

A Utilização do Scratch como Ferramenta no Ensino de Pensamento Computacional para Crianças

Alexandre Hild Aono¹, Hugo Vianna Silva Rody¹, Daniela Leal Musa¹,
Vanessa Andrade Pereira¹, Jurandy Almeida¹

¹Instituto de Ciência e Tecnologia, Universidade Federal de São Paulo – UNIFESP
12247-014, São José dos Campos, SP – Brasil

{alexandre.aono,hugo.rody,musa,vapereira,jurandy.almeida}@unifesp.br

Abstract. *This paper presents an initiative of teaching computational thinking for elementary school students through an expository methodology combined with the use of Scratch as an accessible and intuitive learning environment. The proposed learning method relies on a theoretical explanation about computational thinking-related topics and then on practical activities where the students developed parts of a game that were merged into a final project. This experiment has enabled us to evaluate methodologies for learning computer programming and using of games as a motivating pedagogical tool.*

Resumo. *Este artigo apresenta uma iniciativa de ensino de pensamento computacional para estudantes de uma escola de ensino fundamental com a utilização de uma metodologia expositiva aliada à utilização do Scratch como um ambiente de aprendizado intuitivo e acessível. O método de aula adotado consistiu em uma explicação teórica sobre um tópico relacionado à aquisição de pensamento computacional seguida de atividades práticas de programação nas quais os alunos construíram partes de um jogo que posteriormente foram reunidas em um projeto final. Este experimento permitiu avaliar metodologias para ensino de programação e utilização de games como ferramenta para incentivo pedagógico na área.*

1. Introdução

O Pensamento Computacional está influenciando a pesquisa em quase todas as áreas, tanto em Ciências quanto em Humanidades [Bundy 2007]. Os conceitos básicos de Computação – abstração, decomposição, entre outros – são de fundamental importância na formação do cidadão, provendo recursos cognitivos necessários à resolução de problemas nas mais diversas áreas [Wing 2008]. De acordo com França et al. (2012), a lista de habilidades e conhecimentos necessários para o pleno exercício da cidadania neste século é extensa, incluindo o chamado Pensamento Computacional, enfatizando que esse, talvez, seja o mais importante e menos compreendido conhecimento.

Os benefícios educacionais de ser capaz de pensar computacionalmente – começando com o uso de abstrações – reforçam e melhoram as habilidades intelectuais e, portanto, podem ser transferidas para qualquer domínio [Wing 2011]. O pensamento computacional envolve a resolução de problemas, a criação de sistemas e o entendimento do comportamento humano por intermédio dos conceitos fundamentais da Ciência da Computação, incluindo uma gama de ferramentas mentais que refletem a amplitude da área de Ciência da Computação [Wing 2006]. Nesse ínterim, Wing (2006) aponta que pensar como um cientista da computação significa mais do que ser capaz de programar

um computador; isso requer pensar em múltiplos níveis de abstração.

De acordo com a Associação de Professores de Ciência da Computação [Stephenson et al., 2005], o conhecimento em Ciência da Computação é essencial ao atual aluno de ensino básico, assim como as outras ciências tradicionais, proporcionando benefícios diretos aos estudantes. Introduzir o ensino de computação a alunos de ensino básico por intermédio do pensamento computacional tem se mostrado apropriado, pois o aluno irá adquirir habilidades de pensamento focadas em aspectos que não se limitam à programação [Wing 2008]. Nesse contexto, o pensamento computacional e o raciocínio lógico deveriam ser ensinados desde cedo, já que aumentam a capacidade de dedução e conclusão de problemas por intermédio do desenvolvimento de uma solução para uma dada situação [França et al. 2012, Scaico et al. 2013].

Barcelos et al. (2012), aponta que a falta de domínio adequado de conhecimentos matemáticos pelos alunos é um possível fator explicativo para a falta de interesse e evasão em cursos da área de computação. Além disso, a desmotivação para inserção na área por parte dos estudantes de ensino básico pode ser atribuída ao fato de que, como essa área não é um componente explorado nos currículos escolares, muitos estudantes possuem ideias incorretas e atitudes negativas sobre a área, percebendo como chatas e entediadas as carreiras neste campo [Scaico et al. 2013].

