

O *serious game* CodeCombat e o professor como mediadores da aprendizagem do pensamento computacional

Luís F. Severgnini¹, Eliana M. S. Soares¹

¹ Programa de Pós-Graduação em Educação – Universidade de Caxias do Sul (UCS)
CEP 95.070-560 – Caxias do Sul – RS – Brazil

{lfsevergnini, emsoares}@ucs.br

Abstract. *This paper presents a snippet of a case study which analyzes a workshop mediated by the serious game CodeCombat, with the goal of evaluating it as a mediator element of computational thinking learning. The theoretical guides are the Vygotskian mediation concept, the serious games potentialities and the computational thinking principles. Results show that although the game offers situations that allow computational thinking development, specially of decomposition and generalization principles, teacher's mediation is still very important for students to internalize actions taken within the game.*

Resumo. *Este artigo apresenta um recorte de um estudo de caso que analisa uma oficina mediada pelo serious game CodeCombat, com o objetivo de avaliá-lo como elemento mediador da aprendizagem do pensamento computacional. Utilizam-se como norteadores teóricos o conceito de mediação, as potencialidades dos serious games e os princípios do pensamento computacional. Os resultados demonstram que o jogo oferece situações que oportunizam o desenvolvimento do pensamento computacional, principalmente da decomposição e generalização. Apesar disso, a mediação do professor ainda é muito importante para que o aluno internalize as ações realizadas dentro do jogo.*

1. Introdução

Nos últimos anos, uma temática em particular tem ganhado notoriedade na academia: o pensamento computacional (PC). O PC é “uma abordagem para resolução de problemas que envolve o uso de um conjunto de práticas e princípios da Ciência da Computação para formular uma solução que seja executável por um computador” [Beecher 2017]. Por oferecer boas perspectivas para áreas como a matemática e a lógica, o PC tem recebido especial atenção nos planos curriculares do Ensino Básico de países como Estados Unidos, Itália, Inglaterra e Finlândia [Valente 2016]. Essa atenção também se deve à relevância do tema para o contexto tecnológico atual: cada vez mais precisamos compreender e criar tecnologias digitais em vez de apenas consumi-las e o PC é uma forma de endereçar esse problema [Severgnini 2018].

Em consonância com essa questão tecnológica, os jogos digitais também têm sido tema de pesquisa nas mais diversas áreas do conhecimento, desde saúde e publicidade, até matemática e computação. Uma parte dessa pesquisa parece convergir

para a investigação do PC e como os jogos digitais podem contribuir para a sua aprendizagem. Nesse sentido, observamos que existem diferentes abordagens de pesquisa no que se refere ao PC e aos jogos digitais. [Junior, Cavalheiro e Foss 2018] e [Melo et al. 2018], por exemplo, focam em aspectos como o *game design*, o teste e a avaliação de jogos educativos e *serious games*, a partir de uma perspectiva mais voltada aos princípios do PC; [Poloni 2018] e [Severgnini 2018], por outro lado, atribuem maior destaque às questões de aprendizagem do PC, apresentando um arcabouço teórico mais enraizado na educação do que na computação *per se*.

Neste artigo, apresentamos um recorte de um estudo de caso qualitativo em que analisamos um jogo digital em particular, o *serious game CodeCombat*, a partir de sua utilização em uma oficina de Introdução à Ciência da Computação, com o objetivo de verificar se ele pode ser um elemento mediador da aprendizagem do pensamento computacional. Para tanto, consideramos o conceito *vigotskiano* de mediação, levando em conta as potencialidades dos *serious games* e os princípios do pensamento computacional como nossos norteadores teóricos. Por conta disso, entendemos que os resultados apresentados neste artigo podem contribuir para a pesquisa relacionada aos processos de ensino e aprendizagem do pensamento computacional e ao uso de jogos digitais (principalmente os *serious games*) como mediadores desses processos.

2. Referencial teórico

2.1. *Serious games*

Os jogos sérios, mais conhecidos como *serious games*, são jogos cujo propósito primário não é entretenimento, prazer ou diversão [Michael e Chen 2005]. [Ritterfield, Cody e Vorderer 2009] acrescentam que *serious games* são desenvolvidos com o propósito de ensinar, treinar, promover hábitos saudáveis e mudança social.

