

Significação da Aprendizagem Através do Pensamento Computacional no Ensino Médio: uma Experiência com *Scratch*

Fellipe Oliveira Ramos¹, Lilian da Silva Teixeira²

¹Licenciado em Ciências da Computação – Instituto Federal Baiano, *campus* Senhor do Bonfim

²Doutoranda em Educação e Contemporaneidade – Universidade do Estado da Bahia. Mestre em Educação – Universidade Federal de Sergipe. Docente do Instituto Federal Baiano, *campus* Senhor do Bonfim- BA.

ramosfellipe@hotmail.com, lilian.teixeira@bonfim.ifbaiano.edu.br

Abstract. *This article is the result of the analysis of a short course entitled "Scratch and Games Development" offered to students in high school classes of basic education. The main objective was to analyze the performance of practices that exploit the Computational Thinking with Scratch. The methodology that guided the work was based on a qualitative approach, exploratory, and as the main instrument used to carry out short course with high school students. We conclude that these students are able to interact and produce digital objects in a successful programming language environment presenting specificities peculiar to their students condition of basic education.*

Resumo. Este artigo é resultado da análise de um minicurso intitulado “Scratch e Desenvolvimento de Jogos” oferecido a alunos de classes do Ensino Médio da Educação Básica. O objetivo principal foi analisar o desempenho de práticas que exploram o Pensamento Computacional com o *Scratch*. A metodologia que orientou o trabalho foi embasada na abordagem qualitativa, de caráter exploratório, e como principal instrumento utilizou-se a realização de minicurso com os alunos do Ensino Médio. Conclui-se que os referidos alunos são capazes de interagir e produzir objetos digitais num ambiente de linguagem de programação com êxito apresentando especificidades peculiares à sua condição de estudantes da educação básica.

Introdução

Nos dias atuais, inúmeros estudos têm se desenvolvido a fim de demonstrar as possibilidades e a necessidade de inserção de conceitos de Computação na educação básica. Estas iniciativas relatam experiências bem sucedidas do ensino de lógica de programação, estruturação de algoritmos e abstração de conceitos computacionais a alunos dos Ensinos Fundamental e Médio. Por meio da exploração de ambientes de linguagem de programação e da exploração de jogos digitais, os alunos são estimulados à exploração do raciocínio lógico necessário para a construção de algoritmos [Marques *et al.*, 2011; Rapkiewickz, 2006].

Estudos como os de Oliveira (2009) e Souza; Lencastre (2013) comprovam a eficiência da promoção do ensino do Pensamento Computacional por meio do *Scratch*.

A inserção da Computação na educação Básica, enquanto ciência pode se dar por meio da perspectiva do desenvolvimento de atividades que exploram o Pensamento Computacional, de acordo com as Diretrizes Curriculares Nacionais para os cursos de graduação em Computação, no Parecer nº 136/2012, “a introdução do Pensamento Computacional e algorítmico na educação básica fornece os recursos cognitivos necessários para a resolução de problemas, transversal a todas as áreas do conhecimento” [Brasil, 2012].

Diante do exposto, o presente trabalho apresenta como problemática principal a seguinte questão: Como se dá o desempenho de estudantes do Ensino Médio no desenvolvimento de atividades com a linguagem de programação *Scratch*? Para tanto, objetivou-se analisar o desempenho de práticas que exploram o Pensamento Computacional sob a perspectiva da Aprendizagem Significativa por meio de um minicurso sobre a linguagem de programação *Scratch* desenvolvido com alunos de turmas do 1º ao 3º ano do Ensino Médio do Colégio Estadual Luís Eduardo Magalhães, na cidade de Senhor do Bonfim/BA.

1. Pensamento Computacional

O termo Pensamento Computacional surge pela primeira vez com Wing (2006), ao afirmar que ele se constrói nos poderes e nos limites dos processos computacionais, sejam eles executados por um ser humano ou por uma máquina. Para a autora, os métodos computacionais proporcionam a resolução de problemas e o desenho de sistemas que não seriam resolvidos pelo ser humano. Portanto, ele envolve a formulação de problemas e suas respectivas soluções, que são representadas de forma que possam ser realizadas por agentes de processamento de informação. Sob essa perspectiva as ferramentas computacionais são utilizadas a fim de transformar determinado problema aparentemente difícil em um processo que, auxiliado pelas ferramentas computacionais, possa ser resolvido mais facilmente.

