano e Sales (2021) | X | X | | |
| Uzun, Pugliesi e Roland (2018) | X | X | | |
| Vázquez-Ingelmo, García-Holgado, García-Peñalvo e Rodríguez-Conde (2019) | X | X | | X |
| Gupta e Nguyen-Duc (2021) | X | X | | X |
| Alves e Matos (2017) | | X | X | |

É para preencher esta lacuna que propomos o presente trabalho, que apresenta uma SD para estruturar as atividades e intervenções na disciplina de ES com o método ABPj, através de um *capstone project*, tendo em vista os objetivos educacionais da taxonomia revisada de Bloom.

4 Desenho da Pesquisa

4.1 Objetivos

O objetivo deste estudo é propor e avaliar a aplicação da ABPj - através de uma SD - em uma disciplina de ES, seguindo os objetivos educacionais da taxonomia revisada de Bloom, buscando compreender a percepção dos alunos sobre o uso desta metodologia em sala de aula. Para atingir estes objetivos, propomos as seguintes questões de pesquisa (QP):

- **QP1.** Qual é a percepção dos alunos sobre o uso de aulas expositivas tradicionais e exames no ensino de ES?
- **QP2.** Como os alunos percebem a contribuição de um trabalho na forma de um projeto para o aprendizado de tópicos específicos de ES?
- **QP3.** Quais são os aspectos positivos observados pelos alunos em um curso com ABPj?

- **QP4.** Quais são os aspectos negativos que os alunos observaram em um curso com ABPj?

Para responder a estas questões, a metodologia em questão foi aplicada, durante o primeiro semestre de 2022, a um grupo de 23 estudantes do curso de Bacharelado em Ciência da Computação do Instituto Federal de Educação, Ciência e Tecnologia do Ceará, campus Tianguá, matriculados na disciplina de ES. A intervenção se deu ao longo do semestre durante 20 semanas, totalizando 40 horas-aula de 50 minutos cada. O estudo realizado teve como ferramenta a SD descrita na Seção 5, aplicada em paralelo às aulas expositivas previstas.

4.2 Método de Elaboração da Sequência Didática

Um dos paradigmas utilizados para elaboração e validação de uma SD é o Design Research (DR). Os objetivos do DR são o estudo, pesquisa e investigação do artificial feito por seres humanos, e a forma como essas atividades são direcionadas (Bayazit, 2004). Uma das principais vantagens do DR no contexto educacional é que ele exige que profissionais e pesquisadores colaborem na identificação de problemas reais de ensino e aprendizagem, na criação de soluções com base em princípios de *design* existentes e no teste e refinamento tanto das soluções quanto dos princípios de *design* até que resultados satisfatórios tenham sido alcançados por todos os envolvidos (Reeves, 2006).

Para operacionalizar a pesquisa em DR, é necessário um método que implemente este paradigma, como é o caso do *Design Science Research* (DSR). O DSR é um método de pesquisa que se dedica ao desenvolvimento de estudos que tenham como objetivo a prescrição, o projeto e a construção de artefatos para a solução de um problema (Dresch, Lacerda, & Miguel, 2015). Trata-se de um processo iterativo, dividido em três ciclos: (i) relevância (investigação do problema); (ii) *design* (desenvolvimento e/ou evolução do artefato); e (iii) rigor (validação da solução). O ciclo de design pode ser realizado quantas vezes forem necessárias para que possíveis refinamentos na solução sejam efetuados (Horita, Graciano, & Santos, 2018). Assim, as etapas contidas para elaboração da SD deste trabalho seguiram os processos indicados na Figura 5.

A etapa “Identificação do Problema” buscou delinear o problema a ser tratado, investigando quais elementos deveriam ser melhorados e por que. Para isso, uma revisão da literatura foi conduzida anteriormente como parte desta pesquisa. Por meio da análise dos trabalhos encontrados, verificou-se que embora houvesse muitas propostas de utilização de ABPj em disciplinas relacionadas a ES, nenhuma delas considerou o uso de uma SD, para estruturar as intervenções em sala de aula, ou a taxonomia revisada de Bloom para a definição dos objetivos educacionais, que é a proposta deste trabalho.

A etapa “Planejamento do Artefato” envolveu estratégias para apoiar a preparação, desenvolvimento e avaliação do artefato proposto. Para tanto houve um *brainstorming* com um grupo de três professores que já ministraram disciplinas relacionada a ES. Este *brainstorming* foi conduzido de forma a obter uma base para a pesquisa e gerar novos direcionamentos. De início, um dos membros do grupo de pesquisa relacionado a este trabalho atuou como facilitador e realizou uma apresentação com as informações sobre o contexto do estudo para que os demais professores pudessem contribuir com suas ideias. Com isso, os participantes trocaram informações de forma mútua, esclarecendo dúvidas, arguições e questionamentos de modo a analisar o nível de adequação da proposta, a qual foi avaliada como adequada e/ou muito adequada pelos participantes.

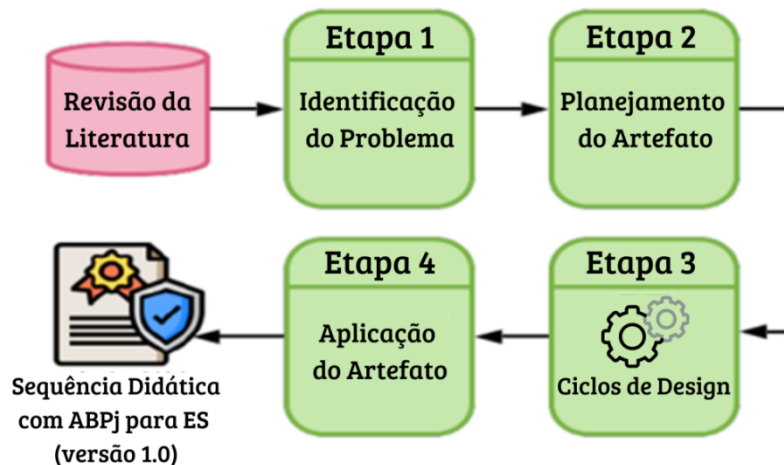


Figura 5: Etapas realizadas para a elaboração do artefato SD para a disciplina de ES

Com base nas informações validadas durante o *brainstorming*, a etapa “Ciclo de Design” resultou na definição dos módulos da SD e suas respectivas tarefas, ferramentas a serem utilizadas pelos alunos, produtos de trabalho esperados, objetivos educacionais a serem alcançados e rubricas de avaliação. Esta etapa foi realizada ao longo de três ciclos, onde cada ciclo de prototipação da SD ocorreu em um semestre no período compreendido entre janeiro de 2021 e julho de 2022. O resultado desta etapa é detalhado na Seção 5 deste trabalho.

A etapa “Aplicação do Artefato” visa avaliar a aplicação desta SD em uma turma regular da disciplina de ES, sob a forma de uma prova de conceito, ministrada por um dos pesquisadores deste trabalho. O estudo foi realizado durante o primeiro semestre de 2022, com um grupo de alunos do quinto semestre do curso de Bacharelado em Ciência da Computação, na unidade curricular Engenharia de *Software*.

4.3 Método de Pesquisa

O referido estudo caracteriza-se por uma pesquisa do tipo descritiva, com uma abordagem qualitativa. A pesquisa descritiva visa registrar e analisar fatos ou fenômenos sem manipulá-los, procurando descobrir a frequência com que um fenômeno ocorre, sua relação e conexão com outros, sua natureza e características (Gil, 2008).

O instrumento de coleta de dados adotado neste trabalho é o *survey*. De acordo com Easterbrook, Singer, Storey e Damian (2008), os instrumentos do tipo *survey* são usados para identificar características de uma população e geralmente estão associados à aplicação de questionários. Tais questionários destinam-se a recolher dados para descrever, comparar ou explicar conhecimentos, atitudes e comportamentos (Pfleeger & Kitchenham, 2001).

Assim, foram definidos os seguintes passos para a execução desta pesquisa: (i) Definição dos objetivos do estudo e questões de pesquisa (Seção 4.1); (ii) Elaboração da SD (Seções 4.2 e 5); (iii) Criação do questionário (Seção 4.4) e (iv) Análise e Discussão dos resultados (Seção 6).