O ensino de Pensamento Computacional se mostra, além de motivador para inserção dos estudantes de ensino básico em Computação, uma ferramenta para o desenvolvimento de habilidades relacionadas à construção do pensamento de maneira lógica, tornando o ensino de programação mais acessível para um maior número de indivíduos e estimulando muitas capacidades cognitivas [Scaico et al. 2013].

Visando verificar o processo de ensino e aprendizagem em Pensamento Computacional, propusemos um curso de introdução a esses conceitos utilizando uma abordagem expositiva em conjunto com um “Ambiente Introdutório de Ensino” [Fincher et al. 2010] pelo qual os participantes puderam utilizar os conceitos de programação aprendidos para construção de um jogo, tornando a aquisição de conhecimento acessível e motivadora. A cada novo tópico os alunos desenvolveram uma subparte do jogo, relacionando a utilização da programação com os passos para criação de um jogo.

2. Fundamentação Teórica

A maior fraqueza que aprendizes de programação enfrentam são relacionadas a atividades para resolução de problemas, criação de soluções e suas respectivas modelagens na forma de programas [de Kereki et al. 2008].

A possibilidade de criar um jogo digital encanta e atrai cada vez mais adeptos para aprendizagem computacional. Em 2016, cresceram os cursos que oferecem ensino de computação com conteúdo orientado ou pautado por jogos no mercado e se faz cada vez mais presente a questão da inclusão do ensino de programação na grade curricular.

Os jovens de hoje são, como define Marc Prensky (2012), “nativos digitais”, ou seja, eles nasceram e foram criados em ambientes dominados pela tecnologia digital. Sua relação com a tecnologia é espontânea e intuitiva, fato que pode ser utilizado para trazer benefícios e apoiar a prática da aprendizagem em vários campos incluindo o pensamento lógico utilizado em linguagens de programação.

Respeitando a condição desses jovens é possível destacar os esforços de alguns

acadêmicos em fazer uso do jogo digital como aliado da aprendizagem [Soares, Petarnella (orgs.) 2012, Prensky 2012, Alves e Nery (orgs.) 2015, Alves e Coutinho (orgs.) 2016]. A intenção de conciliar aprendizagem com entretenimento já era uma questão relevante para teóricos clássicos da pedagogia [Vygotsky 1984, Wallon 1979, Piaget 1973] e continua em pauta [França 2015].

No entanto, a aquisição de conhecimentos computacionais suficientes para construção de um jogo pode representar uma barreira, pois sem habilidades necessárias para tal, esse processo de motivação pode se tornar ineficiente, ocasionando frustrações e desmotivando a aprendizagem de computação [França, 2015]. Além de que, conforme aponta Scaico (2013), a inserção de programação com abordagens equivocadas poderá gerar um efeito contrário ao que se pretende.

Dessa forma, ambientes visuais de programação facilitam o desenvolvimento de software em um contexto que é divertido e não ameaçador [Meerbaum-Salant et al. 2013]. De acordo com de Kereki et al. (2008), alguns ambientes de programação como Alice, Jeroo e GreenFoot foram testados e avaliados e se mostraram não intuitivos, sendo os resultados não significativos.

Esses fatores justificam a utilização do Scratch [MIT 2014] como ferramenta a ser utilizada no curso proposto, pois além de encorajar jovens a manter atitudes positivas sobre Ciência da Computação [Meerbaum-Salant et al. 2013], esse ambiente traz uma linguagem que contribui para a aprendizagem de programação através de um conceito inovador de desenvolvimento de código orientado a projeto, que privilegia a Computação Criativa, expressão que é utilizada para reconhecer que o conhecimento e as práticas que os jovens precisam adquirir para criar software devem ser provenientes dos seus interesses pessoais [Scaico 2013].

De acordo com Resnik et al. (2009), quando o Scratch foi criado, eles desejavam desenvolver uma abordagem para programação cujo apelo seria para pessoas que nunca haviam se imaginado programadoras antes, o que claramente representa os participantes de nosso curso: crianças.

