

Construindo Jogos Educacionais Através de Linha de Produto

Diego Castro^{1,2}

¹ Programa de Engenharia de Sistemas e Computação
COPPE (Universidade Federal do Rio de Janeiro)
Rio de Janeiro – RJ – Brazil

² Centro Federal de Educação Tecnológica Celso Suckow da Fonseca (CEFET)
Campus Maria da Graça
Rio de Janeiro – RJ – Brazil

{diegocbcastro}@cos.ufrj.br

Resumo. *Numerosos estudos demonstram que os jogos possuem a capacidade de atrair a atenção dos alunos e são, portanto, utilizados para ensinar diversas disciplinas, sendo uma delas a Engenharia de Software (ES). O desenvolvimento de jogos é uma tarefa demorada que envolve vários colaboradores e estágios. Como resultado, um grande número de jogos educacionais são desenvolvidos a partir de jogos pré-existentes, o que é semelhante à abordagem oportunista de Reutilização de Software (RS). Além disso, existem diferentes tipos / versões de jogos, que em grande parte das vezes podem apresentar características semelhantes. A Linha de Produtos de Software (LPS) é uma área de pesquisa dentro da RS, que envolve uma metodologia projetada para consolidar sistemas que compartilham um conjunto comum de recursos para desenvolver softwares semelhantes. Portanto, o objetivo desta pesquisa é desenvolver uma LPS de jogos educacionais para agilizar o processo de desenvolvimento e expansão de jogos com foco em ES, disponibilizando a mesma para os docentes para que eles consigam gerar seus próprios jogos em um curto período de tempo e com mínimo esforço, podendo ainda expandir e reaproveitar jogos previamente criados por meio da LPS.*

1. Introdução

Vários estudos demonstraram que os jogos oferecem uma oportunidade significativa para melhorar o método de ensino passivo, que normalmente depende de apresentações de slides e aulas ministradas pelo professor, resultando na diminuição do envolvimento dos alunos na sala de aula [Steinkuehler e Squire 2024]. A maioria dos jogos possuem características visuais, interativas e práticas, que despertam o interesse do usuário e servem como fontes primárias de entretenimento e distração. Conseqüentemente, estes jogos apresentam uma oportunidade original e inovadora para reforçar o conhecimento dos alunos [Batista et al. 2022]. Devido a essas vantagens, é possível encontrar diversos jogos na literatura sendo utilizados para o ensino de diversas disciplinas, sendo a Engenharia de Software (ES) uma delas [Kosa et al. 2016].

Embora os jogos tenham o potencial de melhorar as práticas educacionais, o desenvolvimento de tais jogos é muitas vezes uma tarefa demorada e complicada, necessitando do envolvimento de numerosos especialistas e custando uma quantidade significativa de tempo e dinheiro [Bilińska-Reformat et al. 2020]. Como resultado da

natureza demorada e financeiramente custosa deste procedimento, os desenvolvedores de jogos educativos frequentemente priorizam a adaptação de jogos pré-existentes ao invés do desenvolvimento de novos jogos [Abbott 2019]. Essa prática de criar um produto a partir de outro pré-existente é muito semelhante à abordagem de Reutilização de Software (RS) oportunista [Ncube et al. 2008], onde o objetivo é modificar um produto existente para gerar um novo, sem seguir necessariamente um processo claramente definido. Esta abordagem minimiza principalmente o período de desenvolvimento de software, pois utiliza o processo de modificação de soluções existentes ao invés de construir novas. Além disso, vários jogos embora sejam de tipos diferentes possuem diversas características semelhantes, o que reforça ainda mais o conceito de RS.

A Linha de Produtos de Software (LPS) é uma área de pesquisa proeminente dentro da RS, que consiste em um conjunto de sistemas de software que apresentam características comuns que podem ser customizadas. Essa customização ocorre por meio da manipulação de pontos de variação, que são pedaços do software que podem ser modificados. O principal objetivo da LPS é melhorar a produtividade, minimizar o tempo de desenvolvimento e possibilitar a expansão de softwares pré-produzidos [Krueger 1992].