4.4 Questionário

O meio utilizado para compreender a percepção dos alunos sobre o uso da metodologia ABPj foi um questionário (Tabela 1), cuja estrutura baseou-se no instrumento desenvolvido por Souza et al (2019). O questionário está organizado em quatro seções (coluna “Seção”) e dez questões

(coluna “Pergunta”), rotuladas com identificadores únicos, Q1 a Q10 (coluna “ID”). As perguntas misturam respostas abertas e opções na escala Likert de cinco pontos.

Tabela 1: Estrutura do Questionário

| Seção | ID | Pergunta | Respostas |
|---------------------------------|-----|--|---|
| Informação sobre o participante | Q1 | Período Acadêmico | [1 a 10] |
| | Q2 | Equipe durante o projeto | Resposta Aberta |
| | Q3 | Este foi seu primeiro contato com a disciplina de ES? Qual outro curso relacionado a ES você já participou? | [Sim/Não] e Resposta Aberta |
| | Q4 | Você teve alguma experiência com ES no mercado de trabalho? | |
| Método de Aprendizagem | Q5 | Avalie a afirmação: "Acredito que os trabalhos práticos, focados no desenvolvimento de um projeto de <i>software</i> , são fundamentais para o desenvolvimento de habilidades e/ou aprendizagem da ES". | Escala Likert: (1) Discordo totalmente; (2) Discordo parcialmente; (3) Indiferente; (4) Concordo parcialmente; (5) Concordo totalmente |
| | Q6 | Avalie a afirmação: "Acredito que as aulas expositivas tradicionais, com métodos pontuais de avaliação (exames e trabalhos específicos), são suficientes para o aprendizado da ES". | Escala Likert: (1) Discordo totalmente; (2) Discordo parcialmente; (3) Indiferente; (4) Concordo parcialmente; (5) Concordo totalmente |
| Tópicos de Aprendizagem | Q7 | Avalie o quanto a atribuição do Projeto de <i>Software</i> contribuiu para o desenvolvimento de habilidades/aprendizado dos seguintes tópicos: a. Requisitos de <i>software</i> b. Métodos Ágeis c. Interface com o Usuário d. Gestão de Configuração e. Implementação de <i>software</i> f. Testes de <i>Software</i> | Escala Nominal: (1) Nada; (2) Muito pouco; (3) Razoavelmente; (4) Consideravelmente; (5) Totalmente |
| | Q8 | Avalie o quanto a atribuição do Projeto de <i>Software</i> lhe proporcionou as seguintes experiências: a. Trabalho Cooperativo b. <i>Feedback</i> e Revisão c. Investigação e Inovação d. Oportunidade para Reflexão e. Diretrizes para a conclusão f. Voz e Escolha do Aluno | Escala Nominal: (1) Nada; (2) Muito pouco; (3) Razoavelmente; (4) Consideravelmente; (5) Totalmente |
| Aspectos Positivos e Negativos | Q9 | Quais são os aspectos positivos do Projeto de <i>Software</i> como trabalho prático? | Resposta Aberta |
| | Q10 | Quais são os aspectos negativos do Projeto de <i>Software</i> como trabalho prático? | Resposta Aberta |

A primeira seção (Informação sobre o Participante) é composta por perguntas sobre a experiência dos participantes. A segunda (Método de Aprendizagem) é composta por questões destinadas a avaliar a percepção dos alunos sobre o uso de projetos práticos de desenvolvimento no curso e sobre o uso de aulas tradicionais. A terceira parte (Tópicos de Aprendizagem) questiona os alunos sobre sua percepção sobre a contribuição do projeto na aprendizagem de tópicos específicos de ES e sobre a aquisição de experiências durante o projeto. A quarta

(“Aspectos positivos e negativos”) solicita que os participantes descrevam os aspectos positivos e negativos do uso da metodologia ABPj, segundo seu ponto de vista.

5 Proposta de SD com ABPj para a disciplina de Engenharia de *Software*

A presente SD foi desenvolvida para uma disciplina de ES de um curso de Bacharelado em Ciência da Computação. O planejamento para esta unidade curricular define que todo o conteúdo deve ser abordado em 20 semanas, perfazendo um semestre letivo. O principal objetivo da disciplina, por meio de aulas, atividades práticas e pesquisa, é fazer com que cada estudante, ao final do semestre, tenha adquirido o conhecimento necessário para analisar e discutir conceitos básicos de ES, seus métodos e ferramentas. Originalmente, a ementa desta disciplina compreende os temas, conteúdos, procedimentos e métodos avaliativos do Quadro 3.

Quadro 3: Temas tradicionalmente abordados na disciplina de ES

| Temas | Conteúdos Trabalhados | Procedimentos | Avaliação |
|-------------------------------|---|--|--|
| Modelos de Processos | <ul style="list-style-type: none"> • Introdução ao SWEBOK • Modelos de Processos Prescritivos • Processo Unificado e RUP • Modelos de Processos Ágeis | <ul style="list-style-type: none"> • Exposição oral dialogada • Exemplos de situações cotidianas | <ul style="list-style-type: none"> • Prova dissertativa • Seminários |
| Requisitos de <i>Software</i> | <ul style="list-style-type: none"> • Engenharia de Requisitos • Requisitos Funcionais e não funcionais • Técnicas de elicitação de Requisitos • Especificação de Requisitos | <ul style="list-style-type: none"> • Exposição oral dialogada • Exemplos de situações cotidianas | <ul style="list-style-type: none"> • Prova dissertativa • Seminários |
| Interface com o Usuário | <ul style="list-style-type: none"> • Interação Humano-Computador • Desenvolvimento de interfaces e Prototipagem • Estilos de interação | <ul style="list-style-type: none"> • Exposição oral dialogada • Exemplos de situações cotidianas | <ul style="list-style-type: none"> • Prova dissertativa • Seminários |
| Gestão de Configuração | <ul style="list-style-type: none"> • Itens de Configuração • Controle de Versões • Controle de Mudança • Integração Contínua e DevOps | <ul style="list-style-type: none"> • Exposição oral dialogada • Exemplos de situações cotidianas | <ul style="list-style-type: none"> • Prova dissertativa • Seminários |
| Testes de <i>Software</i> | <ul style="list-style-type: none"> • Processo de Testes • Testes de Unidade/Integração/Sistema • Testes de Sistema • Testes Automatizados e TDD | <ul style="list-style-type: none"> • Exposição oral dialogada • Exemplos de situações cotidianas | <ul style="list-style-type: none"> • Prova dissertativa • Seminários |
| Qualidade de <i>Software</i> | <ul style="list-style-type: none"> • Atributos de qualidade de <i>software</i> • Maturidade de Processo de <i>Software</i> • CMMI e MPS.BR | <ul style="list-style-type: none"> • Exposição oral dialogada • Exemplos de situações cotidianas | <ul style="list-style-type: none"> • Prova dissertativa • Seminários |

Em adição às aulas expositivas já previstas na ementa e como forma de inserir atividades práticas ao longo da disciplina, a presente SD foi proposta. Para a elaboração da SD, foi utilizado como referência, o esquema de atividades proposto por Dolz et al (2004), a metodologia ABPj definida por Bender (2015) e os objetivos educacionais da taxonomia revisada de Bloom (Krathwohl, 2002), compreendendo as seguintes etapas: a) Apresentação da situação; b) Desenvolvimento do projeto; c) Apresentação de Resultados. Estas etapas estão descritas na Figura 6 e serão mais bem detalhadas nas Subseções seguintes.

