



Uma Abordagem Prática de *Design Thinking* no Processo de Ensino e Aprendizagem na Engenharia de Software

Rodrigo Gusmão de Carvalho Rocha - UFape - rodrigo.rocha@ufape.edu.br

ORCID: 0000-0003-1993-5044

Igor Medeiros Vanderlei - UFape - igor.vanderlei@ufape.edu.br,

ORCID: 0000-0002-1939-6979

Danillo Bion - UFape - danillobion@gmail.com

ORCID: 0000-0003-1698-6276

Ana Beatriz Almeida Vanderlei - UFape - anabiaavanderlei@gmail.com,

ORCID: 0000-0002-7184-605X

Jean Carlos Teixeira de Araújo - UFape - jean.teixeira@ufape.edu.br

ORCID: 0000-0002-1688-4782

Resumo. Com o aumento significativo da indústria de software nas últimas décadas, permanecem em destaque os desafios na produção de soluções computacionais, como a falta de mão de obra qualificada e processos eficientes de desenvolvimento de software. O objetivo deste artigo é apresentar uma abordagem prática que utiliza o conceito de *Design Thinking* como recurso elementar para o processo de ensino e aprendizagem na construção de software, possibilitando uma formação técnica diversificada com interatividade e abordagem com foco no usuário. Esta proposta foi concebida e avaliada em laboratórios de ensino de tecnologia em uma universidade pública. Como resultado, adaptou-se o processo de ensino e aprendizagem para proporcionar uma formação holística aos discentes e um maior alinhamento ao mercado de trabalho.

Palavras-chave: *Ensino de Desenvolvimento de Software, Design Thinking, Processo de Software.*

A Practical Approach to Design Thinking in the Software Engineering Teaching-Learning Process

Abstract. *There is remarkable growth in the software industry in recent decades, the challenges in computational solutions production remain highlighted, such as qualified labor shortage and efficient software development processes. This paper's goal is to present a practical approach that uses Design Thinking concept as an elementary resource to teaching-learning process in software engineering, enabling a diversified technical education with interactivity and a user-focused approach. This proposal was conceived and evaluated in technology teaching laboratories at a public university. As a result, the teaching-learning process was adapted to provide holistic training to students and greater alignment to the software industry.*

Keywords: *Software Development Teaching, Design Thinking, Software Process.*

1. Introdução

Segundo a IDC (2020), nos últimos anos, houve um crescimento significativo no mercado de TI. Mesmo com a pandemia e toda a crise internacional, o mercado expandiu e se mantém em alta, com previsão de aumento em 7% para 2021. Como consequência, o desenvolvimento de sistemas tem recebido aumento expressivo de



demanda e investimentos. Nesse contexto, a produção de soluções computacionais possui conhecidos desafios, como a falta de mão de obra qualificada e um processo de desenvolvimento de software que permita o aprimoramento da garantia de qualidade dos produtos desenvolvidos. Dessa maneira, os cursos superiores da área de tecnologia precisam estar atentos a esses movimentos de mercado, de modo a formar mais profissionais com as habilidades necessárias para o atendimento da demanda em constante mudança.

Neste contexto, durante a formação dos estudantes na graduação, conceitos aprendidos e exercitados em disciplinas do curso ou até em projetos de pesquisa, ensino e extensão, não garantem suficiência para uma boa atuação profissional. Em seu trabalho, Portela et al. (2017) afirmam que os discentes precisam, além do conhecimento teórico, adquirir habilidades técnicas, sendo expostos a problemas reais do ambiente profissional, permitindo maior evolução como profissional da área. Rosa et al. (2021) citam em seu estudo que o ambiente pode proporcionar aos estudantes a experiência prática da aplicação de técnicas, que é muito necessário para formação de um profissional da área. A formação acadêmica do profissional da computação é um desafio, pela permanente atualização e reinvenção, mas também devido às relações de mercado que se complexificam cotidianamente. E neste caso, os laboratórios de tecnologia contribuem diretamente na formação do aluno e também na sua inserção no mercado de trabalho (VANDERLEI et al, 2020).