Desde a concepção à implementação, desenvolver jogos educativos pode ser complicado, portanto um professor com tempo limitado não conseguiria criá-los para seus alunos. Scratch, eAdventure, Alice e Roblox Studio, são vários motores de jogos que estão disponíveis e já foram utilizados com sucesso para simplificar este processo. No entanto, mesmo com estas ferramentas e conceitos, desenvolver jogos educativos envolve bastante esforço e pode sobrecarregar um professor [González García et al. 2019].

Esta pesquisa de doutorado pretende, portanto, desenvolver uma solução que permita explorar os conceitos de LPS no desenvolvimento de videogames educativos. Ao contrário do método convencional, que pode levar dias ou até meses e requer um investimento financeiro significativo, esta proposta visa disponibilizar uma LPS para docentes que permitirá criar jogos num curto espaço de tempo e com custos reduzidos, possibilitando professores sem conhecimento nenhuma em programação consiga criar seus jogos por meio de modificações direto em pontos de variação da LPS. Embora o foco da linha seja o desenvolvimento de jogos para ES, a plataforma está sendo construída de forma genérica, podendo ser utilizada por professores de qualquer disciplina.

O restante deste artigo é apresentado da seguinte forma: a Seção 2 apresenta os objetivos que o trabalho pretende alcançar, a Seção 3 demonstra a metodologia de pesquisa que está sendo usada no trabalho, a Seção 4 detalha a proposta em construção e, por fim, a Seção 5 conclui com as considerações finais e limitações

2. Objetivos de pesquisa

Os educadores estão constantemente explorando novas abordagens pedagógicas que integrem o entretenimento à educação, a fim de melhorar o envolvimento e a motivação dos alunos [Børte et al. 2023, Munna e Kalam 2021]. Um dos principais métodos que vem sendo utilizado com esse propósito é o ensino ativo, que tem como objetivo incentivar os estudantes a aprenderem de forma autônoma, participativa e prática. Existem diversas abordagens para utilizar esse método, como, por exemplo, simulação, experimentação, aprendizagem baseada em problemas, projetos e jogos, sendo a criação deste último o

foco desse trabalho, tendo como objetivo disponibilizar em um curto espaço de tempo e com custos reduzidos jogos em sala de aula, visando aumentar a motivação, engajamento e melhorar reforço do ensino [Børte et al. 2023, Diesel et al. 2017].

Como afirmado anteriormente, os jogos têm o potencial de proporcionar inúmeros benefícios para a educação. No entanto, o seu demorado processo de desenvolvimento representa um desafio à sua implementação generalizada nas salas de aula [Bellotti et al. 2013, González García et al. 2019]. Num esforço para mitigar este problema, numerosos educadores acabam por desenvolver os seus próprios jogos educativos inspirados em outros previamente construídos [Abbott 2019], se assemelhando ao processo de RS oportunista.

Nas revisões da literatura já realizadas [Castro e Werner 2021b, Castro e Werner 2023b], não foram encontradas ferramentas dedicadas à criação / expansão de jogos educacionais, ou seja, ferramentas que possibilitem a criação facilitada de jogos educacionais a partir de um previamente criado. Assim, esses jogos atualmente são criados de três formas [Hofman-Kohlmeyer 2019, Lee et al. 2020]: **(1)** Desenvolvendo o jogo do zero; **(2)** Usar kits de desenvolvimento de software (SDK) ou interfaces de usuário (UIs) fornecidos pelo desenvolvedor do jogo, o que só ocorre para grandes empresas e em pequena escala, ou; **(3)** Compreender o código-fonte e fazer as modificações necessárias, desde que o código fonte tenha sido disponibilizado, o que gera ainda um grande trabalho para o professor que precisa entender o código original para depois modificá-lo. Lembrando que esse trabalho se limita apenas a jogos digitais, ou seja, video games. Portanto, sempre que a palavra jogo for mencionada, ela está se referindo apenas a video games.