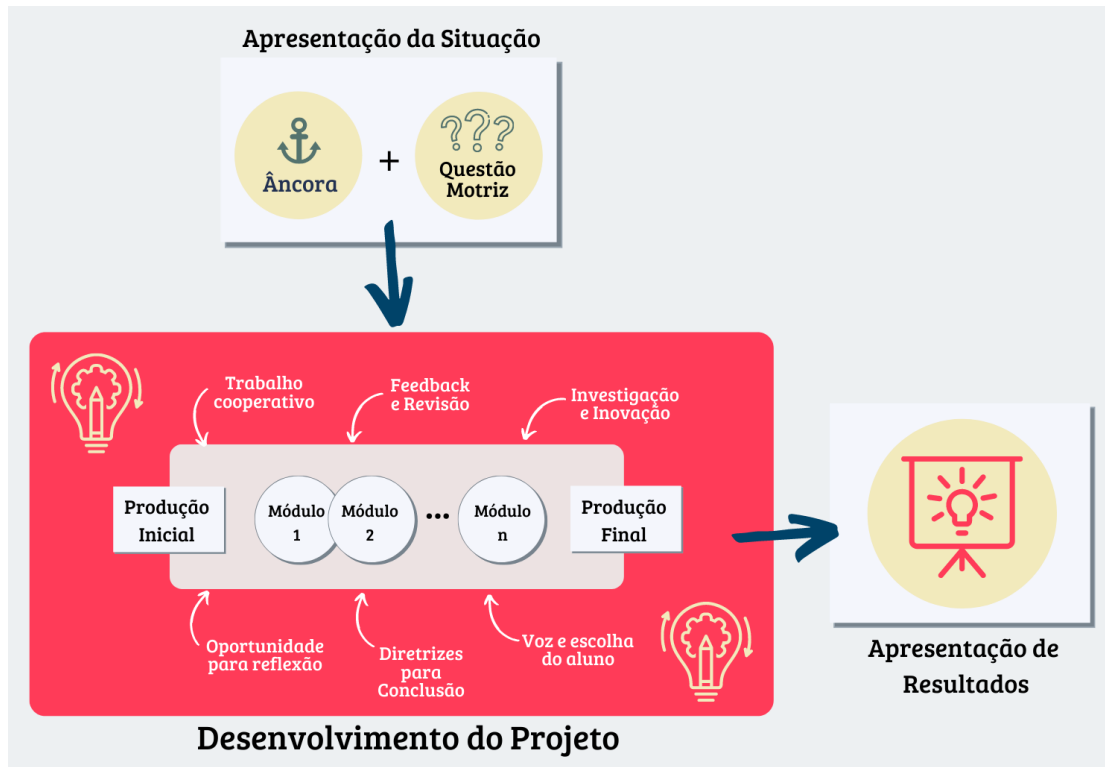


Figura 6: Esquema proposto para a SD com ABPj para a disciplina de ES

5.1 Apresentação da Situação

Segundo Dolz et al (2004), a apresentação da situação visa fornecer aos alunos todas as informações necessárias para que conheçam o projeto que será realizado, ao mesmo tempo em que os prepara para a produção inicial. É um momento em que é construída uma representação da situação a ser executada, onde duas dimensões podem ser distinguidas: a) apresentar um problema bem definido, onde é proposto um projeto coletivo para que os alunos compreendam da melhor forma possível a situação na qual devem atuar, e b) preparar os artefatos que serão produzidos, com a intenção de que os alunos percebam a importância destes artefatos e saibam com quais vão trabalhar (Figura 7).

Neste sentido, para a apresentação do problema é utilizada uma âncora, que consiste em uma introdução e informações básicas sobre o projeto, para preparar o terreno e gerar o interesse dos alunos. Segundo Bender (2015), a âncora é de extrema importância, pois deve proporcionar um bom motivo para que os alunos queiram realizar um projeto. Dessa forma, é importante que ela ilustre ou descreva um projeto realista, pois saber que um produto tem valor no mundo real provavelmente aumentará o envolvimento e a motivação dos alunos para participar.



Figura 7 – Apresentação da Situação da SD

Neste caso, a âncora para esta SD é um sistema fictício a ser desenvolvido pelos estudantes na forma de miniprojetos: a turma é dividida em equipes e cada equipe fica responsável por construir uma parte de um sistema (miniprojeto) ao longo de período letivo. O termo miniprojeto é utilizado devido à impossibilidade de se desenvolver um sistema completo em 20 semanas, motivo pelo qual os estudantes escolherão apenas algumas partes do sistema para implementar. A escolha da âncora é feita de uma forma tal que cada equipe deve ficar com um sistema diferente para desenvolver durante o semestre.

Por outro lado, para a preparação dos artefatos são utilizadas questões motrizes. Segundo Bender (2015), em conjunto com a âncora, uma questão motriz deve tanto despertar a atenção dos alunos quanto focar seus esforços em identificar as informações específicas que eles necessitam e em qual profundidade, com a intenção de abordar o problema. Por fim, a própria questão motriz deve estimular outras questões mais específicas. O Quadro 4 ilustra algumas âncoras desta SD e suas questões motrizes associadas.

Quadro 4: Âncoras e questões motrizes iniciais

| Âncora (tema dos miniprojetos) | Questões Motrizes |
|---|--|
| Sistema de marcação de assentos em um cinema | <ul style="list-style-type: none"> Quais são os sistemas similares que já existem e estão em operação? Quais serão os usuários que irão interagir com o sistema? Quais funcionalidades são mais importantes para este sistema? Quais artefatos devem ser produzidos? |
| Sistema de marcação de consultas médicas | |
| Sistema de reserva de passagem aérea | |
| Sistema de leilões online | |
| Sistema de compra em um supermercado | |
| Sistema de atendimentos em um salão de beleza | |
| Sistema de vendas em um pet shop | |
| Sistema de gerenciamento de rotatividade em um estacionamento | |
| Sistema de agenda escolar estudantil | |
| Sistema de editoração de notícias | |

Uma vez escolhida a âncora para cada equipe e tendo sido estabelecidas as questões motrizes genéricas a serem respondidas, o próximo passo é a execução (ou desenvolvimento) do projeto propriamente dito.

5.2 Desenvolvimento do Projeto (módulos)

Nesta etapa, os estudantes desenvolverão um miniprojeto de *software* que corresponde à entrega de um subconjunto de funcionalidades de um sistema real. O desenvolvimento desta etapa segue a metodologia descrita na Figura 8.

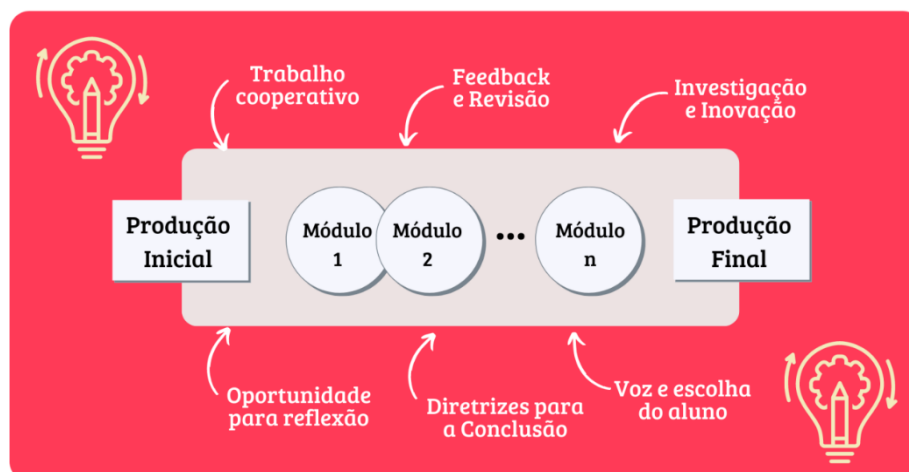


Figura 8 – Desenvolvimento do Projeto

Para nortear as atividades a serem desenvolvidas ao longo da execução do projeto é entregue um enunciado que deve contemplar todas as atividades do SWEBOK (Bourque & Fairley, 2004) trabalhadas na disciplina. O enunciado do projeto estabelece o contexto e uma série de metas gerais que o sistema a ser modelado deve atender. Estas metas correspondem também aos objetivos educacionais de aprendizagem, segundo a taxonomia revisada de Bloom (Krathwohl, 2002). O enunciado é fornecido aos alunos no início do curso, uma vez que tenham adquirido uma série de conceitos básicos sobre o assunto. No entanto, são os alunos que devem completar a especificação de requisitos, desenvolver o protótipo de interface do sistema com seus próprios critérios e codificar a solução proposta, sendo avaliadas a originalidade das soluções e a correta execução dos processos de engenharia.