Dessa forma, este artigo apresenta uma abordagem prática que utiliza o conceito de *Design Thinking* (DT) como ferramenta elementar para o processo de ensino e aprendizagem da Engenharia de Software (ES). Os produtos de software desenvolvidos através de técnicas de DT naturalmente possuem maiores chances de garantir a qualidade de software prevista. Este conceito envolve um processo colaborativo que utiliza a sensibilidade e criatividade para tentar atender às necessidades das pessoas, não necessariamente somente no visual do produto, mas como uma solução de negócio viável (BROWN, 2019).

Este artigo está organizado da seguinte maneira: na Seção 2, os principais conceitos do trabalho são apresentados, como Engenharia de Software, *Design Thinking*, Prototipação e Design de Interface. Em seguida, na Seção 3, os métodos e ambientes de experimentação são retratados. Na Seção 4, a proposta deste trabalho é descrita e detalhada, por fim, na última seção, as considerações finais são destacadas.

2. Engenharia de Software, *Design Thinking* e seus principais conceitos

Sommerville (2011) define software como produtos abstratos e intangíveis, pois esses sistemas podem se tornar complexos rapidamente, difíceis de entender e caros de mudar. O autor ainda afirma que as empresas operam em um ambiente dinâmico e global, no qual precisam responder rapidamente às novas oportunidades de mercado e mudanças econômicas, com a criação e entrega rápida de softwares.

A Engenharia de Software tem o intuito de prover as atividades de desenvolvimento com controle e planejamento. Pressman (2014) define ES como um processo onde existe um conjunto de métodos (práticas), diversos papéis e artefatos e uma série de ferramentas que permitem aos colaboradores criarem softwares de alta qualidade. Neste mesmo contexto, com o objetivo de desenvolver software com qualidade e que proporcionem uma melhor interação entre os envolvidos, o conceito de *Design Thinking* oferece uma metodologia no planejamento e execução de um projeto



de software que busca inovação, colaboração e experimentação, aplicada a projetos reais com a abordagem com foco no usuário (BROWN, 2019).

Marques et al. (2015) também afirmam que projetos de software têm maiores chances de resolver limitações quando o foco da satisfação é nas necessidades dos usuários. Lindberg et al. (2011) fundamentam que integrar o conceito de DT com sucesso no desenvolvimento de aplicações ainda é um grande desafio, sendo necessário mais relatos da aplicabilidade e dos resultados obtidos, tanto no processo de desenvolvimento, quanto na qualidade final das soluções desenvolvidas. Fawcett et al. (2013) cita que o DT aplica diversas técnicas criativas e de inovação para desenvolver produtos modernos.

Uma das vantagens da aplicação de DT é que seus conceitos podem ser aplicados em qualquer cenário e fase (THORING e MÜLLER, 2011). As principais macro etapas, definidas por Vieira et al. (2014) são: Imersão, que busca imaginar e observar os cenários a partir de perspectivas dos usuário para coletar o contexto e limitações do projeto; Ideação, que utiliza os insights obtidos na Imersão para propor soluções que estejam de acordo com as necessidades; E por fim, a prototipação, que pode funcionar em paralelo com essas etapas anteriores, com objetivos de validar e avaliar as ideias sugeridas, da mesma forma que, antecipar algumas limitações que surgem no decorrer da prototipação.

3. Ambiente de Experimentação e Métodos Utilizados

Esta seção apresenta o contexto do ambiente de experimentação desse estudo, no caso, os laboratórios de tecnologia, de uma instituição pública de ensino superior, responsáveis pela execução de projetos de software e um dos protagonistas para esse aperfeiçoamento do processo de ensino e aprendizagem de Engenharia de Software no âmbito do curso de Ciência da Computação, assim como os métodos utilizados e os objetos de estudo.