3. Experiências Correlatas Utilizando o Scratch

O uso do Scratch como ferramenta para o ensino de programação tem sido proposto por diversos autores. Dias et al. (2014) e Kereki et al. (2008) utilizaram o Scratch no Ensino Superior como forma de introduzir os conceitos de programação aos universitários. No trabalho de Dias et al (2014), o curso foi dividido em aulas, nas quais eram realizados pequenos desafios descritos como algoritmos na forma de linguagem natural, de modo que os alunos teriam de utilizar os conceitos aprendidos previamente para resolver a temática utilizando Scratch. Em Kereki et al. (2008), foram desenvolvidos exercícios divididos por assunto em vez de serem reunidos em um único projeto com um objetivo final.

Para o Ensino Médio, Meerbaum-Salant et al. (2013) propuseram atividades para ensino de conceitos de programação com o uso exclusivo do Scratch. Scaico (2013) realizou uma aula teórica seguida de duas aulas práticas, nas quais foram desenvolvidos projetos que abordassem os conteúdos aprendidos. Em Rodriguez et al. (2015), houve atividades relacionadas à exploração dos recursos do Scratch e discussões acerca dos conceitos de lógica de programação por trás dos comandos criados, sendo, no final do curso, construído um jogo educativo por grupos de alunos.

As propostas desenvolvidas para o Ensino Fundamental focam no ensino de conceitos introdutórios de lógica e computação. Em Oliveira et al. (2014), foram utilizadas brincadeiras para introduzir alguns conceitos (número binários, algoritmos e introdução às linguagens de programação) e o Scratch utilizado para o aprendizado dos conteúdos relacionados à programação. A estratégia de ensino descrita em von Wangenheim et al. (2014) teve como base a aprendizagem ativa e os comandos do Scratch foram usados para a criação de histórias interativas. Em Schoeffel et al. (2015), os conceitos de lógica foram introduzidos para a resolução de desafios matemáticos e o Scratch foi utilizado nas aulas práticas de programação. No trabalho de Wilson et al. (2010) foram realizadas atividades práticas em que o uso da programação era utilizado para resolver um determinado problema. Gülbahar et al. (2014) também abordaram conceitos introdutórios de computação durante a construção de programas no Scratch.

Em nosso curso, foi proposta a construção de um jogo como elemento principal da construção de conhecimento. A cada aula, os alunos aprenderam conceitos de Pensamento Computacional de maneira expositiva de modo a instigá-los a pensar nos conceitos por trás da utilização do Scratch. Com essa introdução, os alunos realizaram atividades práticas. A princípio utilizaram-se atividades com personagens e cenários que não eram relacionados ao jogo final com intuito de praticar o conteúdo e, no fim de cada aula, foi realizada a construção de uma subparte do projeto final, demonstrando como o elemento de programação aprendido era importante para a construção do jogo.

A utilização do Scratch em cursos superiores, conforme elencado acima, visava tornar o ensino de programação acessível e menos traumático aos alunos de cursos de Computação. No Ensino Médio, a utilização do Scratch se deu em conjunto com algumas aulas teóricas e/ou discussões acerca de conteúdos de programação. No entanto, no Ensino Fundamental, as abordagens aqui apresentadas foram práticas, não sendo realizada uma explanação teórica sobre o conteúdo por trás da construção de animações (essa construção de conhecimento foi realizada de maneira indireta). Observa-se também que a construção do projeto final nos trabalhos que fizeram uso dessa forma de avaliação não tiveram sua construção em conjunto com os tópicos de lógica apresentados. Em nosso curso, optamos por fornecer aos alunos o entendimento teórico por trás da utilização do Scratch e enfatizar sua importância como parte do jogo sendo construído, mostrando aos alunos a necessidade daquele processo algorítmico ser aprendido e utilizado.

4. Estudo de Caso Realizado

A abordagem aqui proposta consta de um minicurso introdutório de Pensamento Computacional que tem como objetivo estimular e ensinar aos jovens do Ensino Fundamental os conceitos de computação, por meio do uso do Scratch e tendo como agente motivador a criação de um jogo.