O enunciado do projeto serve como insumo para a Produção Inicial do Projeto. Segundo Dolz et al (2004), a produção inicial “permite ao professor avaliar as capacidades já adquiridas e ajustar as atividades e os exercícios na sequência didática às possibilidades e dificuldades reais da turma”. Ao se apresentar o enunciado do projeto aos estudantes, através de discussões em conjunto é feito um mapa conceitual de forma colaborativa. Um mapa conceitual é um instrumental importante para organizar e representar o conhecimento, sendo utilizado para (i) identificar os conhecimentos que são necessários adquirir e (ii) aferir o que os alunos já conhecem a respeito do assunto, com o objetivo de tornar a aprendizagem mais significativa. O resultado desta etapa - ou seja, a Produção Inicial do Projeto - é um mapa conceitual feito colaborativamente por todos os alunos da turma.

O restante do projeto a ser desenvolvido segue uma estrutura incremental baseada em entregas, de forma que as entregas são realizadas ao longo da disciplina como se fosse um projeto real. No contexto desta SD, cada atividade do SWEBOK corresponde a um módulo e cada entrega marca o fim de um módulo. A definição de cada módulo, juntamente com seus objetivos educacionais (ou metas a serem cumpridas) e as ferramentas/modelos/tecnologias a serem adotadas são descritos no Quadro 5.

Quadro 5: Atividades a serem seguidas na SD

| Módulos | Metas a Cumprir / Objetivos Educacionais | Categorias da Taxonomia Revisada de Bloom | Ferramentas, Modelos e/ou Tecnologias |
|------------------------|--|---|--|
| Requisitos de Software | <ul style="list-style-type: none"> Identificar casos de uso para o sistema escolhido Criar o Diagrama de Casos de Uso para as funcionalidades do sistema. Criar Especificação de Casos de Uso, Especificação de Regra de Negócios, Glossário. | <ul style="list-style-type: none"> Entender Aplicar Analisar Avaliar Criar | <ul style="list-style-type: none"> Lucidchart¹ Modelos de requisitos em Word, fornecidos como <i>template</i> |
| Métodos Ágeis | <ul style="list-style-type: none"> Transformar os casos de uso em histórias de usuário Criar um cartão para cada história de usuário no quadro Kanban Gerenciar as atividades do projeto através dos cartões. | <ul style="list-style-type: none"> Entender Aplicar Analisar Avaliar Criar | <ul style="list-style-type: none"> Trello² |

Continua na próxima página.

¹ <https://www.lucidchart.com/pages/pt>

² <https://trello.com/pt-BR>

Continuação da página anterior.

| | | | |
|-------------------------|--|---|--|
| Interface com o Usuário | <ul style="list-style-type: none"> Desenhar um protótipo navegável com as telas do sistema, inferidas a partir dos casos de uso | <ul style="list-style-type: none"> Aplicar Analisar Avaliar Criar | <ul style="list-style-type: none"> Figma³ |
| Gestão de Configuração | <ul style="list-style-type: none"> Estabelecer um repositório no GitHub para versionar os artefatos do projeto | <ul style="list-style-type: none"> Aplicar Analisar Avaliar | <ul style="list-style-type: none"> Git/GitHub⁴ |
| Implement. de Software | <ul style="list-style-type: none"> Implementar os casos de uso identificados; Versionar os artefatos produzidos; Acompanhar a situação de cada história de usuário no Trello. | <ul style="list-style-type: none"> Entender Aplicar Analisar Avaliar Criar | <ul style="list-style-type: none"> Python⁵ Java⁶ Git/GitHub Trello |
| Testes de Software | <ul style="list-style-type: none"> Elaborar testes unitários para as unidades de implementação desenvolvidas. | <ul style="list-style-type: none"> Entender Aplicar Analisar Avaliar Criar | <ul style="list-style-type: none"> PyTest⁷, se Python JUnit⁸, se Java |

No primeiro módulo, os alunos devem realizar a especificação de requisitos do sistema, com a elaboração de Diagrama de Casos de Uso, Especificações de Casos de Uso e de Regras de Negócio e a definição do Glossário de Termos. No segundo, os requisitos devem ser reorganizados na forma de histórias de usuário e devem ser gerenciados segundo a metodologia ágil de desenvolvimento de *software*. No terceiro módulo, a meta é desenhar um protótipo de interface que represente os casos de uso previamente identificados. No quarto módulo, o objetivo é versionar todos os artefatos já gerados e os que ainda serão produzidos em um repositório remoto central, como forma de facilitar a gestão de conflitos e de mudanças nestes artefatos. No quinto módulo, inicia-se a etapa de codificação das unidades de implementação, em uma linguagem de programação a ser escolhida em conjunto pelos integrantes da equipe. Finalmente, no sexto módulo são realizados os testes unitários nas unidades de implementação.

Bender (2015) afirma que a possibilidade de escolha por parte do aluno pode ser o componente mais importante de um projeto, sendo crucial para se obter a participação ativa e a apropriação do projeto por parte destes. No decorrer do projeto definido nesta SD, os estudantes podem escolher a âncora que utilizarão, podem identificar novas questões motrizes que sejam relevantes para seus projetos (além das sugeridas), definem as funcionalidades que serão contempladas na sua solução, os padrões de interface para a interação com o usuário, a linguagem de programação que utilizarão para materializar o *software*, além de ter autonomia para várias decisões de projeto, de forma similar ao que acontece em projetos reais de *software*.

Outra premissa importante é que todo o trabalho no projeto deve ser feito de forma colaborativa entre os integrantes da equipe, sendo desencorajados os trabalhos individuais.

³ <https://www.figma.com/>

⁴ <https://github.com/>

⁵ <https://www.python.org/>

⁶ <https://www.java.com/pt-BR/>

⁷ <https://docs.pytest.org/>

⁸ <https://junit.org/junit5/>

Segundo Bender (2015), conforme os alunos ganham experiência em ensino na ABPj, eles também se tornam mais experientes no trabalho em grupo, pois se acostumam a planejar atividades em conjunto, a especificar papéis para os membros do grupo, a trabalhar em conjunto para resolver problemas, a apoiar as ideias uns dos outros e a oferecer, mutuamente, avaliações apropriadas e úteis.

Durante todo o processo, o professor desempenha um papel de facilitador ou tutor, fornecendo *feedback* aos grupos. Segundo Bender (2015), o professor deve usar de todos os meios disponíveis para estimular a investigação e recompensar o pensamento inovador à medida que os alunos avançam em seu projeto. Além disso, a assistência estruturada deve ser rotineiramente fornecida.

Nesse contexto, incentivar a reflexão dos alunos dentro da experiência de ABPj é algo enfatizado por praticamente todos os proponentes deste método (Bender, 2015). Nesta SD, durante os *feedbacks* do professor (principalmente ao final dos módulos) as seguintes questões são colocadas para reflexão de cada grupo: “Como esta atividade pode ser mais bem realizada?”, “O que nós aprendemos desta experiência?”, “Se nós fôssemos realizar novamente a atividade, o que não fizemos, mas faríamos desta vez?”, “Se nós fôssemos realizar novamente a atividade, o que nós fizemos e não faríamos novamente?”, entre outros questionamentos.

As diretrizes para conclusão das atividades em cada módulo são dadas na forma de rubricas de avaliação para cada um dos módulos. Uma rubrica é um sistema de classificação pelo qual o professor determina a que nível de proficiência um aluno é capaz de desempenhar uma tarefa ou evidenciar conhecimento de um conteúdo (Lobato et al, 2007). A rubrica, como ferramenta e metodologia de avaliação formativa baseada em competências, auxilia na comunicação avaliativa entre professores e alunos, ao mesmo tempo em que facilita a gestão da auto avaliação da aprendizagem por estes (Martínez & Raposo, 2014).

Por último, ao concluir todos os módulos, obtêm-se a produção final que é o último estágio da etapa do Desenvolvimento do Projeto. Segundo Dolz et al (2004), a produção final dá ao aluno a possibilidade de pôr em prática as noções e os instrumentos elaborados separadamente nos módulos. Neste caso, a produção final seria a implantação (ou *deploy*) do projeto, ou seja, implantar tudo o que foi desenvolvido previamente nos módulos em um ambiente de produção. Nesta etapa, supostamente, o sistema já estaria disponível para utilização pelos usuários finais. O último passo da SD é a apresentação da produção final.

5.3 Apresentação de Resultados e Avaliações

A última etapa da SD apresentada neste trabalho é a apresentação e publicação dos resultados obtidos durante todo o semestre letivo através das diferentes entregas.