3.1 Laboratórios de Experimentação

Esta pesquisa utilizou como ambiente de experimentação dois laboratórios de tecnologia, que atuam com desenvolvimento de software, o Laboratório Multidisciplinar de Tecnologias Sociais (LMTS) e o Laboratório de Pesquisa, Inovação e Desenvolvimento BCC Coworking, nos quais foram realizados os testes e avaliação do processo de *design thinking* proposto. Para tal, foram utilizados 5 projetos de software de cada laboratório, que tiveram atividades diretamente ligadas a esta proposta. São projetos que variam de finalidade, tecnologias, arquiteturas e tamanho, porém, dispõem do processo de concepção com foco em *design thinking*.

3.2 Metodologia e Processo de Desenvolvimento de Software Empregados

Seguindo conceitos da Engenharia de Software, os projetos são conduzidos por um conjunto de etapas e métodos bem definidos, como também através do uso de ferramentas e tecnologias consolidadas. Como guia de referência geral foram utilizados o SWEBOK (BOURQUE e FAIRLEY, 2014) e práticas do PMBOK (2017), realizando apenas as devidas adaptações às necessidades do contexto educacional, que envolve diretamente:

- Práticas Ágeis
- Tecnologias de Desenvolvimento: a depender dos projetos

- Gestão de Projetos: Scrum, Trello, reuniões periódicas, definição de responsáveis;
- Ferramentas de comunicação: Slack, Hangout, Calendar, entre outros;
- Papéis envolvidos: Clientes, coordenações, designer, gerente de projetos, desenvolvedores, testadores, usuários;
- Artefatos projetados: Estudo de Viabilidade, Termo de Abertura de Projeto, Plano de Projeto, Formulários para testes de aceitação manual e gerência de mudanças.

3.3 Objetos de Estudo: projetos de software

Nestes laboratórios já foram desenvolvidas várias soluções tecnológicas que estão em uso e outras continuam em desenvolvimento. Neste sentido, para a escolha dos projetos que empregaram atividades desta proposta de processo de *design thinking*, foram utilizados como critérios a quantidade de atividades de *design thinking* e os projetos mais recentes, uma vez que estes participaram de todas ou quase todas as etapas propostas neste processo.

É importante evidenciar que estes objetos de estudo foram concebidos utilizando a abordagem proposta desde o início do projeto. Dessa forma, todas essas etapas e fases propostas foram executadas em todos os projetos, com exceção de algumas particularidades de cada aplicação.

4. Proposta da Abordagem Prática de *Design Thinking* na Engenharia de Software

Esta seção descreve as atividades propostas neste Framework, ilustrando com detalhes todos os procedimentos envolvidos, como por exemplo: atividades simples, artefatos, papéis desempenhados, as técnicas e métodos utilizados. Para cada atividade, os objetos de estudo (que se referem aos projetos desenvolvidos) são evidenciados, revelando aquele procedimento descrito para com o projeto em questão.

A proximidade do usuário final ou do cliente no decorrer dos projetos é indispensável para um melhor atendimento do que foi previsto e também um maior acabamento nos produtos. Para demonstrar melhor, a **Figura 1** apresenta de forma ilustrada cada atividade realizada nesta proposta. Nas subseções seguintes estão descritas as características e finalidades em cada atividade presente neste framework.





Figura 1: Proposta de Processo de *Design Thinking* no desenvolvimento de software

4.1 Apresentação Inicial (*Briefing*)

É um procedimento inicial do processo, com exposição do cliente e/ou seus representantes sobre as necessidades, expectativas, preferências, regras, limitações, entre outros, contando com a participação também da equipe do laboratório.

Esta etapa tem o objetivo captar e extrair o máximo de informações possíveis sobre o projeto e as necessidades do cliente, identificando e analisando as principais prioridades que apoiam futuras decisões. As respostas e expectativas do cliente servem como base para todas as tomadas de decisão do design. Este momento é fundamental e deve abordar tópicos como: objetivo do produto, público-alvo, idade do usuário final, se existe legislação específica, se existem materiais adicionais podem ser compartilhados, prioridades, restrições, prazos, entre outros.