Concordamos com o pesquisador e educador Damien Djaouti sobre as possibilidades e limitações dos *serious games* dependerem de cada jogo e do contexto de uso. Apesar disso, há um conjunto de possibilidades e limitações que os *serious games* compartilham [Djaouti 2016]. As possibilidades são: o aumento da motivação dos alunos, a aprendizagem por tentativa e erro, a observação das diferenças de ritmo de aprendizagem e o encorajamento das interações pedagógicas entre os alunos. Por outro lado, as limitações são: o uso de jogos sérios irrelevantes, a falta de integração do jogo no trabalho do professor e as restrições materiais e logísticas [Djaouti 2016; Habgood 2007].

2.2. Pensamento computacional

O pensamento computacional (PC) é um termo que foi popularizado por Jeannette [Wing 2006] para caracterizar uma forma de pensamento que utiliza conceitos e técnicas da Ciência da Computação para a resolução de problemas em diferentes contextos. O conceito de PC já existia muito antes de Wing e até mesmo de Papert, outro notório pesquisador da área. É possível encontrá-lo em Turing, Von Neumann, Boole, Lovelace, Pascal e outros pioneiros da computação. Não obstante, Wing conseguiu sintetizar de forma acessível um pouco daquilo que representa o PC: “o pensamento computacional é o processo de pensamento envolvido na formulação de um

problema e na expressão de sua(s) solução(ões) de tal forma que um computador – humano ou máquina – possa efetivamente executar” [Wing 2014].

Outro autor que aborda esse assunto é Karl Beecher, que, sob uma definição compatível com a de Wing, elenca os principais conceitos associados ao PC: pensamento lógico, pensamento algorítmico, decomposição, generalização e reconhecimento de padrões, modelagem, abstração e avaliação [Beecher 2017]. Devido a limitações de escopo, daremos ênfase a apenas dois desses conceitos: decomposição e generalização.

A decomposição é uma abordagem de resolução de problemas que prescreve a divisão de um problema complexo em partes mais simples e mais fáceis de lidar (no estilo *dividir e conquistar*). A generalização, por sua vez, consiste no reconhecimento de padrões e na aplicação de uma mesma solução à problemas similares. Esses dois conceitos foram utilizados como norteadores teóricos do PC em nossa análise.

2.3. Mediação *vigotskiana*

No que se refere à aprendizagem, tomamos como base o aporte teórico *vigotskiano*. Escolhemos Vigotski principalmente porque ele nos fornece um conceito que nos parece poderoso e compatível com o tema das tecnologias digitais na educação: a *mediação*. De um ponto de vista epistemológico, podemos dizer que mediação é a concepção de que o homem acessa e transforma a realidade por meio de elos intermediários, os instrumentos e signos [Vigotski 2009]. [Oliveira 2016, p. 26] elucida essa noção ao dizer que “enquanto sujeito do conhecimento, o homem não tem acesso direto aos objetos, mas um acesso mediado, isto é, feito através de recortes do real, operados pelos sistemas simbólicos de que dispõe”.

Portanto, quando nos referimos a um jogo como elemento mediador, denotamos que ele atua como um elo intermediário entre o jogador (sujeito) e uma certa realidade (objeto). Mais especificamente, vemos o jogo como mediador no sentido de que ele oferece tarefas desafiadoras que, aliadas às intervenções do professor, desencadeiam processos de *internalização*, principalmente quando ocorrem desde à Zona de Desenvolvimento Proximal (ZDP) [Vigotski e Cole 2007].

A internalização consiste na reconstrução individual (interna) de operações socioculturais e históricas (externas) [Vigotski 2007, p. 56]. No aporte teórico *vigotskiano*, internalizar e aprender expressam sentidos quase equivalentes. Logo, quando falamos em *internalização*, referimo-nos aos indícios de aprendizagem percebidos nos questionamentos, expressões, ações e reações dos alunos.