Tendo isso em mente, podemos concluir que o processo de desenvolvimento de jogos é caro e demorado, limitando muitos educadores a criar jogos para uso em suas aulas. Sendo este o problema que esta pesquisa visa explorar. Portanto, este trabalho tem como objetivo proporcionar economia de tempo e custos para o desenvolvimento de jogos educativos para ES, resultando em uma maior utilização de jogos em sala de aula adotando o conceito de LPS para conseguirmos criar os jogos em larga escala. Este objetivo geral pode ser decomposto em objetivos específicos: **[1]** Caracterizar o estado da arte sobre o uso de LPS para o desenvolvimento de jogos; **[2]** Desenhar o design da aplicação; **[3]** Implementar uma plataforma para desenvolvimento de jogos educativos por meio dos conceitos de LPS; **[4]** Realizar um estudo de avaliação da plataforma implementada.

3. Metodologia e Atividades

Esta pesquisa segue uma metodologia inspirada no modelo *Design Science Research* (DSR) [Peppers et al. 2007], seguindo três etapas básicas, sendo elas: conscientização, sugestão e solução. A Figura 1, apresenta um modelo em BPMN (*Business Process Model and Notation*) da metodologia utilizada. As cores verde, amarelo e branco demonstram respectivamente as atividades que já foram executadas, estão em andamento e que ainda não iniciaram. A seguir, cada uma das atividades são descritas com mais detalhes.

[1] Revisão da literatura preliminar: Etapa que realizou duas revisões. Ela procurou entender como estava o desenvolvimento de jogos pela comunidade e quais práticas de RS eram mais utilizadas para desenvolver jogos. Vale destacar