4.2 Análise do projeto

Após a primeira atividade, é necessário que a equipe de desenvolvimento estude o material coletado e faça uma análise da conjuntura. Em seguida, os artefatos produzidos pelos times de desenvolvimento dos laboratórios, no caso, "Histórias de Usuário" e "Termo de Abertura de Projeto" são examinados. A partir desta documentação completa é possível identificar todos os perfis de usuário do projeto, priorizar funcionalidades e entender de maneira geral a proposta da solução. Essa fase é essencial para a projeção e desenvolvimento do layout, conseqüentemente, o sucesso do design do projeto.

4.3 Pesquisa Direcionada

Durante esse ciclo, apoiado pelo entendimento inicial e estudo geral, o designer busca referências para suas futuras projeções do produto. Diversas fontes podem servir como referência sob aquele tema ou necessidade, desde notícias relevantes, dados estatísticos, design, redesign de projetos, *personas*, entre outros. É uma forma do protótipo ser concebido sob um ecossistema de informações referente a um tema.

Há também a realização de pesquisa por *personas*, que é a projeção de um personagem fictício para representar os diferentes grupos de usuário que irão interagir com aquela aplicação. Nesta pesquisa, é possível identificar algumas características demográficas, desejos, necessidades, idade, nível educacional, entre outros. Dessa forma, a etapa de prototipação que é a próxima, é apoiada por exemplos e informações gerais referentes aos dados estatísticos, projetos de design similares e as *personas* do projeto.

4.4 Prototipação

Uma das principais atividades do *Design Thinking* é a prototipação, atividade dedicada a projeções de funcionamento daquela solução ou de parte dela, sendo uma maneira prática, rápida e fácil de apresentar ideias para maior compreensão dos fluxos de projeto. A prototipação é um recurso que oferece aos designers maior entendimento do produto, servindo de método para explorar ideias preliminares de forma rápida, objetiva e econômica. Nos projetos deste estudo foi adotada a prototipação de alta

fidelidade, caracterizando de forma eficiente, protótipos de visão geral do sistema e também por perfis de acesso ao produto. É neste momento, já com distinção das arquiteturas definidas, que alguns pontos fundamentais devem ser consolidados, como por exemplo, o sistema de Grid que será utilizado, a cartela de cores, a tipografia, assim como ícones que serão utilizados, componentes, estilos e acessibilidade.

A definição de um sistema de grid adequado ao projeto permite a portabilidade do sistema para diferentes tipos de dispositivo, como desktop, tablet e/ou smartphone sem perder informações ou quebra de layout, apenas reorganizando os elementos para cada tipo de dispositivo. A **Figura 2** traz um exemplo desse planejamento da forma de visualização de interfaces para diferentes plataformas.

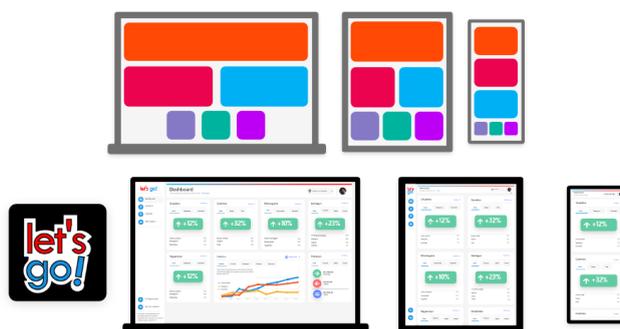


Figura 2: Planejamento de visualização de interfaces para diferentes plataformas

Em seguida, elementos visuais devem ser definidos, por exemplo, a cartela de cores que será utilizada. Para alguns casos já existe uma logomarca ou a preferência do cliente por determinado padrão de cores. Neste sentido, deve ser levado em consideração a teoria das cores, seus efeitos psicológicos no usuário e o seu significado, procurando sempre harmonizar a marca do cliente com o layout do aplicativo.