A ZDP, por sua vez, é uma noção *vigotskiana* que denota a distância entre o *desenvolvimento real*, o que um indivíduo consegue fazer por conta própria, e o *desenvolvimento potencial*, que se refere àquilo que esse mesmo indivíduo consegue realizar com o auxílio de alguém mais experiente [Vigotski 2007; 2009]. Conforme o autor [2007; 2009], o melhor ensino ocorre na ZDP, isto é, se adianta ao *desenvolvimento real* dos alunos ao mesmo tempo em que não ultrapassa seu *desenvolvimento potencial*.

3. Método

Consideramos que esta pesquisa tenha uma natureza qualitativa, pois nela examinamos em profundidade e extensão as qualidades de um certo fenômeno, o uso de um *serious game* como mediador de aprendizagem do pensamento computacional. Para [Lüdke e André 2013], a pesquisa de abordagem qualitativa se caracteriza principalmente por utilizar-se do ambiente natural como fonte direta de dados e do pesquisador como seu principal instrumento, o que pressupõe um convívio prolongado do pesquisador no ambiente investigado, resultando na alteração desse ambiente. Por isso, a consciência dessa condição é imprescindível para que não se invalidem os resultados da pesquisa.

3.1. CodeCombat: um estudo de caso

Como procedimento metodológico desta pesquisa, escolhemos o estudo de caso. De acordo com [Yin 2010, p. 39], um estudo de caso é “uma investigação empírica que a) investiga um fenômeno contemporâneo em profundidade e em seu contexto de vida real, especialmente quando b) os limites entre o fenômeno e o contexto não são claramente evidentes”. A escolha do estudo de caso se justifica uma vez que tínhamos a intenção de fazer uma investigação com tais características.

Constituímos nosso estudo de caso a partir de uma Oficina de Introdução à Ciência da Computação mediada pelo *serious game CodeCombat*, um jogo educativo de aventura em que os jogadores comandam as ações dos personagens por meio de algoritmos. Para um melhor entendimento dos procedimentos metodológicos, descreveremos primeiramente o *serious game*, seguido da oficina. Por último, descreveremos a geração do *corpus* de pesquisa e o procedimento de análise.

3.2. O *serious game*

CodeCombat é um jogo sério educativo de aventura que visa principalmente a aprendizagem de programação de computadores. Diferentemente de plataformas como *Scratch* e *Code.org*, em que o código é produzido através de blocos de arrastar e soltar (*drag and drop*), os programas produzidos pelos alunos são escritos em linguagens de programação reais (*Javascript* ou *Python*), tal como costumam ser em um contexto real de uso.

Em vez de apenas clicar e pressionar botões (des)ordenadamente, esse *serious game* convida o jogador a pensar, reconhecer os problemas, formular hipóteses, escrever possíveis soluções e verificar se elas resolvem o problema. À primeira vista, isso pode parecer um tanto quanto entediante ou enfadonho, mas *CodeCombat* procura entregar uma experiência divertida, uma vez que foi criado para “dar aos alunos a sensação de poder de um mago na ponta de seus dedos enquanto digitam códigos” [CodeCombat 2019]. O jogador assume o papel de um herói que derruba seus inimigos usando algoritmos como lâmina de sua espada. Não se trata de resolver problemas descontextualizados, como ocorre em alguns jogos de *puzzles*; em vez disso, os jogadores são imersos em uma aventura que começa devagar, mas que, aos poucos, se torna desafiadora. Escolhemos esse jogo pois confiávamos que, em conjunto com a mediação do educador, ele possibilitaria que a oficina levasse o conceito de *serious*

game educativo ao limite de suas potencialidades, oferecendo oportunidades de aprendizagem de uma forma lúdica.