Segundo Bender (2015), como os projetos pretendem ser exemplos autênticos dos tipos de problemas que os alunos enfrentariam no mundo real, é essencial algum tipo de publicação ou apresentação pública dos resultados. Neste caso, a produção final é apresentada através de um vídeo explicativo produzido por toda a equipe e publicado na *Web*, evidenciando as etapas que foram seguidas através dos módulos, bem como os artefatos gerados em cada módulo e sua importância, finalizando com a demonstração do *software* gerado e suas funcionalidades.

Em relações à avaliação, durante os *feedbacks* dados pelo professor são realizadas avaliações formativas, através de diálogos, comentários e reflexões propostas aos estudantes. Por outro lado, as avaliações somativas são realizadas ao final de cada módulo, de acordo com as entregas que são feitas pelo grupo e em concordância com o que foi definido previamente nas rubricas de avaliação. Neste modelo, as entregas são avaliadas considerando o grupo. Assim, a nota do grupo se aplica a cada um dos membros.

6 Análise e Discussão dos Resultados

6.1 Análise

Nesta seção, analisamos os resultados do questionário aplicado à turma da disciplina de ES na qual foi aplicada a metodologia ABPj através da SD definida na Seção anterior. Da turma inicial, de 23 estudantes matriculados, 3 estudantes trancaram o curso e 3 desistiram da disciplina, restando 17 estudantes concludentes. Os alunos foram convidados a responder o questionário da Subseção 4.4 no último dia de aula, a título de avaliação da disciplina. Estes foram informados de que a participação não era obrigatória, que suas respostas não impactariam nas notas e que seu anonimato seria preservado. Dos alunos concludentes, 13 participaram da pesquisa.

Em relação aos períodos letivos em que os alunos estavam formalmente matriculados no momento de sua participação no estudo (Q1), apenas dois alunos não estavam no quinto semestre (um estava no sexto e outro no sétimo semestre). Todos os alunos trabalharam em equipes de até três estudantes (Q2), não havendo nenhum caso de projeto executado de forma individual. Para a totalidade dos respondentes, a disciplina representava o primeiro contato com a ES, tanto a nível acadêmico quanto profissional (Q3 e Q4).

Quanto à seção “Método de Aprendizagem”, o objetivo da questão Q5 foi entender a percepção dos participantes sobre a relevância de um trabalho prático, focado no desenvolvimento de um projeto de *software*, no contexto de desenvolvimento de competências ou aprendizagem de ES. A Figura 9 mostra que a maioria dos participantes indicou que é muito importante. A maioria das respostas (92,3%) foi “Concordo totalmente” com a afirmação e apenas um aluno respondeu “Discordo totalmente”. O objetivo da questão Q6 foi avaliar se os participantes concordam que o uso de aulas expositivas tradicionais, com métodos pontuais de avaliação (exames e trabalhos específicos), é suficiente para o aprendizado da ES. A Figura 9 mostra que as opiniões foram divididas. Três participantes (23%) responderam “Discordo totalmente” com a afirmação, 2 participantes (15,4%) responderam “Discordo parcialmente”, 1 participante (7,7%) respondeu que é “Indiferente” em relação à afirmação, 6 participantes (46,2%) responderam que “Concordo parcialmente” com a afirmação e 1 participante (7,7%) respondeu “Concordo totalmente” com a afirmação. Portanto, a maioria dos estudantes concorda que é importante introduzir projetos de desenvolvimento no contexto do ensino de ES e também acreditam, até certo ponto, que as aulas e exames tradicionais podem ser suficientes para aprender ES.

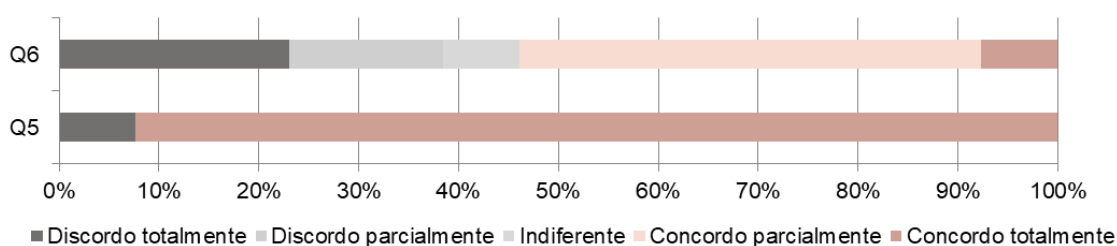


Figura 9: Métodos de Aprendizagem (Q5 e Q6)

Na questão Q7 (Figura 10), perguntou-se aos participantes em que medida o projeto de desenvolvimento de *software* contribuiu para aprender ou desenvolver habilidades em seis tópicos abordados no curso de ES. Os tópicos avaliados foram: “Requisitos de *Software*”,

“Métodos Ágeis”, “Interface com o Usuário”, “Gestão de Configuração”, “Implementação de *Software*” e “Testes de *Software*”.

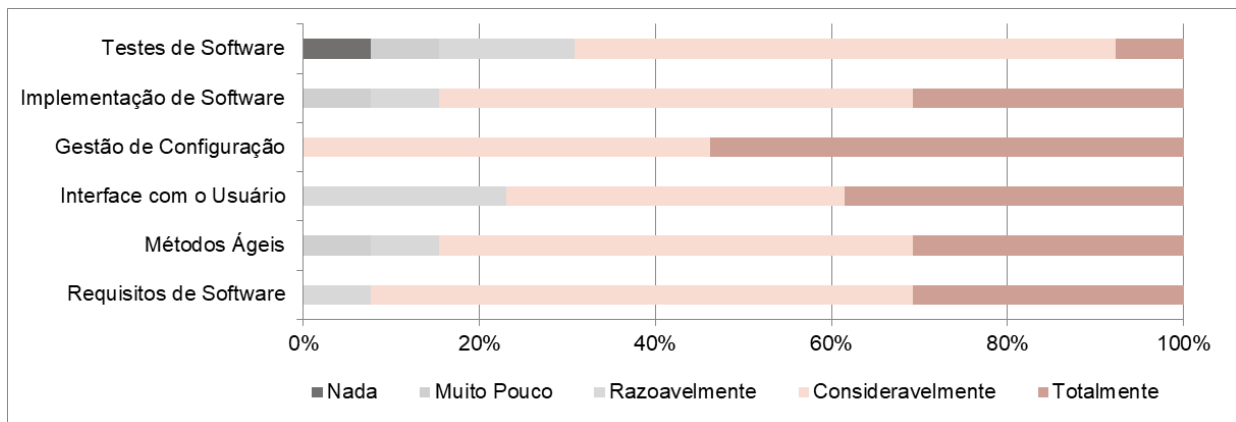


Figura 10: Contribuição do projeto no aprendizado de tópicos específicos de ES (Q7)

A Figura 10 mostra a distribuição das respostas dos estudantes. Os resultados mostram que a atribuição de um projeto de *software* contribuiu para o desenvolvimento de habilidades em todos os tópicos. As respostas “Consideravelmente” e “Totalmente” foram dadas por 92,3% dos estudantes em “Requisitos de *Software*”, 84,6% em “Métodos Ágeis”, 76,9% em “Interface com o Usuário”, 100% em “Gestão de Configuração”, 84,6% em “Implementação de *Software*” e 69,2% em “Testes de *Software*”. Acreditamos que o resultado em Q7 foi obtido devido ao esforço em incluir ferramentas e práticas comumente utilizadas na indústria de *software*, o que pode ter dado uma experiência mais realista aos estudantes e ter contribuído para a motivação em completar o projeto.

Na questão Q8, perguntou-se aos estudantes em que medida o projeto de desenvolvimento de *software* contribuiu para aquisição de experiências nos aspectos essenciais de projetos que seguem a ABPj. Os aspectos avaliados foram: “Trabalho Cooperativo”, “*Feedback* e Revisão”, “Investigação e Inovação”, “Oportunidade para Reflexão”, “Diretrizes para a Conclusão” e “Voz e Escolha do Aluno”. A Figura 11 mostra que o projeto de *software* contribuiu para a aquisição de competências em todos os tópicos, pois a maioria das respostas foi “Consideravelmente” e “Totalmente”. No entanto, alguns estudantes tiveram problemas com suas equipes e isto pode ter contribuído para as avaliações negativas apresentadas.