Um outro procedimento substancial é a tipografia, que é um elemento muito valioso para elaboração de um protótipo, através dela pode-se enriquecer visualmente um sistema. Com o auxílio do embasamento das cores é possível trabalhar o contraste da cor da fonte escolhida em relação ao fundo da página. Para esta definição, observa-se também a característica das *personas* identificadas na seção de pesquisas direcionadas. Por exemplo, um projeto que tem como público alvo idosos ou pessoas com pouca escolaridade, deve-se dar preferências a fontes maiores e menos rebuscadas para facilitar a leitura e compreensão.

A função dos ícones é servir como uma abreviação visual para uma determinada ação, conceito ou produto. Além de atuar como um item estético, os ícones são fundamentais na interação do usuário com o sistema tornando a aplicação mais amigável. A escolha dos ícones deve sempre respeitar a identidade visual que se deseja alcançar para o projeto. O tipo, tamanho e cores devem harmonizar com toda a aplicação. A **Figura 3 e 4** contém uma exemplificação da utilização de ícones e componentes.

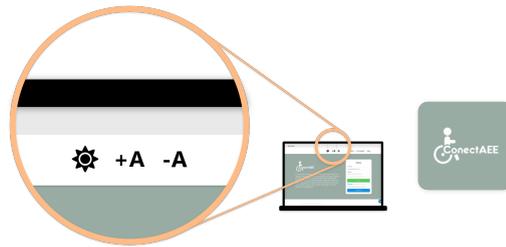


Figura 3: Exemplificação de ícones em um aplicativo

A definição dos componentes utilizados devem seguir os mesmos parâmetros anteriores, dessa maneira, Botões, *Checkboxes*, *Radio buttons*, *Dropdowns*, *Inputs*, *Breadcrumbs*, *Paginação*, *Slides*, *Image Carousel*, *Containers*, Estilos, informações no geral e *Acessibilidade*, assim como outros, devem seguir com similaridade os padrões estabelecidos para os recursos supracitados.



Figura 4: Exemplos de componentes utilizados em sistemas

4.5 Avaliação de Usabilidade

No decorrer do projeto, através das sprints semanais e das entregas realizadas, o time pode discutir constantemente, esclarecendo dúvidas, direcionamentos e seguindo orientações precisas para aproximar aquelas ações aplicadas do atendimento das necessidades dos usuários. Neste processo de avaliação de usabilidade, além de todos os testes e verificações que o designer realiza, outros membros do projeto também verificam se aquelas projeções estão condizentes com as necessidades do cliente ou usuário e se agregam valor na sua utilização, desde professores em funções de coordenação nos projetos até os demais desenvolvedores e envolvidos.

4.6 Validação do Protótipo

Esta etapa se refere a uma fase técnica de validação simples, mas essencial para produtividade e verificação dos protótipos propostos, uma vez que é o momento que um desenvolvedor e/ou o próprio designer verificam e ajustam a configuração do que foi projetado para a implementação real.

4.7 Integração com o Projeto

Essa é a última etapa deste processo, e é de extrema importância uma vez que o projeto será integrado às novas partes desenvolvidas. É imprescindível que o time de desenvolvimento esteja alinhado, uma vez que as atualizações acontecem em versões do



projeto já em andamento. Vai depender da divisão de atividades e equipe de desenvolvimento, sob quem ficará o papel de integrar, se de um desenvolvedor sênior, para avaliar aquele “release” ou se os demais membros do time.

4.8 Avaliação do Impacto da Proposta no Processo de Ensino e Aprendizagem da Engenharia de Software

A proposição de uma abordagem que considerasse todo o processo de desenvolvimento de aplicações, envolvendo clientes e projetos reais, com diversos papéis e práticas conhecidas na indústria de software, vem sendo realizada desde os primeiros sistemas que utilizaram algumas das atividades propostas neste trabalho.