Um dos cenários do jogo capaz de ilustrar essa hipótese é o nível “O Prisioneiro” (ver Figura 1), oitavo nível do primeiro mundo (curso) de *CodeCombat*. Tal como os níveis que o antecedem, “O Prisioneiro” tem objetivos claramente definidos, que são: libertar o prisioneiro, derrotar o guarda e pegar o diamante. No entanto, o guarda que protege a *dungeon* é mais forte e resistente que aqueles encontrados pelo jogador até então, de modo que o herói não consegue derrotá-lo sozinho. Além disso, a ajuda do prisioneiro não garante uma investida bem-sucedida, pois um arqueiro como ele não tem nenhuma chance em um duelo de curta-distância. Assim, o jogador precisa elaborar uma solução que evite um confronto direto individual com o inimigo e que lance mão da capacidade do arqueiro de atacar à distância. Ou seja, não somente deve o jogador identificar os objetivos e resolver um problema estático, como ele precisa considerar o dinamismo do ambiente e o comportamento inesperado dos inimigos ao elaborar sua estratégia, valendo-se tanto de seu pensamento computacional quanto de sua criatividade.



Figura 1. O Prisioneiro: além de pensamento computacional, o nível exige criatividade e imaginação do jogador. À esquerda, o cenário do jogo; à direita, a solução para o problema escrita por um aluno na linguagem Python.

3.3. A oficina

Para a realização da pesquisa, selecionamos alunos do 9º ano do Ensino Fundamental, de uma escola pública municipal, com idade entre 13 e 15 anos. Entendemos que é importante fomentar o desenvolvimento do pensamento computacional o mais cedo possível, por isso escolhemos tal faixa etária. Os encontros da oficina foram realizados aos sábados, no laboratório de informática da escola, durante o período da manhã, para que não houvesse colisão com a grade curricular prevista aos alunos. Foram realizados quatro encontros, com duração de uma hora cada. Ao todo, a oficina contou com a presença de nove participantes, dos quais seis mantiveram regularidade. Ressaltamos também que a oficina só teve seu início mediante a anuência da direção da escola e a autorização do Comitê de Ética em Pesquisa. Além disso, a participação dos sujeitos de pesquisa estava condicionada à apresentação de autorização dos responsáveis.

A pauta de cada encontro era definida com base na ementa do primeiro mundo do jogo, que se apresenta como um curso de Introdução à Ciência da Computação. No que se refere à programação, os conceitos foram abordados na seguinte ordem: sintaxe básica, algoritmos, argumentos (no uso de funções já disponibilizadas pelo jogo), *strings* (cadeias de caracteres), laços de repetição (*while loop* infinito) e variáveis.

Além de explorar o jogo, tanto individualmente quanto em grupos, propomos aos sujeitos atividades como a *programação em pares*, que previa um aluno como *motorista* e outro como *navegador* (e vice-versa). Ademais, observamos a atividade do grupo, atentos a quaisquer dúvidas, comentários e dificuldades que surgissem.

3.4. A geração do *corpus* de pesquisa

Geramos os primeiros dados logo no início do primeiro encontro da oficina, por meio de um questionário não-avaliativo, que tinha como objetivo apenas caracterizar e conhecer o perfil dos sujeitos, seus gostos e preferências no que se refere aos jogos digitais. Esses dados foram utilizados principalmente para a operacionalização da oficina, para que pudessemos adequar nossa abordagem e o ritmo dos encontros.

As soluções dos alunos para os problemas do *serious game CodeCombat* – dados de maior relevância para esta pesquisa – foram registradas pelo próprio jogo e ficaram disponíveis no “Painel do Professor”, área restrita que permite o acompanhamento individual dos alunos, bem como das turmas. Nessa área, encontram-se as soluções escritas pelos alunos, em cada nível, e indicadores de progresso dos alunos no curso como um todo (quais níveis foram concluídos, quanto tempo foi gasto em cada nível, quais conceitos de programação foram trabalhados).

Além disso, documentamos as atividades por meio de anotações e filmagens. As anotações foram feitas para registrar a percepção do mediador durante os encontros, dirigindo-se aos movimentos dos alunos, seus questionamentos e a maneira como trabalhavam entre si. As filmagens, por sua vez, foram realizadas com o intuito de registrar todas as interações significativas que ocorressem na oficina, principalmente as que não percebêssemos em tempo real.