Visando a verificação da abordagem proposta, foi realizado um estudo de caso em alunos do 6º ano do Ensino Fundamental de uma Escola Particular. O conteúdo programático foi desenvolvido para abordar, de maneira acessível, os principais conceitos de Pensamento Computacional. A construção de um jogo foi utilizada como abordagem principal. Dessa forma, o conteúdo programático foi elaborado de modo que, a cada aula, os alunos fossem capazes de desenvolver uma subtarefa necessária para construção do jogo e, ao mesmo tempo, aprender um conceito relacionado ao pensamento computacional.

Para desenvolvimento do curso, foi feita uma parceria com o colégio Rezende e Rezende, localizado em Jacareí - SP. Sua aplicação foi em Fevereiro de 2017 no início do período letivo dos alunos. Foram desenvolvidas 8 aulas com duração aproximada de 1 hora e 30 minutos ministradas por um aluno do Instituto de Ciência e Tecnologia da Universidade Federal de São Paulo sob supervisão de um docente e auxílio de 8 monitores (também alunos do Instituto). A preparação do material didático foi realizada pelos alunos participantes do projeto.

A escola conta com um laboratório com capacidade para 20 alunos com estrutura apropriada para execução do projeto. No total, foram 20 alunos com idades de 10 e 11 anos. Cada aluno teve um computador para utilização, o que possibilitou que a utilização da ferramenta fosse individualizada e pudesse ser aproveitada por todos os alunos. O conteúdo foi apresentado na forma de slides, mostrando os tópicos relacionados à programação em conjunto com a ferramenta Scratch versão 2.0, por não necessitar de conexão com Internet.

4.1. Conteúdo Programático

Como parte teórica do curso, foram abordados os conteúdos descritos na Tabela 1. Cada conteúdo introduziu conceitos de programação em conjunto com habilidades para desenvolvimento de aplicações utilizando o Scratch.

Tabela 1. Conteúdo programático do curso.

Conteúdo	Descrição
Introdução à programação e utilização do Scratch	Utilização da programação para resolução de tarefas cotidianas para os alunos, o ambiente de programação Scratch e diferentes tipos de comandos para criação de cenários
Conceitos de algoritmos e criação de figuras e efeitos visuais e sonoros com Scratch	Conceitos de algoritmos, descrição narrativas de algoritmos, efeitos sonoros e visuais disponíveis no Scratch, utilização de camadas e criação de cenas com animações
Procedimentos	Implementação e criação de procedimentos em Scratch e técnicas para construção de comandos
Variáveis	Tipos de dados no Scratch, criação e manipulação
Estruturas de decisão	Técnicas para resolução de problemas, uso dos comandos de seleção e fluxo de controle
Estruturas de repetição	Estruturas de repetição para execução de comandos eficientes e variáveis de controle
Processamento de strings	Técnicas para processamento de strings
Projeto Final	Criação do jogo e reunião dos conceitos aprendidos

4.2. Criação do Jogo

O jogo escolhido para criação foi o “Flappy Bird”, muito conhecido e difundido entre jovens de diferentes idades. É um jogo eletrônico para dispositivos móveis criado em 2013 pela dotGears (Figura 1). Escolheu-se esse jogo pela facilidade de criação com os conteúdos propostos e pela relação com a realidade dos participantes. O jogo se baseia na movimentação de um pássaro em um cenário com diversos obstáculos, tendo como objetivo o acúmulo da maior pontuação possível, controlando o pássaro para que não colida nos elementos do cenário.



Figura 1. Jogo proposto para criação.

Para sua criação, foram seguidos alguns passos que vão ao encontro dos conteúdos aprendidos no decorrer do curso. Cada parte do jogo foi desenvolvida na aula respectiva em conjunto com outras atividades (por exemplo, utilização de outros personagens e cenários) para aquisição de habilidades lógicas utilizando o Scratch. Cada etapa na criação do jogo se relacionou com um elemento de programação apresentado e previamente praticado. Dessa forma, foi utilizado o seguinte roteiro, com seus respectivos elementos de programação:

- (a) Criação do personagem e do efeito sonoro sob as diferentes condições do jogo: criação de figuras e efeitos visuais levando-se em conta a construção lógica por trás do conceito de algoritmos;
- (b) Ações dos personagens e dos obstáculos sob as diferentes condições do cenário: cada ação foi dividida em um procedimento diferente, demonstrando a importância na utilização de procedimentos para organização da estrutura geral;
- (c) Manipulação das coordenadas geográficas de acordo com as teclas pressionadas;
- (d) Pontuação obtida pelo jogo: utilização de variáveis e contadores para controle dos pontos obtidos pelo jogador;
- (e) Movimentação do cenário e do personagem: estruturas de repetição necessárias para garantir o funcionamento correto do jogo;
- (f) Ações do personagem ao bater nos obstáculos ou cair: quebra de fluxo sob determinadas estruturas de decisão;
- (g) Mensagens de início e final de jogo: processamento de strings de acordo com uma situação específica.