A *The Royal Society* (2012) concebe Pensamento Computacional como “o processo de reconhecimento de aspectos da computação no mundo que nos rodeia, e de aplicar ferramentas e técnicas da Ciência da Computação a fim de entender e analisar sistemas e processos naturais e artificiais”. Ou seja, pensar através de elementos computacionais proporciona ao ser humano a compreensão do ambiente em que este está inserido, além de dar meios para que ele interfira neste ambiente.

Na educação básica, especificamente no Ensino Médio, atividades na área de Pensamento Computacional promovem uma formação mais ampla do sujeito para enfrentar os desafios da contemporaneidade. Além de proporcionar ao educando aprendizagens na área de leitura, aritmética, assim como um acentuado desenvolvimento do pensamento analítico. [Carvalho *et al.* 2013]

2. *Scratch*: uma ferramenta criativa e acessível

O *Scratch* tem sido uma das ferramentas mais utilizadas na promoção do Pensamento Computacional em ambientes escolares. É uma linguagem de programação *open source* (termo destinado a softwares de código aberto que disponibilizam seu código-fonte publicamente para fins de estudo, mudança e melhoria de seu design.), colaborativa idealizada e desenvolvida por uma equipe de investigação do *Media Laboratory* do *Massachusetts of Institute of Technology* (MIT). Por ser um ambiente de programação

visual permite a criação de projetos ricos por meio de mídias interativas, sendo possível a criação de games, histórias animadas, tutoriais, simulações, e outros, por meio de seus blocos programáveis.

Divulgado em maio de 2007, o *Scratch* começou a ser desenvolvido em 2003. Em sua primeira versão era indicado para jovens de 8 a 16 anos, todavia segundo informações do site do MIT é “utilizado por pessoas de todas as idades, inclusive crianças mais jovens”, que “aprendem a pensar de forma criativa, trabalhar de forma colaborativa e pensar de forma sistemática”. Segundo Maloney *et al.* (2010), este projeto inicial de *Scratch* foi motivado por necessidades e interesses de jovens de centros pós-escolares de computação como o *Intel Computer Clubhouse* que, por meio de seus ambientes criativos estimula jovens ao desenvolvimento de novas habilidades e ideias tecnológicas.

Um dos principais objetivos do *Scratch* é introduzir noções de linguagem de programação àquelas pessoas que não possuem experiência com ambientes de linguagens de programação. Ao exigir de seu usuário decisões óbvias através da utilização de seus blocos visuais, ele apresenta *layout* simples, com janela única, e utilização mínima de comandos, proporciona-se aos jovens a manipulação de mídias através da exploração e partilha de suas produções, sem a necessidade de compreensão de uma sintaxe complexa como possuem outras linguagens de programação. [Maloney *et al.* 2010, p. 5].

Os projetos em *Scratch* são formados por um palco e vários objetos. O palco é como um plano de fundo onde serão executadas as ações dos objetos, que são associados a sons, imagens, variáveis. Os comandos que dão vida aos objetos precisam somente ser arrastados e soltos no local específico, onde são encaixados uns nos outros. Ao serem combinados sempre formam programas corretos, inexistindo erros sintáticos. [Malan e Leitner 2007].

Apesar do *Scratch* ser uma produção relativamente recente, pode-se afirmar que sua difusão em todo o mundo foi muito rápida. Graças à comunidade de colaboradores, atualmente está disponível em inúmeros idiomas. Atualmente o *Scratch* encontra-se na versão 2.0, e pode ser utilizado de forma *online*, por meio do *website* <https://scratch.mit.edu>, bem como, pode ser feito o *download* gratuito do editor *offline*, disponível para os sistemas operacionais *Mac OS*, *Windows* e *Linux*.

3. Aprendizagem Significativa um conceito relacionado à prática com *Scratch*

Concebida por David Ausubel na década de sessenta, a teoria da Aprendizagem Significativa (AS) é representada pela interação cognitiva existente entre um conhecimento prévio e um novo conhecimento. Dessa forma, ela ocorre em “qualquer atividade na qual a representação cognitiva de experiência prévia e os componentes de uma situação problemática apresentada são reorganizados a fim de atingir um determinado objetivo” [Ausubel 1968].