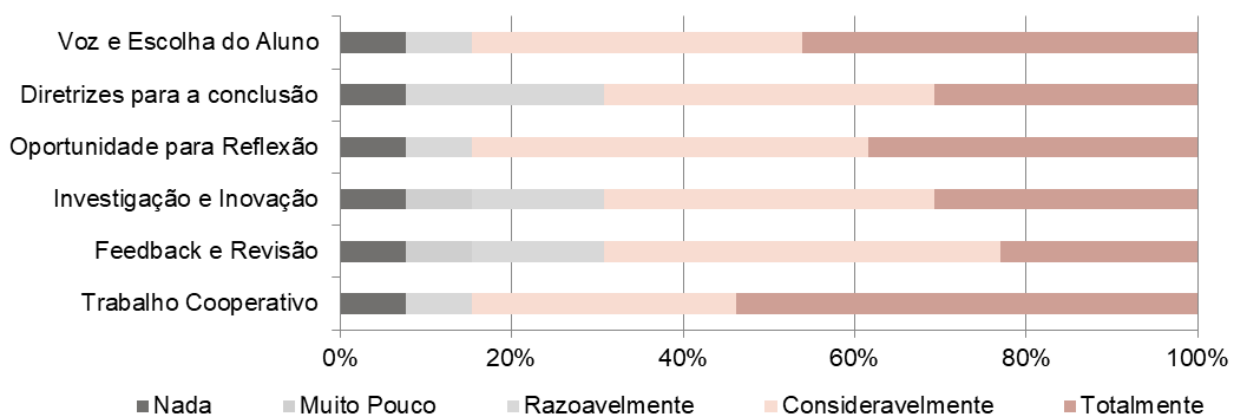


Figura 11: Aquisição de experiências nos aspectos essenciais em projetos que seguem a ABPj (Q8)

Nas questões Q9 e Q10, os participantes foram solicitados a descrever os aspectos positivos e negativos da atribuição do projeto de *software*. Para analisar as respostas, foi utilizada uma abordagem baseada na fase de codificação da *Ground Theory* (Stol, Ralph, & Fitzgerald, 2016). Para tanto, as respostas foram individualmente analisadas e marcaram-se os segmentos relevantes com “códigos” (palavras-chave). Dessa forma, foi possível contar o número de ocorrências de códigos e o número de itens em cada categoria para entender quais aspectos positivos e negativos são apontados de forma recorrente pelos participantes e, assim, discutir possíveis aprendizados.

As categorias de códigos utilizadas nesta análise, tanto para os aspectos positivos quanto para os negativos, foram baseadas nas categorias identificadas no trabalho de Souza et al (2019). Para os aspectos positivos da questão Q9, as categorias são as seguintes: (i) “Processo de Aprendizagem”: agrupa códigos que se relacionam às falas dos participantes sobre como o projeto os auxiliou na aquisição, retenção e aprofundamento do conhecimento, e como facilitou a compreensão dos tópicos de ES; (ii) “Profissionalismo”: agrupa códigos relacionados à percepção dos participantes sobre como o projeto simula um ambiente de trabalho semelhante ao contexto profissional de ES e (iii) “Prática”: agrupa códigos relacionados aos aspectos positivos da natureza prática do projeto. Já para os aspectos negativos da questão Q10, as categorias de códigos são: (i) Falta de Tempo; (ii) Implementação; (iii) Organização da Disciplina e (iv) Uso do Processo de *Software*.

O Quadro 6 lista todos os aspectos positivos e negativos captados respectivamente dos resultados de Q9 e Q10, suas categorias e o número de ocorrências. Quanto aos aspectos positivos, identificamos 13 códigos únicos na amostra. Os dados mostram que os aspectos positivos estão distribuídos uniformemente em todas as categorias, indicando que ao incluir a atividade de desenvolvimento de um projeto de *software* houve benefícios igualmente percebidos quanto à prática, processo de aprendizagem e profissionalismo.

Quadro 6: Aspectos positivos e negativos e respectivas categorias

| Aspectos Positivos | Categoria | Total | Total por Categorias |
|---------------------------------------|--------------------------|-------|----------------------|
| Aplicar conhecimentos | Prática | 1 | 5 |
| Praticar conhecimentos | | 2 | |
| Motivação para desenvolver | | 1 | |
| Treinar conhecimentos | | 1 | |
| Importância de cada área de ES | Processo de Aprendizagem | 1 | 5 |
| Melhor compreensão dos tópicos | | 1 | |
| Pesquisar conteúdos | | 1 | |
| Proporciona aprendizado | | 1 | |
| Ruptura com práticas anteriores | | 1 | |
| Contato com ambiente real de trabalho | Profissionalismo | 1 | 6 |
| Experiência realista de mercado | | 3 | |
| Modelo próximo da realidade | | 1 | |
| Oportunidade de mercado | | 1 | |

Continuação na próxima página.

Continuação da página anterior.

| Aspectos Negativos | Categoria | Total | Total por Categorias |
|---|------------------------------------|-------|----------------------|
| Extenso conteúdo | Falta de Tempo | 3 | 7 |
| Dificuldade de conciliar com outras disciplinas | | 2 | |
| Pouco tempo disponível | | 2 | |
| Dificuldade para solucionar o problema | Implementação | 1 | 3 |
| Falta de conhecimentos prévios | | 2 | |
| Formação tardia de grupos | Organização da Disciplina | 1 | 5 |
| Definição tardia do projeto | | 1 | |
| Falta de feedback | | 1 | |
| Poucos exemplos práticos | | 2 | |
| Dificuldade em fazer algo novo | Uso do Processo de <i>Software</i> | 2 | 4 |
| Complexidade alta | | 1 | |
| Levantamento de Requisitos | | 1 | |

Em relação aos aspectos negativos, identificamos 12 códigos únicos. Os dados mostram que os aspectos negativos estão mais concentrados na categoria “Falta de Tempo” e “Organização da Disciplina”, com os alunos relatando dificuldades principalmente em relação ao cumprimento do conteúdo programático e a conciliar as atividades da disciplina de ES com outras disciplinas. Como pontos a melhorar na SD, segundo os alunos, estão a definição tempestiva de projetos e equipes, assim como a inclusão de mais exemplos práticos e *feedbacks* mais frequentes.

6.2 Discussão

Nesta seção, revisitamos as questões de pesquisa (QP) definidas na Seção 4.1 e discutimos nossas descobertas em relação aos resultados obtidos. Em relação à utilização de um projeto prático de desenvolvimento de *software* para o aprendizado de ES, os estudantes são fortemente favoráveis à sua relevância.

Para a QP1 (“Qual é a percepção dos alunos sobre o uso de palestras tradicionais e exames no ensino de ES?”), aproximadamente metade dos alunos concordam que as aulas expositivas tradicionais e os métodos de avaliação podem ser suficientes para o aprendizado da ES. Isso pode ser explicado em parte porque a SD com o projeto prático ocorreu em paralelo às aulas tradicionais da disciplina que, por sua vez, também davam subsídios ao desenvolvimento do projeto.

Para a QP2 (“Como os alunos percebem a contribuição de um trabalho na forma de um projeto para o aprendizado de tópicos específicos de ES?”), os alunos em sua maioria perceberam uma contribuição positiva do projeto de *software* em seu processo de aprendizagem de ES. Os resultados mostram que a atribuição de um projeto de *software* contribuiu para o desenvolvimento de habilidades em todos os tópicos, mas principalmente em “Gestão de Configuração” e “Requisitos de *Software*” que são áreas essencialmente práticas.

Para o QP3 (“Quais são os aspectos positivos observados pelos alunos em um curso com ABPj?”), os aspectos positivos observados pelos alunos estavam relacionados à oportunidade de

aquisição, retenção e aprofundamento do conhecimento, à simulação de ambientes de trabalho semelhantes ao contexto profissional e à possibilidade de aplicar os conhecimentos adquiridos na disciplina. Portanto, acreditamos que a metodologia ABPj foi recebida positivamente pelos alunos.

Por último, para o QP4 (“Quais são os aspectos negativos que os alunos observaram em um curso com ABPj?”), os aspectos negativos mais predominantes observados nas respostas dos alunos disseram respeito, nesta ordem, à falta de tempo devido principalmente ao conteúdo considerado extenso, à forma de organização de algumas atividades na disciplina, à dificuldade na adequação ao processo de *software* e em tópicos relativos à atividade de implementação. Portanto, houve uma aceitação geral positiva da metodologia ABPj, através da SD proposta, por parte dos alunos. Um projeto de desenvolvimento de *software* não só ajudou a equilibrar teoria e prática, mas também proporcionou aos alunos a oportunidade de compreender alguns aspectos que só a teoria não abordaria.