3.5. Procedimento de análise

A partir de nosso referencial teórico, estabelecemos os seguintes norteadores teóricos para realizarmos a análise do *corpus* de pesquisa: mediação, ZDP e sociointeração (interação entre pares), no que se refere à aprendizagem; as possibilidades e limitações dos *serious games*; e os conceitos de decomposição e generalização para abordar o PC. Reconhecemos a existência de vários outros norteadores, dentro e fora de nosso referencial teórico, mas escolhemos apresentar apenas estes no escopo deste artigo.

Dada a diferente natureza dos dados que compunham o *corpus* de pesquisa, organizamos nossa análise em duas etapas:

- a) análise das soluções (programas) dos alunos, à luz dos norteadores teóricos do PC, decomposição e generalização. Para tanto, analisamos cada solução, linha a linha, desde o primeiro nível até “O prisioneiro”, e observamos sua respectiva execução no cenário do jogo, de modo a identificar relações entre os desafios e

sub-desafios com o conceito de decomposição, assim como padrões existentes nesses níveis do jogo;

- b) análise das anotações e gravações das oficinas, com base nos norteadores teóricos *vigotskianos* e dos *serious games*, observando principalmente de que formas o jogo *CodeCombat* oportunizou acesso ao objeto de conhecimento (PC), em quais situações a mediação do professor foi necessária e como os participantes interagiram entre si.

4. Resultados e discussão

Na análise das soluções desenvolvidas pelos alunos nos níveis do jogo, encontramos indícios de decomposição e generalização, dois conceitos relacionados ao PC. O conceito de decomposição foi percebido em pelo menos dois níveis: no nível “4a - *Distração Ilusória*” (ver Figura 2), em que o jogador precisa distrair os guardas para pegar o tesouro; e no nível “8 - *O Prisioneiro*”, em que o herói precisa libertar seu aliado para enfrentar um poderoso inimigo. No nível 4a, o jogador não pode ir diretamente em busca do tesouro, pois acabaria derrotado. Assim, precisa decompor o problema “pegar o tesouro” em dois problemas menores: distrair os guardas e encontrar o tesouro. Do mesmo modo, o nível 8, detalhado anteriormente, exige que o jogador decomponha o problema maior, que é “derrotar o inimigo e pegar o tesouro”, em três problemas menores: libertar o prisioneiro, derrotar o inimigo e pegar o tesouro.



Figura 2. Nível “4a - Distração Ilusória”

Diferentemente da decomposição, a generalização somente se evidencia na perspectiva nível a nível, quando observamos as transições de um nível a outro. Por exemplo, reconhecemos um padrão na transição do nível “4 - *Mina Inimiga*” ao 4a: o conceito de argumentos. O uso de um argumento na função *hero.move()* permite que o herói se movimenta *N* vezes com um único comando. Para garantir que o novo conceito seja percebido pelo jogador pelo menos uma vez, *CodeCombat* exige que esse nível seja resolvido em no máximo cinco instruções; para tanto, é necessário utilizar argumentos nas instruções dadas ao herói, explicitando a quantidade de vezes que ele deve se movimentar. No nível subsequente 4a, a mesma exigência não é feita ao jogador; porém, percebemos na análise das soluções que alguns jogadores reconheceram o padrão de movimentação de imediato e passaram a utilizá-lo, mesmo quando não havia exigência por parte do jogo (ver Figura 3). Outro indício de generalização do conceito de argumento é o questionamento dos alunos em relação ao comando *hero.attack()* que, diferentemente do comando *hero.move()*, não possui um argumento para indicar a

quantidade de repetições da instrução. Verificamos o uso de padrões como esse em vários outros momentos ao longo do jogo.

| Nível 4: Enemy Mine | Nível 4a: Illusory Interruption | Nível 5a: Favorable Odds |
|--|--|--|
| <pre># Use argumentos nos comandos hero.moveRight(3) hero.moveUp() hero.moveRight() hero.moveDown(3) hero.moveRight(2)</pre> | <pre># Ative a isca e pegue a gema. Caso pi hero.moveRight() hero.moveDown(2) hero.moveUp(2) hero.moveRight(3)</pre> | <pre># Ataque os dois ogros e pegue hero.moveRight() hero.attack("Krug") hero.attack("Krug") hero.moveRight() hero.moveUp() hero.attack("Grump") hero.attack("Grump") hero.moveLeft(2)</pre> |

Figura 3. Soluções de um participante da oficina para os níveis 4, 4a e 5a, escritas em Python.