Moreira (1999) considera que a Aprendizagem Significativa opera a partir do momento em que o sujeito faz uso de conceitos aprendidos anteriormente como base para a aprendizagem de outros conceitos.

Uma prática pedagógica que se inspira nos princípios da teoria da Aprendizagem Significativa proporciona que os novos conceitos demorem de ser esquecidos, e caso esquecidos sejam mais facilmente lembrados. Desse modo, ao trabalhar conteúdos que requerem do sujeito novas competências e habilidades, no contexto da Aprendizagem Significativa, o indivíduo transferirá rapidamente estes conceitos para a resolução de problemas outros a serem apresentados, mesmo que estes estejam configurados em contextos diferentes daqueles aprendidos.

A realização da oficina sobre *Scratch* com os alunos do Ensino Médio, evidenciou o quanto os conceitos computacionais, e ainda, conceitos da matemática podem ser facilmente aprendidos quando essa aprendizagem ocorre de forma significativa, pois a interação com a linguagem de programação do *Scratch* provoca no sujeito a organização do seu pensamento amparado em noções subsunçoras num processamento de significação do conhecimento a ser construído.

4. Materiais e Métodos

O presente trabalho foi desenvolvido sob a metodologia de abordagem qualitativa. “A pesquisa nessa área lida com seres humanos que, por razões culturais de classe, de faixa etária, ou por qualquer outro motivo, têm um substrato comum com o investigador.” (Minayo *et al.* 2011, p. 21)

Desse modo, foi utilizado-se a metodologia de pesquisa de campo de caráter exploratório, que apresenta um aspecto de pesquisa empírica, em que aplicou-se um procedimento sistemático (execução de minicurso) para a obtenção de dados de análise. No caso em questão, optou-se por utilizar como instrumento interventivo a realização de um minicurso com os sujeitos-alvos deste trabalho.

A experiência do ensino do trabalho com conceitos básicos da área do Pensamento Computacional por meio do *Scratch* se deu através do minicurso intitulado “*Scratch* e Desenvolvimento de Jogos”, oferecido a 12 alunos do Colégio Modelo Luis Eduardo Magalhães, na cidade de Senhor do Bonfim/BA, com duração de 16 horas. O minicurso foi dividido em duas etapas: Definições, Conceitos Iniciais de Computação, Algoritmos; Raciocínio Lógico; e, Programação com o *Scratch*.

4.1 Descrição

Etapa 1: Definições, Conceitos Iniciais de Computação, Algoritmos e Raciocínio Lógico. Nesta etapa abordou-se fundamentos introdutórios da computação, como evolução do computador, números binários, *bits* e *bytes*, *hardware*, *software*, *peopleware* e *firmware* e processamento de dados. Um dos objetivos desta etapa foi apresentar aos alunos, mesmo que de forma introdutória, fundamentos básicos da computação, por meio de conceitos discutidos por aqueles que fazem cursos superiores na área de computação. Estes conceitos foram apresentados aos alunos de forma teórica, mas relacionados a situações práticas, e contextualizadas. Posteriormente eles foram instigados a conceituar computação e informática e a identificar o uso de elementos computacionais na sociedade, seus benefícios e malefícios.

Também foram apresentados aos alunos, conteúdos diretamente ligados à programação, abordou-se conceitos e exemplos de algoritmos, bem como exemplos e utilização de linguagens de programação. Foi explicado que os algoritmos representam

um passo a passo necessário para a execução de determinadas tarefas, são primordiais na área de programação, inclusive na a construção de programas no *Scratch* e estão presentes no conceito de Pensamento Computacional.

Após conhecer estes conceitos os alunos foram desafiados a utilizar o raciocínio lógico a fim de desenvolver algoritmos: “ir para a escola”, “trocar pneu” e “trocar uma lâmpada”.

Etapa 2: Programação com o *Scratch*. Por meio de desafios lançados aos alunos, que envolviam o desenvolvimento de jogos e programas no *Scratch*, buscou-se seguir os conceitos de sequências, *loops*, eventos, paralelismo, condicionais, operadores e dados, propostos por Brennan e Resnick (2012) no desenvolvimento de atividades com o Pensamento Computacional. Dessa forma, esta etapa foi dividida em dois momentos, que abordaram conceitos teóricos de programação seguidos de desafios no *Scratch* a serem vencidos por meio da utilização desses conceitos.