Pode-se afirmar que a configuração proposta e experimentada em ambos laboratórios, aumentou a produtividade do time de desenvolvimento e também a qualidade final do produto, conseqüentemente tornou o ensino e aprendizagem mais lúdico para os discentes envolvidos. Permitiu também que o tempo de desenvolvimento fosse menor, uma vez que os requisitos e limitações eram identificados através das iterações e logo corrigidos e adaptados às necessidades impostas pelos cliente e usuários. Destaca-se também o fato de que novas ideias surgiam no decorrer de cada acompanhamento do cliente e ou usuários.

Esta proposta pode ser replicada a cenários completamente diferentes e também adequada a outros contextos, de forma que posso servir como referência. São atividades que podem ser adaptadas conforme preferências e necessidades de times de desenvolvimento, atendendo suas especificidades dentro dos seus respectivos processos de desenvolvimento de software.

Um fato interessante não menos importante é a formação de mão de obra qualificada para trabalhar em equipes com *Design Thinking*, não somente da parte do designer, mas também dos desenvolvedores e todos os demais stakeholders, uma vez que todos participaram do processo de prototipação iterativa com técnicas de DT. Os alunos de graduação que participam de um projeto real, com cliente real e com stakeholders diversos, participam de uma carga horária prática alta, que permite atuar imerso na realidade da sua área de formação, capacitando-os de maneira específica para o mercado de trabalho e incentivando a inovação e desenvolvimento de soluções com foco no usuário.

Com a abordagem proposta, os alunos participam assumindo diversos papéis nos projetos executados, como desenvolvedores, designers, testadores, gerentes de projetos, gerentes de configuração, entre outros. Além de interagir diretamente com os usuários/clientes, que passaram a atuar também como agentes ativos no processo de ensino e aprendizagem dos estudantes, propiciando aos mesmos uma formação holística.

5. Considerações Finais

A busca por utilização de técnicas e recursos na produção de software é contínua por parte da demanda pelo aprimoramento do processo de desenvolvimento, assim como com a qualidade do produto final. Dessa maneira, o conceito de *Design Thinking* surgiu como alternativa prática para apoiar o processo de ensino e aprendizagem na



produção de sistemas por parte de alunos do curso de Ciência da Computação, que compreendem melhor o modelo de negócio com foco no usuário, contribuindo para a criação e desenvolvimento de produtos e serviços competentes.

O objetivo deste artigo foi apresentar uma abordagem prática de *Design Thinking* como recurso fundamental para o processo de ensino e aprendizagem na Engenharia de Software, possibilitando uma formação técnica diversificada com interatividade e abordagem com foco no usuário. É composto por uma sequência de passos, utilizando de alguns métodos e artefatos para gerar novos artefatos de acordo com a experiência do usuário. E neste caso, estes passos e artefatos são acompanhados, avaliados e validados continuamente tanto pelo time de desenvolvimento. Esta proposta foi projetada nos últimos dois anos, através de práticas e experiências que acontecem em dois laboratórios de tecnologia da instituição que produzem software. Participaram dessa experimentação dez projetos de software, sendo cinco de cada laboratório, projetos estes que envolviam papéis, propósitos, tecnologias e arquiteturas diferentes.

Como resultado, o processo de ensino e aprendizagem foi adaptado dentro do contexto dos laboratórios de tecnologia para proporcionar uma formação holística aos discentes e um maior alinhamento ao mercado de trabalho. Acredita-se que os resultados deste trabalho permitem que outras instituições possam utilizá-lo como um tipo de guia de referência, não necessariamente para todo o processo, mas para atividades específicas que se adequam dentro do escopo daquele projeto e time de desenvolvimento. Qualquer instituição de ensino que tenha como componente curricular disciplinas de desenvolvimento, com a produção de software como soluções tecnológicas pode estudar e replicar uma ou algumas das atividades descritas, ampliando suas práticas e conhecimentos em seu processo de desenvolvimento através desta abordagem.