Complementamos a análise das soluções dos alunos com a análise das anotações e gravações da oficina. Percebemos que o jogo respeita a forma de pensar particular de cada um, não impondo uma abordagem de resolução de problemas específica (salvo em exceções). Outro aspecto relevante percebido na análise das gravações é que o ritmo da oficina era determinado pelo progresso dos alunos no jogo. Assim, os alunos que levavam mais tempo para resolver os níveis não precisavam se preocupar em “ficar para trás”, ao passo que os alunos mais velozes não precisavam esperar para avançar no jogo. Como mediadores, vimos essa diferença no ritmo de aprendizagem como uma oportunidade para encorajar a sociointeração, as interações pedagógicas entre os pares, de modo que os alunos mais avançados auxiliassem seus colegas, no papel de outro mais experiente em conjunto com o professor-mediador (ver Figura 4, lado direito).



Figura 4. (E) Professor-mediador problematiza nível do jogo com intuito de atuar na ZDP do aluno. (D) Sociointeração: um aluno mais experiente auxilia seu colega a resolver problemas.

Constatamos também que as dúvidas apresentadas pelos alunos costumavam ser sobre como comunicar ao jogo o *plano* que eles haviam traçado mentalmente. Nesses casos, dialogamos com eles e sugerimos que expressassem suas ideias em palavras ou até mesmo apontando para a tela. Dessa forma, foi possível atuar de uma maneira que os ajudasse sem resolver o problema em seu lugar e que, ao mesmo tempo, possibilitasse que eles percebessem que já haviam encaminhado boa parte da solução – uma abordagem coerente com a proposta *vigotskiana*.

Ainda nas gravações, observamos que o *feedback* do *serious game* propicia a resolução dos problemas por meio de tentativa e erro, permitindo que um aluno formule

hipóteses e as teste em seguida. Embora entendamos que essa característica tenha bastante potencial de mediação, como aponta [Djaouti 2016], ela possibilita que um problema seja resolvido mesmo quando o jogador não o compreende. Se o problema não for compreendido, é improvável que haja internalização. Ao mesmo tempo, para que haja decomposição, um dos princípios do PC, é necessário que o problema seja compreendido por completo. A partir de [Vigotski 2009], entendemos que, para possibilitar uma melhor compreensão, o *serious game* também precisa atuar na ZDP do jogador. O jogo *CodeCombat* adapta os caminhos que cada jogador deve percorrer ao longo da aventura, oferecendo *feedback* dinâmico quando percebe alguma dificuldade do aluno ou erro em sua solução. O *serious game* também propõe níveis complementares aos jogadores quando julga que eles precisam aprofundar um determinado conceito. Porém, a dinamicidade desse *feedback* é limitada, uma vez que o jogo não consegue dialogar com o aluno e, portanto, não consegue adequar-se à sua ZDP. O professor, por outro lado, consegue fazê-lo (ver Figura 4, lado esquerdo). Assim, parece-nos que a mediação do professor é importante para complementar a mediação do *serious game*, pois corrobora a compreensão do aluno em relação aos desafios do jogo, possibilitando que ele decomponha tais problemas e internalize os conceitos empregados em suas soluções.

5. Conclusão

CodeCombat é um *serious game* cujo *design* convida seus jogadores a vivenciarem aspectos relacionados ao pensamento computacional: formular problemas e expressar suas soluções de modo que um computador possa executá-las [Wing 2014]. Não somente isso, mas o faz propiciando que os alunos fiquem motivados e possam formular e testar hipóteses com um *feedback* instantâneo, o que respeita o ritmo de aprendizagem de cada um e ainda fomenta interações e mediações pedagógicas que podem desencadear o processo de internalização.