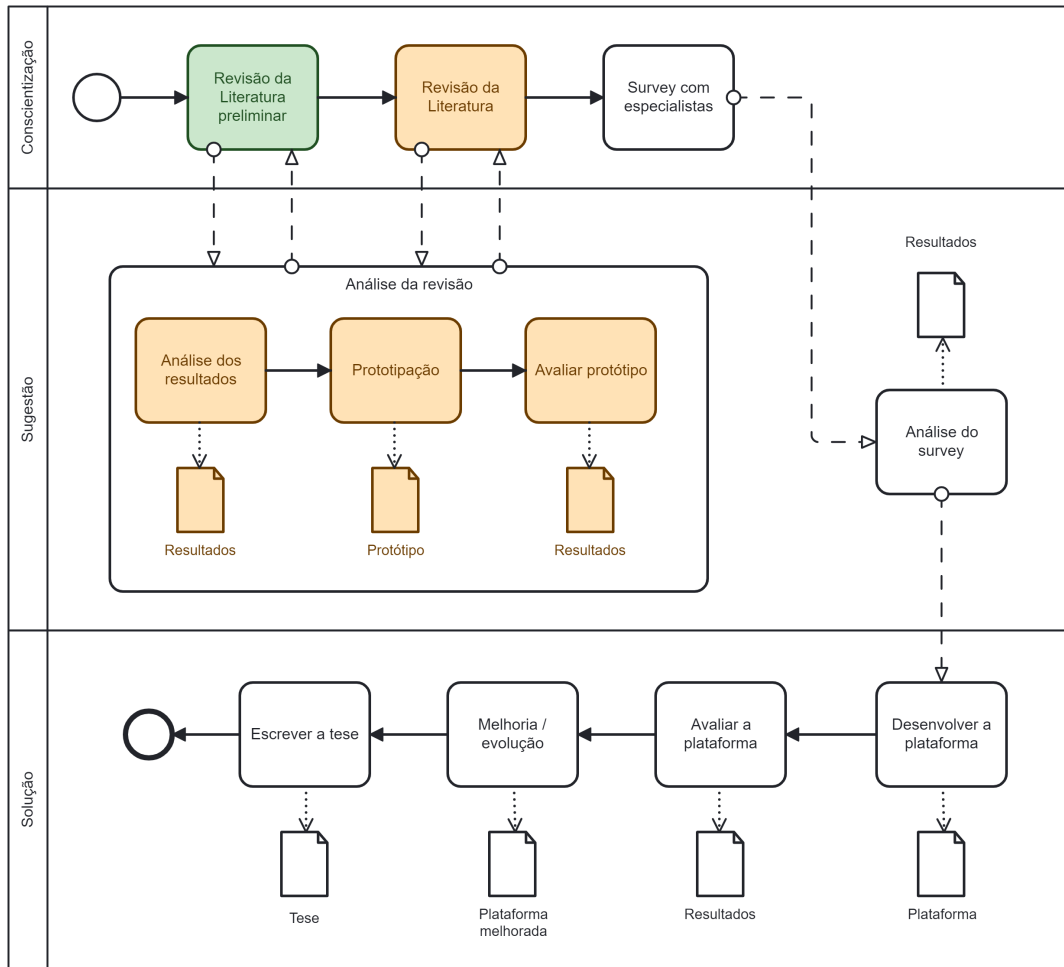


Figura 1. Metodologia desenhada em BPMN inspirada no modelo de DSR [Peffer et al. 2007].

que todo o processo de revisão seguiu o protocolo definido por Kitchenham et al. [Kitchenham et al. 2009]. **[1.1] Multivocal sobre mods e jogos:** Essa revisão buscou demonstrar como a comunidade de jogos costumam desenvolver seus jogos [Castro e Werner 2023b]. Lembrando que mods é um termo utilizado para se referenciar a jogos que sofreram uma alteração de forma a fazê-lo operar de forma diferente da original [Bilińska-Reformat et al. 2020], ou seja, é uma forma de RS oportunista sendo realizada pela comunidade de jogos. **[1.2] Mapeamento sistemático sobre o desenvolvimento de jogos por meio de RS:** Essa revisão buscou demonstrar como a RS pode ajudar no desenvolvimento de jogos [Castro e Werner 2020]; Ajudou a buscar a forma de reuso mais adequada para criação de jogos educacionais em larga escala. **[2] Revisão da Literatura:** Como mencionado, a segunda revisão demonstrou que LPS seria a melhor abordagem para expandir / criar variantes a partir de jogos já existentes e, por isso uma terceira revisão está sendo realizada focando em desenvolvimento de jogos educacionais por meio de LPS; **[3] Análise da revisão:** Subprocesso que busca analisar as revisões executadas e propor ideias a partir dos insumos encontrados. **[3.1] Análise dos resultados:** Após o término das revisões, cada um dos insumos encontrados são analisados. **[3.2] Prototipação:** A partir dos insumos encontrados nas duas primeiras

revisões, dois protótipos foram construídos para validar as informações encontradas [Castro e Werner 2021a]. **[3.2] Avaliar protótipos:** Após o desenvolvimento dos dois protótipos, eles foram validados por meio de estudos de usabilidade e experiência, por meio de modificações no questionário MEEGA [Petri et al. 2016]; **[4] Survey com especialistas:** Após a finalização da revisão mencionada no item 2, espera-se que seja executado um survey com especialistas para confirmar os insumos encontrados nas revisões e prototipações; **[5] Desenvolvimento da plataforma:** Após o término das etapas de conscientização e sugestão, espera-se ter insumos suficientes para a construção de uma plataforma para desenvolvimento de jogos por meio de LPS; **[6] Avaliar a plataforma:** Após o desenvolvimento da plataforma a mesma será avaliada em relação a sua usabilidade e experiência no desenvolvimento de jogos, comparando o tempo e o custo de desenvolvimento em relação as engines atuais [Castro e Werner 2023a]; **[7] Melhoria / evolução:** Após a avaliação da plataforma, será dedicado um tempo para o desenvolvimento das melhorias encontradas na avaliação; **[8] Escrita da tese:** Após todas as etapas anteriores será feita a escrita da tese e finalização desse trabalho.