Como lições aprendidas, destacamos que: (i) o uso de uma SD com ABPj em paralelo às aulas teóricas pode sobrecarregar os alunos, sendo necessário rever a distribuição da carga horária da disciplina de forma a acomodar melhor o conteúdo teórico e o conteúdo prático; (ii) é preciso uma definição clara do projeto e das atividades o mais cedo possível (preferencialmente nas primeiras aulas da disciplina), bem como oferecer *feedback* das entregas com mais regularidade e trazer mais exemplos práticos em sala de aula; (iii) a noção de realismo, em projetos de ABPj, não está relacionada necessariamente ao uso de projetos reais, mas também à criação de um contexto que simule ambientes profissionais.

6.3 Limitações do Trabalho

O estudo realizado apresentou limitações importantes principalmente em relação à amostra, uma vez que esta tinha um tamanho reduzido e foi obtida por conveniência, o que nos restringe quanto à generalização dos resultados obtidos. No entanto, embora seja assim, acreditamos que este estudo ainda é significativo para entender mais sobre a percepção dos estudantes a respeito do uso da metodologia ABPj em disciplinas de ES.

Um ponto não abordado neste estudo foi a avaliação de satisfação e aceitação dos professores em relação à SD proposta, o que poderia ter explicado melhor os aspectos positivos e negativos identificados na codificação, segundo o ponto de vista dos estudantes.

7 Considerações Finais e Trabalhos Futuros

Este artigo descreveu a aplicação da metodologia ABPj, através de uma SD, para a disciplina de ES, que utiliza os objetivos educacionais da taxonomia revisada de Bloom. Além disso, como forma de validar a proposta através de uma prova de conceito, avaliou-se a aplicação desta metodologia em uma turma regular de ES, utilizando como instrumento de coleta um *survey*. Com os resultados obtidos, foi possível analisar a percepção dos alunos sobre o uso de projetos de *software* para o aprendizado de tópicos específicos de ES.

Nossos resultados mostraram que: (i) os alunos concordam que é importante usar projetos práticos de desenvolvimento de *software* no contexto do ensino de ES, ao invés de depender apenas de aulas e provas tradicionais; (ii) há uma percepção positiva da contribuição do trabalho de projeto na aprendizagem em todos os tópicos específicos de ES; e (iii) os alunos identificaram aspectos positivos relacionados ao projeto em relação ao processo de aprendizagem e profissionalismo.

Para trabalhos futuros, pretende-se executar novos ciclos de design da SD proposta com a opinião de um número maior de especialistas como docentes da área de ES, profissionais da indústria de *software* e pedagogos, a fim de obter direcionamentos em relação à aplicabilidade desta SD. Destaca-se também a possibilidade de se realizar um estudo mais longo, com a aplicação desta SD em mais turmas e com mais alunos participantes, com a intenção de avaliar também o impacto na aprendizagem obtido em cada aplicação, bem como realizar correlações entre as respostas dos questionários com o desempenho obtido pelos alunos, obtido através das notas. Além de validar a SD com alunos de outras instituições, como forma de prover a validade externa da proposta. Do ponto de vista dos professores, pretendemos avaliar futuramente o grau de aceitação e satisfação destes com a SD proposta.

Referências

- ACM/IEEE (2014). *Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering*. [[GS Search](#)]
- Alves, C. F., & de Matos, M. E. (2017). Sequência didática para conteúdo de engenharia de software. *Revista Brasileira de Ensino de Ciência e Tecnologia*, 10(3). DOI: [10.3895/rbect.v10n3.4645](#). [[GS Search](#)]
- Anicic, K. P., & Stapic, Z. (2022). Teaching Methods in Software Engineering: Systematic Review. *IEEE Software*. DOI: [10.1109/MS.2022.3152629](#). [[GS Search](#)]
- Araújo, D. L. de (2013). O que é (e como faz) sequência didática?. *Entrepalavras*, 3(1), 322-334. DOI: [10.22168/2237-6321.3.3.1.322-334](#). [[GS Search](#)]
- Barbosa, E. F., & de Moura, D. G. (2013). Metodologias ativas de aprendizagem na educação profissional e tecnológica. *Boletim Técnico do SENAC*, 39(2), 48-67. DOI: [10.26849/bts.v39i2.349](#). [[GS Search](#)]
- Bayazit, N. (2004). Investigating design: A review of forty years of design research. *Design issues*, 20(1), 16-29. [[GS Search](#)]
- Bender, W. N. (2015). *Aprendizagem baseada em projetos: educação diferenciada para o século XXI*. Penso Editora. [[GS Search](#)]
- Bloom, B. S. (1956). Taxonomy of educational objectives: The classification of educational goals. *Cognitive domain*. [[GS Search](#)]
- Bloom, B. S., Madaus, G. F., & Hastings, J. T. (1971). *Handbook on Formative and Summative Evaluation of Student Learning*. New York: McGraw-Hill. [[GS Search](#)]
- Bourque, P., & Fairley, R.E (2004). *SWEBOK : Guide to the Software Engineering Body of Knowledge*. Los Alamitos, CA: IEEE Computer Society. [[GS Search](#)]
- Britto, R., & Usman, M. (2015). Bloom's taxonomy in software engineering education: A systematic mapping study. In *2015 IEEE Frontiers in Education Conference (FIE)* (pp. 1-8). IEEE. DOI: [10.1109/FIE.2015.7344084](#). [[GS Search](#)]
- Carvalho, A. F. de, Junqueira, J. R. A., & Furlanetto, P. G. (2017). Estratégias Ativas de Aprendizagem e o Desenvolvimento de Competências Técnicas e Atitudinais. *Revista Ensaios Pioneiros*, 1(1), 139-152. DOI: [10.24933/rep.v1i1.23](#). [[GS Search](#)]
- Cunha, J. A. O., Marques, G. A., Lemos, W. L., Câmara, U. D., Jr., & Vasconcellos, F. J. (2018). Software engineering education in Brazil: a mapping study. In *Proceedings of the*