Como eventuais trabalhos futuros, esta abordagem pode ser analisada sobre diferentes perspectivas, como por exemplo, sobre o efeito do aprendizado de conceitos dos alunos e principalmente sobre a implicação desses conhecimentos e habilidades na absorção destes discentes pelo mercado de trabalho. E de forma mais técnica, como o impacto em assertividade de necessidades dos usuários e o impacto no usuário final. Por fim, uma avaliação e análise geral para projetos reais seria mais uma alternativa de contribuição.

Referências Bibliográficas

- BOURQUE, P. E FAIRLEY, R. E. *SWEBOK: Guide to the Software Engineering Body of Knowledge, Version 3.0*. **IEEE Computer Society**, Disponível em www.swebok.org. 2014.
- BROWN, TIM. *Change by Design: How Design Thinking Transforms Organizations and Inspires Innovation*. **Harper Business**. 2019.
- FAWCETT, P., FISHER, K., BISHOP, A. e MAGASSA, L. *Using design thinking to empower ethnic minority immigrant youth in their roles as information and technology mediaries*. In Proc. of **Extended Abstracts on Human Factors in Computing Systems**, p. 361-366. 2013.
- IDC. 2020. *International Data Corporation*. <https://www.idc.com/>. Acessado em junho de 2021.



- LINDBERG, T., MEINEL, C. E WAGNER, R. *Design Thinking: A Fruitful Concept for IT development?, Design Thinking: Understand – Improve – Apply, Understanding Innovation*. H. Plattner, C. Meinel e L. Leifer, Berlin, **Springer**, p. **3-18**. 2011.
- MARQUES, ANNE. B. CAVALCANTE, E., RIVERO, LOPES, A., CONTE, T. Aplicando design thinking para melhorar a qualidade de um aplicativo móvel. **Brazilian Symposium on Software Quality**. 2015.
- PMBOK. A Guide to the Project Management Body of Knowledge (PMBOK® Guide) – **Sixth Edition**. **Project Management Institute PMI**. 2017.
- PORTELA, C.; VASCONCELOS, A.; OLIVEIRA, S. R. B. Um Modelo Iterativo para o Ensino de Engenharia de Software Baseado em Abordagens Focadas no Aluno. In: **Anais do Simpósio Brasileiro de Informática na Educação-SBIE**. 2017.
- PRESSMAN, R. *Software Engineering: A Practitioner’s Approach*. **8. ed. New York, NY, USA: McGraw-Hill, Inc., ISBN 9780077697747**. 2014.
- ROSA, L. H. C., BERNARDI, G., MEDINA, R. D. Mundos Virtuais como apoio ao Ensino e Aprendizagem de Engenharia de Software: Simulando a Técnica de Entrevista. **RENOTE - Revista Novas Tecnologias na Educação**, v. 19, N 1, 2020.
- SOMMERVILLE, I. *Software Engineering*. **Pearson. International Computer Science Series. ISBN 9780137053469**. 2011.
- THORING, K. e MÜLLER, R. *Understanding the creative mechanisms of design thinking: an evolutionary approach*. In Proc. of Conf. on Creativity and Innovation in Design, p. 137-147. 2011.
- VANDERLEI, I. M.; ROCHA, R. G. C.; ARAUJO, J. C. T.; ANDRADE, M. J. P.; ALENCAR, A. F. Implantação de Laboratório de Tecnologias como Espaço de Formação Profissional e Melhoria no Processo de Ensino-Aprendizagem de Ciência da Computação. **RENOTE - Revista Novas Tecnologias na Educação**, v. 18, p. 11-20, 2020.
- VIEIRA, R., DE JESUS RAFAEL, G. e DA COSTA, J. *Design Thinking Experience*. In **Anais do Simpósio Brasil**. 2014.