4. Proposta

Existem uma grande variedade de tipos de jogos com diferentes características, sendo eles, jogos do tipo, *quiz*, *match*, *puzzle*, jogos de aventura, *storytelling*, entre outros. No entanto, ao pensarmos mais profundamente nas mecânicas de alguns jogos, é possível perceber semelhanças entre algumas delas, como, pontuação, barra de progresso, *feedback*, andar, pular, selecionar, entre outras. Com isso, é possível pensar em uma forma de agrupar algumas dessas características semelhantes e montar uma árvore de características com esses jogos, onde os mesmos vão se diferenciando por meio dos pontos de variação. No contexto de LPS, essa árvore de características pode ser ilustrada pelo modelo *Feature-Oriented Domain Analysis* (FODA) [Kang 2009], que é uma principais formas de ilustração de árvores de reutilização e visa agrupar as características de produtos semelhantes, demonstrar as características obrigatórias, excludentes e seus pontos de variação.

Para exemplificar a proposta iremos apresentar 3 tipos de jogos que são muito utilizados em contextos educacionais para demonstrar uma árvore de características criada com eles, sendo eles, jogos do tipo: *quiz*, *match* e *puzzle*. A seguir cada uma das características mais comuns desses jogos são brevemente descritas, lembrando que esses jogos podem possuir mais características, não limitando-se apenas às demonstradas nesse exemplo.

1. **Quiz:** Pontuação, barra de progresso, tempo, selecionar a resposta, turnos, feedback e progressão de informações;
2. **Match:** Arrastar e soltar, correlacionar, tempo, pontuação, turno, procurar a resposta certa, feedback e progressão de informações;
3. **Puzzle:** Arrastar e soltar, pontuação, tempo, turno, procurar a resposta certa, feedback e progressão de informações;

Como é possível notar, embora sejam jogos de tipos diferentes, muitas das características entre eles são semelhantes, podendo assim agrupar essas características e montar uma árvore com elas. A Figura 2 ilustra essa árvore do tipo FODA com algumas dessas características.