5. Discussão e Resultados

Nesse contexto, foram desenvolvidas habilidades de modo a engajar os alunos a utilizar o pensamento computacional para resolver tarefas condizentes com sua realidade, sendo capazes de utilizar o Scratch como ferramenta para aplicações diversas. Com a aprendizagem dos comandos principais, os alunos puderam desenvolver um jogo, no qual se sentiram confortáveis e puderam ultrapassar a expectativa, criando diferentes versões do “Flappy Bird” com ações diferentes para outras teclas e com cenários que

demonstraram o domínio em utilizar o Scratch. Como objetivo principal, os alunos puderam visualizar os principais conceitos envolvidos na criação de um programa, desenvolvendo o raciocínio sequencial e a habilidade para resolver um dado problema utilizando técnicas inerentes à programação.

Em estudo realizado por Meerbaum-Salant et al. (2013), os resultados mostraram que a maioria dos alunos é capaz de compreender os conceitos de Ciência da Computação. Em nosso caso, os resultados se mostraram bastante semelhantes. Os alunos se sentiram confortáveis com a temática abordada nos slides e as dúvidas que foram levantadas durante a explicação teórica puderam ser resolvidas de maneira eficiente pelos monitores, mostrando o tópico de programação de maneira visual no Scratch.

Eventuais erros cometidos pelos alunos na construção de ações no Scratch foram facilmente contornados. Muitas das vezes sem intervenção de um monitor. Comparando esse fato com o que apontam Malan et al. (2007), compreendemos que o fato de não existir distração com a sintaxe permitiu que não houvesse desmotivação por parte dos alunos, haja vista que não houve evasão em nosso curso. Tal fato se mostra bastante positivo, principalmente quando comparado à iniciativa realizada por Oliveira et al. (2014), que utilizaram a mesma ferramenta para alunos de ensino fundamental e observaram um índice de evasão de 31,25%.

Pudemos observar que a aquisição do conhecimento proposto pelo conteúdo programático foi atingida, pois todos os alunos foram capazes de construir o jogo; ratificando o uso do Scratch para iniciação à programação. Embora os resultados obtidos por Dias et al. (2014) e Kerekí et al. (2008) tenham sido relacionados a alunos de graduação, pudemos observar o mesmo tipo de reação dos participantes: o primeiro contato com programação não foi traumático e notou-se um alto nível de motivação em conjunto com uma percepção positiva de programação.

Em relação à utilização do projeto final (criação do jogo) como elemento chave na construção do conhecimento, consideramos como positiva. Conforme acima mencionado, todos os alunos conseguiram terminar o jogo e depois puderam jogar os jogos criados pelos amigos; fato que causou diversão e satisfação em ter criado uma ferramenta que pode ser utilizada para diversão de outras pessoas. Em pesquisa realizada por Ouahbi et al. (2015), foi testada a seguinte hipótese: "A criação de jogos por estudantes usando um ambiente de programação apropriado pode melhorar a motivação de estudantes por programação?". Os resultados obtidos pelos autores foi positivo e da mesma forma obtivemos as mesmas conclusões.

Como os participantes de nosso curso possuíam idades de 10 e 11 anos, tivemos cuidado ao apresentar os conceitos relacionados às coordenadas dos personagens na tela (plano cartesiano acompanhado de números negativos), pois esses conceitos não fazem parte das competências de Matemática apresentadas para o ano escolar em que estão inseridos. Os participantes tiveram a princípio dificuldade em visualizar essa localização. No entanto, quando puderam ver as animações no Scratch e como a movimentação do personagem mudava o eixo de localização, o entendimento foi natural e os alunos puderam usar esse conceito para construção de animações. Esse fato vai ao encontro do que observou Scaico (2013) ao afirmar que o projeto desenvolvido demonstrou que apesar das inúmeras deficiências que os estudantes possuíam e que estavam relacionadas à escrita, leitura e fundamentação lógico-matemática, foi possível ensinar e aprender os conceitos básicos de programação, ao passo que se aprendia também sobre o mundo

através da própria programação.