- XXXII Brazilian Symposium on Software Engineering (pp. 348-356). DOI: [10.1145/3266237.3266259](https://doi.org/10.1145/3266237.3266259) [GS Search]
- Devadiga, N. M. (2017). Software engineering education: Converging with the startup industry. In *2017 IEEE 30th Conference on Software Engineering Education and Training (CSEE&T)* (pp. 192-196). IEEE. DOI: [10.1109/cseet.2017.38](https://doi.org/10.1109/cseet.2017.38). [GS Search]
- Dolz, J., Noverraz, M., & Schneuwly, B. (2004). Sequências didáticas para o oral e a escrita: apresentação de um procedimento. *Gêneros orais e escritos na escola*. Campinas: Mercado de Letras, 95-128. [GS Search]
- Dresch, A., Lacerda, D. P., & Miguel, P. A. C. (2015). Uma análise distintiva entre o estudo de caso, a pesquisa-ação e a design science research. *Revista Brasileira de Gestão de Negócios*, 17, 1116-1133. DOI: [10.7819/rbgn.v17i56.2069](https://doi.org/10.7819/rbgn.v17i56.2069). [GS Search]
- Easterbrook, S., Singer, J., Storey, M. A., & Damian, D. (2008). Selecting empirical methods for software engineering research. In *Guide to advanced empirical software engineering* (pp. 285-311). Springer, London. DOI: [10.1007/978-1-84800-044-5_11](https://doi.org/10.1007/978-1-84800-044-5_11). [GS Search]
- Ferraz, A. P. D. C. M., & Belhot, R. V. (2010). Taxonomia de Bloom: revisão teórica e apresentação das adequações do instrumento para definição de objetivos instrucionais. *Gestão, & produção*, 17, 421-431. DOI: [10.1590/S0104-530X2010000200015](https://doi.org/10.1590/S0104-530X2010000200015). [GS Search]
- Fu, F. L., Wu, Y. L., & Ho, H. C. (2009). An investigation of cooperative pedagogic design for knowledge creation in web-based learning. *Computers, & Education*, 53(3), 550-562. DOI: [10.1016/j.compedu.2009.01.004](https://doi.org/10.1016/j.compedu.2009.01.004). [GS Search]
- Galhardi, A. C., & Azevedo, M. M. D. (2013). Avaliações de aprendizagem: o uso da taxonomia de Bloom. In *Anais do VII Workshop Pós-graduação e Pesquisa do Centro Paula Souza, São Paulo* (Vol. 1, No. 1, pp. 237-247). [GS Search]
- Gil, A. C. (2002). *Como elaborar projetos de pesquisa*. São Paulo: Atlas. [GS Search]
- Gupta, V., & Nguyen-Duc, A. (2021). *Real-world Software Projects for Computer Science and Engineering Students*. CRC Press. [GS Search]
- Henrique, M. S., Farias, A. B., Cunha, F. O. M., & Scaico, P. D. (2013). Proposta para Construção de Sequências Didáticas para aulas de Matemática com uma Atividade de Computação Desplugada. *Nuevas Ideas en Informática Educativa*. [GS Search]
- Hoed, R. M. (2016). Análise da evasão em cursos superiores: o caso da evasão em cursos superiores da área de Computação. Dissertação de Mestrado, Universidade de Brasília, Brasília, DF, Brasil. [GS Search]
- Horita, F. E. A., Graciano, V. V., Neto, & Santos, R. P. (2018). Design Science Research em Sistemas de Informação e Engenharia de Software: Conceitos, Aplicações e Trabalhos Futuros. *I Jornada Latino-Americana de Atualização em Informática*, 191-210. [GS Search]
- Krathwohl, D. R. (2002). A revision of Bloom's taxonomy: An overview. *Theory into practice*, 41(4), 212-218. DOI: [10.1207/s15430421tip4104_2](https://doi.org/10.1207/s15430421tip4104_2). [GS Search]
- Lima, J., Alencar, F., & Santos, W. (2021). A Preliminary Guide for Assertive Selection of Active Methodologies in Software Engineering Education. In *Brazilian Symposium on Software Engineering* (pp. 170-179). DOI: [10.1145/3474624.3476976](https://doi.org/10.1145/3474624.3476976). [GS Search]

- Lima, J. V. V., Silva, C. A. D., de Alencar, F. M. R., & Santos, W. B. (2020). Metodologias Ativas como forma de reduzir os desafios do ensino em Engenharia de Software: diagnóstico de um survey. In *Anais do XXXI Simpósio Brasileiro de Informática na Educação* (pp. 172-181). SBC. DOI: [10.5753/cbie.sbie.2020.172](https://doi.org/10.5753/cbie.sbie.2020.172). [GS Search]
- Lobato, A. S., Silva, A. S., Lino, A. D. P., Favero, E. L., Harb, M. P. A. A., Brito, S. R. de, & Gonçalves, L. P. (2007). Uma rubrica para avaliação de cursos de programação centrada em avaliação automática. In *SBIE-Workshop de Ambientes de apoio à Aprendizagem de Algoritmos e Programação*. [GS Search]
- Masapanta-Carrión, S., & Velázquez-Iturbide, J. Á. (2017). Una revisión sistemática del uso de la taxonomía de Bloom en la enseñanza de la informática. *Atas do XIX Simpósio Internacional de Informática Educativa e VIII Encontro do CIED–III Encontro Internacional*, 294-299. [GS Search]
- Masson, T. J., Miranda, L. F. D., Munhoz Jr, A., & Castanheira, A. M. P. (2012). Metodologia de ensino: aprendizagem baseada em projetos (PBL). In *Anais do XL Congresso Brasileiro de Educação em Engenharia (COBENGE), Belém, PA, Brasil* (p. 13). sn. [GS Search]
- Oguz, D., & Oguz, K. (2019). Perspectives on the gap between the software industry and the software engineering education. *IEEE Access*, 7, 117527-117543. DOI: [10.1109/access.2019.2936660](https://doi.org/10.1109/access.2019.2936660). [GS Search]
- Oliveira, C. J. A. de., & Lavor, O. P. (2022). Sequência didática para o ensino e aprendizagem de juros compostos com o software Geogebra. *Boletim Cearense de Educação e História da Matemática*, 9(25), 96-110. DOI: [10.30938/bocehm.v9i25.7400](https://doi.org/10.30938/bocehm.v9i25.7400). [GS Search]
- Pfleeger, S. L., & Kitchenham, B. A. (2001). Principles of survey research: part 1: turning lemons into lemonade. *ACM SIGSOFT Software Engineering Notes*, 26(6), 16-18. [GS Search]
- Prikladnicki, R., Albuquerque, A. B., von Wangenheim, C. G., & Cabral, R. (2009). Ensino de engenharia de software: desafios, estratégias de ensino e lições aprendidas. *FEES-Fórum de Educação em Engenharia de Software*, 1-8. [GS Search]
- Martínez, E., Tellado, F., & Raposo, M. (2014). Evaluación educativa utilizando rúbrica: un desafío para docentes y estudiantes universitarios. DOI: [10.5294/edu.2014.17.3.6](https://doi.org/10.5294/edu.2014.17.3.6) [GS Search]
- Reeves, T. (2006). Design research from a technology perspective. In *Educational design research* (pp. 64-78). Routledge. [GS Search]
- Sales, A. B. de, Serrano, M., & Serrano, M. (2020). Aprendizagem baseada em projetos na disciplina de interação humano-computador. *Revista Ibérica de Sistemas e Tecnologias de Informação*, (37), 49-64. [GS Search]
- Serrano, L. M., Serrano, M., & Sales, A. B. de (2021). Desenvolvimento de competências profissionais: relato da experiência utilizando aprendizagem baseada em projetos na disciplina de requisitos de software. *Revista de Ensino de Engenharia*, 40. DOI: [10.37702/ree2236-0158.v40p76-81.2021](https://doi.org/10.37702/ree2236-0158.v40p76-81.2021) [GS Search]
- Souza, M., Moreira, R., & Figueiredo, E. (2019). Students perception on the use of project-based learning in software engineering education. In *Proceedings of the XXXIII Brazilian Symposium on Software Engineering* (pp. 537-546). DOI: [10.1145/3350768.3352457](https://doi.org/10.1145/3350768.3352457) [GS Search]

- Stol, K. J., Ralph, P., & Fitzgerald, B. (2016). Grounded theory in software engineering research: a critical review and guidelines. In *Proceedings of the 38th International conference on software engineering* (pp. 120-131). DOI: [10.1145/2884781.2884833](https://doi.org/10.1145/2884781.2884833). [[GS Search](#)]
- Tonhão, S. de F., Medeiros, A. de S. S., & Prates, J. M. (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* (pp. 143-151). SBC. DOI: [10.5753/educomp.2021.14480](https://doi.org/10.5753/educomp.2021.14480) [[GS Search](#)]
- Toyohara, D. Q. K., Sena, G. J. de, Araújo, A. M. de., & Akamatsu, J. I. (2010). Aprendizagem Baseada em Projetos—uma nova Estratégia de Ensino para o Desenvolvimento de Projetos. In *PBL—Congresso Internacional*. [[GS Search](#)]
- Uzun, M. L. C., Pugliesi, J. B., & Roland, C. E. de F. (2018). Aprendizagem baseada em projetos na perspectiva dos alunos. *Revista Profissão Docente*, 18(39), 403-414. DOI: [10.31496/rpd.v18i39.1212](https://doi.org/10.31496/rpd.v18i39.1212). [[GS Search](#)]
- Valente, M. T. (2020). *Engenharia de Software Moderna: Princípios e Práticas para Desenvolvimento de Software com Produtividade*. Disponível em: <https://engsoftmoderna.info/>. Acesso em: 30 junho 2022. [[GS Search](#)]
- Vázquez-Ingelmo, A., García-Holgado, A., García-Peñalvo, F. J., & Rodríguez-Conde, M. J. (2019). Resultados preliminares tras tres años aplicando aprendizaje basado en proyectos en ingeniería del software. DOI: [10.26754/cinaic.2019.0141](https://doi.org/10.26754/cinaic.2019.0141). [[GS Search](#)]
- Zabala, A. (2015). *A prática educativa: como ensinar*. Penso Editora. [[GS Search](#)]