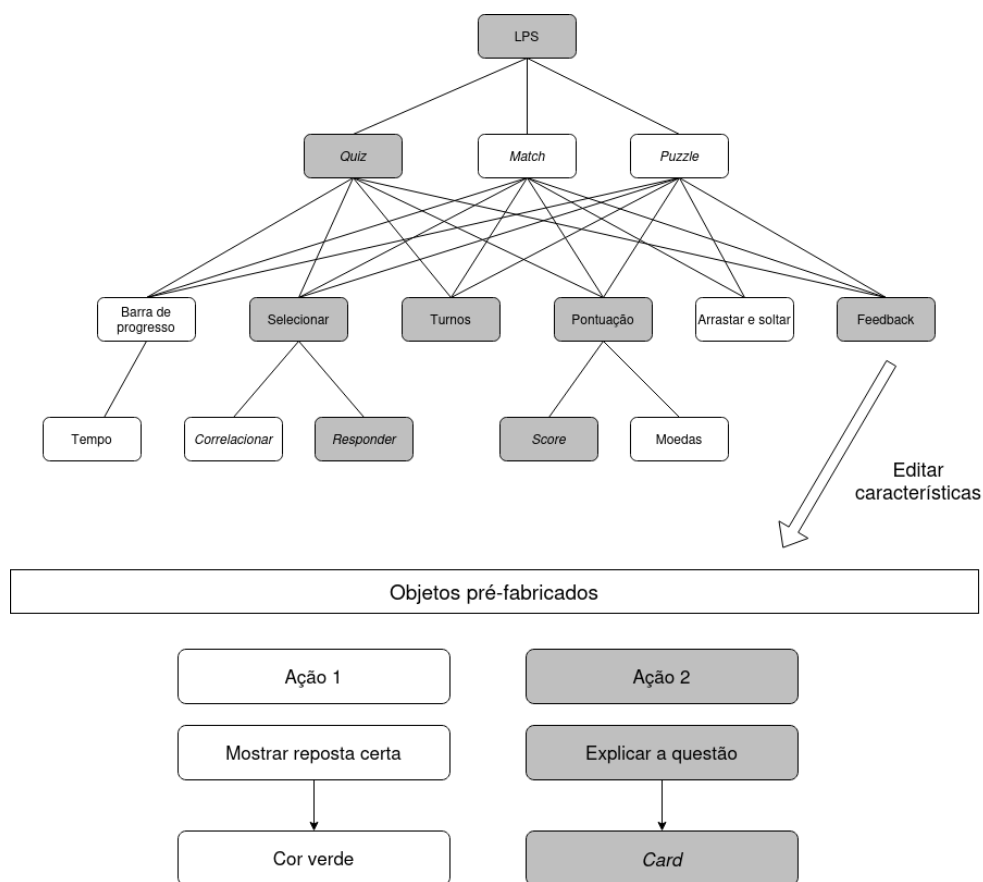


Figura 2. Árvore de características.

Na figura cada uma das características é representada por um nó (quadrado) e os pontos de variação sendo representados pelas arestas, ou seja, por exemplo, ao selecionar a característica de pontuação é possível escolher entre duas opções: *score* ou moedas, sendo este um ponto de variação. Também é possível perceber alguns nós em tons de cinza mais escuro, sendo esse um exemplo de um jogo do tipo *quiz* que poderia ser criado a partir da LPS, possuindo as características de responder as perguntas, feedback, turnos e ganhar pontuação ao longo do jogo.

Através da figura é evidente que vários jogos acabam compartilhando muitas mecânicas, com isso, o objetivo é desenvolver uma plataforma que disponibilize essa LPS para o professor, permitindo ao mesmo selecionar as características que deseja em seu jogo e então ter um jogo criado para sua turma, possibilitando ainda opções de evolução e edição da linha.

É possível ainda observar diversas características dos jogos que podem ser selecionadas, no entanto, para que seja possível criar diferentes jogos para o ensino de ES e deixar a plataforma mais genérica seria necessário permitir a edição e evolução dessas características. Uma linha de produto convencional é composta por componentes pré-fabricados que possam ser editados, com isso, existe a necessidade de permitir essa edição para o professor que esteja criando seu jogo a partir da plataforma. Portanto, todos os componentes da linha poderão ser editados diretamente pelo usuário permitindo que ele customize o que deseja.

Para ilustrar melhor, considere o exemplo do *quiz* já mencionado, no recurso de feedback, por exemplo, haverá uma ação indicando que o jogo deve exibir a resposta correta para o jogador que pode ser editada; no entanto, esse recurso de feedback foi modificado, adicionando uma nova ação para exibir uma explicação ao jogador caso ele erre a pergunta. A Figura 2, na parte de baixo, demonstra um exemplo da ideia de objetos pré-fabricados dentro da LPS. Nela, uma nova ação de explicar a questão está sendo adicionada.

5. Considerações Finais e Trabalhos Futuros

O objetivo deste artigo foi apresentar o andamento da proposta de uma tese sobre jogos educacionais e RS. Ao longo do trabalho foram descritas todas as etapas da metodologia utilizada para realização da pesquisa, como, revisões da literatura, prototipação, avaliação, desenvolvimento, melhorias, entre outras atividades. Algumas delas já foram totalmente ou parcialmente finalizadas e outras ainda precisam ser iniciadas.

Espera-se que ao final deste trabalho de pesquisa seja possível disponibilizar para os docentes uma LPS para o desenvolvimento, expansão e evolução de jogos educativos, possibilitando a criação de N jogos e N versões de jogos previamente criados em um curto espaço de tempo e com custos reduzidos. Espera-se que a ferramenta esteja pronta e avaliada até o final de dezembro, com previsão de defesa da tese de doutorado para março de 2025.