Nesse sentido, considerando nosso convívio intenso no ambiente investigado e de sua resultante alteração, concluímos que o *serious game CodeCombat*, em conjunto com o professor-mediador, que auxilia o aluno a significar as ações realizadas no jogo, levará à aprendizagem do pensamento computacional, principalmente dos aspectos de decomposição e generalização.

Ressaltamos também que a abordagem considerada nesse estudo, relativa à mediação *vigotskiana*, pode ser ampliada para além de contextos onde o *CodeCombat* ou outro *serious game* está presente. Propomos que esse resultado possa ser transposto a ambientes onde existam outros recursos, como objetos de aprendizagem, jogos de entretenimento e ferramentas no estilo *Scratch*, e até mesmo para atividades de computação desplugada.

Finalmente, ressaltamos que o desafio de ser professor-mediador nessa perspectiva é sair do papel de quem explica e dá instruções, para o de alguém que faz intervenções que levam o aluno a descobrir por si mesmo os caminhos para a solução dos problemas, viabilizando que ele seja autor de seu processo de aprender. Essa é uma linha tênue, por isso, um desafio para o discernimento do professor-mediador.

Referências

- Beecher, K. (2017), Computational Thinking: A beginner's guide to problem-solving and programming, BCS.
- CodeCombat. (2019) “Sobre”, <https://br.codecombat.com/about>, Junho.
- Djaouti, D. (2016). Serious Games pour l'éducation : utiliser, créer, faire créer ?. In *Tréma*, pages 51-64. OpenEdition.
- Habgood, M. (2007), The Effective Integration of Digital Games and Learning Content. Tese (Doutorado) - University Of Nottingham, Nottingham, Reino Unido.
- Junior, B. A., Cavalheiro, S. e Foss, L. (2018). Uma análise de um jogo educacional sob a ótica do Pensamento Computacional. In *Proceedings of the SBIE 2018*.
- Lüdke, M. e André, M. (2013), Pesquisa em educação: abordagens qualitativas, Epu, 2ª ed.
- Melo, D., Pires, F. e Ferreira, R. (2018). Robô Euroi: jogo de estratégia Matemática para exercitar o Pensamento Computacional. In *Proceedings of the SBIE 2018*.
- Michael, D. e Chen, S. (2005), Serious Games: Games that Educate, Train, and Inform, Cengage Learning Ptr, 2ª ed.
- Oliveira, M. (2016) “Vygotsky e o processo de formação de conceitos”, Piaget; Vygotsky; Wallon: teorias psicogenéticas em discussão, La Taille, Y., Oliveira, M. e Dantas, H., Summus Editorial, 27ª ed., p. 23-34.
- Poloni, L. (2018), Aprendizagem de programação mediada por uma linguagem visual: possibilidade de desenvolvimento do pensamento computacional. Dissertação (Mestrado) - Universidade de Caxias do Sul.
- Ritterfield, U., Cody, M. e Vorderer, P. (2009), Serious games: Mechanisms and effects, Routledge.
- Severgnini, L. F. (2018). Alice e o Mistério dos Algoritmos: um *serious game* como ferramenta de aprendizagem de lógica de programação para crianças. In *Renote*.
- Valente, J. A. (2016). Integração do pensamento computacional no currículo da educação básica: diferentes estratégias usadas e questões de formação de professores e avaliação do aluno. In *Revista E-curriculum*, pages 864-897.
- Vigotski, L. S. e Cole, M. (org.). (2007), A formação social da mente: O desenvolvimento dos processos psicológicos superiores, Martins Fontes, 7ª ed.
- Vigotski, L. S. (2009) A construção do pensamento e da linguagem, Martins Fontes, 2ª ed.
- Wing, J. (2006). Computational thinking. In *Communications Of The Acm*, pages 33-35. Association for Computing Machinery (ACM).
- Wing, J. (2014) “Computational Thinking Benefits Society”, <http://socialissues.cs.toronto.edu/index.html%3Fp=279.html>, Junho.
- Yin, R. K. (2010), Estudo de caso: planejamento e métodos, Bookman, 4ª ed.