Como forma de avaliar os objetivos de aprendizagem, utilizamos uma abordagem semelhante à utilizada por von Wangenheim et al. (2014). Com base na avaliação dos trabalhos construídos e na observação dos monitores, identificamos que todos os alunos participantes atingiram os objetivos de aprendizagem relacionados à aquisição de Pensamento Computacional para resolução de problemas. De modo geral, como análise do atendimento de objetivos de aprendizagem principais temos que (1) usar o Scratch como ferramenta para resolução de problemas foi atingido por todos os alunos, (2) usar o Scratch como ferramenta para criação de jogos e animações foi atingido pela maioria dos alunos e (3) criar animações adequadas e condizentes com o objetivo proposto foi atingido por todos os alunos.

Visando verificar se os objetivos propostos pelo roteiro do curso (Seção 4.2) foram atingidos, classificamos os alunos de acordo com a forma em que realizaram as atividades propostas, conforme mostra a Tabela 2.

Tabela 2. Análise do desenvolvimento do roteiro do curso

Roteiro	Realizado com auxílio total do monitor	Realizado com auxílio parcial do monitor	Realizado sem auxílio do monitor
Criação do personagem e do efeito sonoro	0%	0%	100%
Programação das ações do personagem	10%	20%	70%
Manipulação das coordenadas geográficas	20%	20%	60%
Uso de variáveis e contadores	30%	50%	20%
Movimentação do cenário e do personagem	20%	60%	20%
Manipulação de mensagens	0%	0%	100%

Os conteúdos mais simples e que envolviam apenas a parte criativa dos participantes foram realizados rapidamente e não necessitaram ter uma interferência por parte dos monitores. A maior dificuldade apresentada no curso foi a manipulação de variáveis e seu incremento.

O curso foi ministrado em horário extracurricular, de forma que os alunos retornavam à escola para assistir às aulas. Muitos dos alunos se sentiram tão confortáveis e motivados com a utilização da ferramenta e com sua utilização para criação de jogos que chegavam mais cedo ao curso para utilizar o Scratch para criar animações que não faziam parte do conteúdo proposto. Diversos comandos que apresentamos para os alunos já haviam sido aprendidos por eles simplesmente no manuseio da ferramenta. Não tivemos problemas com indisciplina e falta de atenção, pois os alunos realmente se engajaram nas atividades e se interessaram pelo que foi apresentado.

6. Conclusão

Este trabalho apresentou um processo de ensino e aprendizagem em Pensamento Computacional, aplicado a alunos do 6º ano do Ensino Fundamental de uma Escola

Privada, que teve como diferencial a explanação dos conteúdos durante o desenvolvimento de um jogo pelos alunos. Assim, foram desenvolvidos o raciocínio sequencial e a habilidade para resolver um dado problema utilizando técnicas inerentes à programação.

Os resultados mostraram que a metodologia proposta cativou a atenção de crianças e jovens e os conceitos foram assimilados facilmente. Com isso, concluímos que a aprendizagem de Pensamento Computacional e Programação pode ser agradável e de fácil entendimento dos aprendizes.