Referências

- Abbott, D. (2019). Modding tabletop games for education. In *Games and Learning Alliance: 7th International Conference, GALA 2018, Palermo, Italy, December 5–7, 2018, Proceedings 7*, pages 318–329. Springer.
- Batista, K., Costa, L. E. O., Oliveira, S. d., Bastos, L., Marins, P., Lima, L., Spiegel, C., Butcher, I., Vitiello, P., Palheiros, R., Quast, K., Jr., O. d. P., Xexéo, G., e Neri, F. (2022). *Jogos de Tabuleiro na Educação*. DEVIR, Brasil.
- Bellotti, F., Kapralos, B., Lee, K., Moreno-Ger, P., e Berta, R. (2013). Assessment in and of serious games: An overview. *Advances in human-computer interaction*, 2013:1–1.
- Bilińska-Reformat, K., Dewalska-Opitek, A., e Hofman-Kohlmeyer, M. (2020). To mod or not to mod—an empirical study on game modding as customer value co-creation. *Sustainability*, 12(21):9014.
- Børte, K., Nesje, K., e Lillejord, S. (2023). Barriers to student active learning in higher education. *Teaching in Higher Education*, 28(3):597–615.
- Castro, D. e Werner, C. (2020). Reusing and deriving games for teaching software reuse. *International Journal on Advances in Software*, pages 207–216.
- Castro, D. e Werner, C. (2021a). Rebuilding games at runtime. In *2021 36th IEEE/ACM International Conference on Automated Software Engineering Workshops (ASEW)*, pages 73–77. IEEE.
- Castro, D. e Werner, C. M. L. (2021b). A structured review of game coding through modeling. *Anais Estendidos do XX Simpósio Brasileiro de Jogos e Entretenimento Digital*, pages 309–313.

- Castro, D. e Werner, C. M. L. (2023a). Exploring product line concepts in game building. *2023 IARIA Annual Congress on Frontiers in Science, Technology, Services, and Applications*, pages 150–152.
- Castro, D. e Werner, C. M. L. (2023b). A multivocal review on derivation games. *2023 IARIA Annual Congress on Frontiers in Science, Technology, Services, and Applications*, pages 144–149.
- Diesel, A., Santos Baldez, A. L., e Neumann Martins, S. (2017). Os princípios das metodologias ativas de ensino: uma abordagem teórica. *Revista Thema*, 14(1):268–288.
- González García, C., Núñez-Valdez, E. R., Moreno-Ger, P., González Crespo, R., Pelayo G-Bustelo, B. C., e Cueva Lovelle, J. M. (2019). Agile development of multiplatform educational video games using a domain-specific language. *Universal Access in the Information Society*, 18(3):599–614.
- Hofman-Kohlmeyer, M. M. (2019). Players as content creators. the benefits of game modding according to polish users. *International Scientific Journal News*, 2:8–26.
- Kang, K. C. (2009). Foda: Twenty years of perspective on feature models. *the keynote of SPLC*.
- Kitchenham, B., Brereton, O. P., Budgen, D., Turner, M., Bailey, J., e Linkman, S. (2009). Systematic literature reviews in software engineering—a systematic literature review. *Information and software technology*, 51(1):7–15.
- Kosa, M., Yilmaz, M., O’Connor, R., e Clarke, P. (2016). Software engineering education and games: a systematic literature review. *Journal of Universal Computer Science*, 22(12):1558–1574.
- Krueger, C. W. (1992). Software reuse. *ACM Computing Surveys (CSUR)*, 24(2):131–183.
- Lee, D., Lin, D., Bezemer, C.-P., e Hassan, A. E. (2020). Building the perfect game—an empirical study of game modifications. *Empirical Software Engineering*, pages 1–34.
- Munna, A. S. e Kalam, M. A. (2021). Teaching and learning process to enhance teaching effectiveness: a literature review. *International Journal of Humanities and Innovation (IJHI)*, 4(1):1–4.
- Ncube, C., Oberndorf, P., e Kark, A. W. (2008). Opportunistic software systems development: making systems from what’s available. *IEEE Software*, 25(6):38–41.
- Peppers, K., Tuunanen, T., Rothenberger, M., e Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems*, 24:45–77.
- Petri, G., von Wangenheim, C. G., e Borgatto, A. F. (2016). Meega+: an evolution of a model for the evaluation of educational games. *INCoD/GQS*, 3:1–40.
- Steinkuehler, C. e Squire, K. (2024). Gaming in educational contexts. In *Handbook of educational psychology*, pages 674–695. Routledge.