Referências

- Bundy, A. (2007). Computational thinking is pervasive. *Journal of Scientific and Practical Computing*, 1(2), 67-69.
- Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical transactions of the royal society of London A: mathematical, physical and engineering sciences*, 366(1881), 3717-3725.
- França, R. D., et al. (2012). Ensino de ciência da computação na educação básica: Experiências, desafios e possibilidades. In *XX WEI*.
- Wing, J. (2011). Research notebook: Computational thinking - What and why? *The Link Magazine, Spring*.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.
- Stephenson, C., Gal-Ezer, J., Haberman, B., & Verno, A. (2005). The new educational imperative: Improving high school computer science education. *Final Report of the CSTA Curriculum Improvement Task Force*.
- von Wangenheim, et al. (2014). Ensino de computação com scratch no ensino fundamental—um estudo de caso. *RIBIE* 22.03: 115.
- Barcelos, T. S.; Silveira, I. F. (2012). Pensamento Computacional e Educação Matemática: Relações para o Ensino de Computação na Educação Básica. In *Anais XX Workshop sobre Educação em Computação*. p. 52-55.
- Scaico, Pasqueline Dantas, et al. (2013). Ensino de programação no ensino médio: Uma abordagem orientada ao design com a linguagem scratch. *RIBIE* 21.02: 92.
- Fincher, S.; Cooper, S.; Kölling, M.; Maloney, J. (2010). Comparing alice, greenfoot & scratch. *Proceedings do 41st SIGCSE'10*, Milwaukee, Wiscosin, USA. p. 192-193.
- de Kereki, Inés Friss et al. (2008). Scratch: Applications in computer science 1. *Frontiers in Education Conference*. FIE 2008. 38th Annual. IEEE.
- Meerbaum-Salant, Orni, Michal Armoni, and Mordechai Ben-Ari (2013). Learning computer science concepts with scratch. *Computer Science Education* 23.3: 239-264.
- MIT. SCRATCH (2014). <http://scratch.mit.edu>. March.
- Alves, Lynn e Nery, Jesse (orgs., 2015). *Jogos eletrônicos, mobilidades e educações*. Bahia: EdUFBA.
- Alves, Lynn e Coutinho, Isa de J. (orgs., 2016). *Jogos Digitais e Aprendizagem*. Campinas: Papirus.

- Piaget, Jean (1973). *A Formação do Símbolo na Criança*. Rio de Janeiro: Zahar.
- Prensky, Marc (2012). *Aprendizagem baseada em jogos digitais*. São Paulo: Editora SENAC.
- Soares, Eliana e Petarnella, Leandro (orgs., 2012). *Cotidiano Escolar e Tecnologias*. Campinas: Ed. Alínea.
- Vygotsky, L. S. (1984). *A formação Social da mente*. São Paulo: Martins Fontes.
- Wallon, Henri (1979). *Psicologia e Educação da Criança*. Lisboa: Veja/Universidade.
- França, R. S., & Tedesco, P. (2015). Explorando o Pensamento Computacional no Ensino Médio: do design à avaliação de jogos digitais. In *Anais do XXI WIE*.
- Resnick, M. et al. (2009). Scratch: programming for all. *Communications of the ACM* 52.11: 60-67.
- Malan, D. and Leitner, H. (2007). Scratch for budding computer scientists, *Proc. of SIGCSE*, USA.
- Dias, Klissiomara L. et al. (2014). A linguagem Scratch no ensino de programação: Um relato de experiência com alunos iniciantes do curso de licenciatura em computação. *XXII Workshop de Educação em Computação*. Brasília. 2014.
- Oliveira, Millena L. S, et al. (2014). Ensino de lógica de programação no ensino fundamental utilizando o Scratch: um relato de experiência - *XXII WEI*, Brasília.
- Ouahbi, Ibrahim, et al. (2015). Learning basic programming concepts by creating games with scratch programming environment. *Procedia-Social and Behavioral Sciences* 191: 1479-1482.
- Wilson, Amanda et al. (2010). Evaluating Scratch to introduce younger schoolchildren to programming. *Proceedings of the 22nd Annual Psychology of Programming Interest Group*. Universidad Carlos III de Madrid, Leganés, Spain.
- Rodriguez, Carla, et al. (2015). Pensamento Computacional: transformando ideias em jogos digitais usando o Scratch. *Anais do WIE*. Vol. 21. No. 1.
- Gülbahar, Yasemin et al. (2014). The effects of teaching programming via Scratch on problem solving skills: A discussion from learners' perspective. *Informatics in Education-An International Journal*. Vol 13_1: 33-50.
- Schoeffel, Pablo, et al. (2015). Uma Experiência no Ensino de Pensamento Computacional para Alunos do Ensino Fundamental. *Congresso Brasileiro de Informática na Educação*. Vol. 4. No. 1.