No primeiro momento abordou-se os conceitos de Estruturas de Controle de Seleção (simples e composta) e Estruturas de Controle de Repetição (pré-teste, pós-teste e variável de controle), relacionando-os com os blocos existentes no *Scratch*. No Desafio 1 foi proposto o desenvolvimento de uma animação composta por dois personagens, e um palco da escolha do aluno, sendo que os personagens deveriam trocar suas fantasias por meio da utilização das Estruturas de Seleção e/ou Repetição. Nesse momento inicial os alunos foram motivados a descobrir as funcionalidades do *Scratch* de forma autônoma.

Na sequência foram discutidos os conceitos de programação: paralelismo, operadores e armazenagem, recuperação e atualização de dados por meio de variáveis e listas, mas também foram desenvolvidas práticas no ambiente do *Scratch* juntamente com uma explanação sobre a importância da compreensão e da aplicação de coordenadas no *Scratch* por meio da utilização de blocos específicos. Para aplicar estes conceitos foi proposto o Desafio 2, no qual os alunos desenvolveram um jogo simples de labirinto. Para tanto, teria que ser desenhar um caminho por meio da ferramenta “plano de fundo”; inserir 7 atores (um teria movimentos e percorreria o caminho, afim de caçar os outros seis atores que estariam espalhados no palco); implementar paralelismo; utilizar blocos com operadores; e implementar um placar por meio da criação de uma variável que contabilizaria os atores tocados no caminho. A importância das coordenadas foi observada por meio do estabelecimento de uma regra que garantia ao ator principal do jogo retornar ao seu ponto inicial quando tentasse sair do caminho estabelecido.

5. Resultados e Análise

Todo o trabalho foi organizado em quatro momentos. Dois momentos referentes a primeira etapa, e dois momentos referentes a 2ª etapa.

Na primeira etapa do minicurso as atividades foram teóricas, e culminaram com a produção de algoritmos. A princípio os alunos demonstraram resistência, principalmente em relação aos conceitos computacionais no plano teórico, porém a medida que estes conceitos eram relacionados às atividades do cotidiano os alunos envolviam-se mais demonstrando estarem surpresos ao compreenderem a gênese de construção de softwares a partir da lógica das sequências que compõe um algoritmo.

Todos os alunos envolvidos na atividade produziram algoritmos relacionados a atividades do cotidiano. É interessante observar que cada sujeito produziu sua própria lógica sequencial, alguns mais elaborados que outros, mas todos alcançaram o resultado esperado.

A problematização foi um fator importante para fazer os sujeitos refletirem sobre o que estavam produzindo, a medida que organizavam suas sequências, questionava-se se estes não estavam esquecendo de inserir na sequência “passos” importantes para se alcançar o fim pretendido.

A figura 01 a seguir apresenta o algoritmo “trocar pneu” desenvolvido por um dos alunos que participaram do minicurso.

Trocar pneu:

- 1- Pegar o macaco
- 2- Ir debaixo do carro e levantá-lo
- 3- Pegar a chave-de-roda
- 4- Desmontar os parafusos-de-roda
- 5- Trocar o pneu
- 6- Encaixar o outro pneu
- 7- Enroscar os parafusos
- 8- Colocar o macaco e tirá-lo debaixo do carro

Figura 01. Produção Algoritmo “Trocar Pneu”- Aluno A

Observa-se que o aluno A, trouxe todos os passos necessários para executar a ação de forma correta, porém no passo 2, e no 8, ele acaba integrando duas ações distintas que poderiam estar separadas.

Ao encerrar as atividades da 1ª etapa, iniciou-se a fase de aproximação com os conteúdos de programação necessários para a prática com a linguagem *Scratch*. Ao apresentar os conceitos computacionais voltados à programação, os sujeitos voltavam a mostrar as mesmas resistências do momento teórico da primeira etapa.

Os alunos no ambiente do laboratório de informática apresentam um comportamento de grande expectativa para a interação direta com a máquina e seus recursos digitais.

A seguir apresentam-se as figuras 4 e 5, que são produções dos alunos após o momento de apresentação do *Scratch* e execução do Desafio 1. Os alunos foram desafiados a produzir uma animação, conforme a descrição aqui apresentada na subseção 4.1, sem a intervenção do orientador do minicurso, os alunos foram estimulados à exploração do ambiente da linguagem *Scratch*.

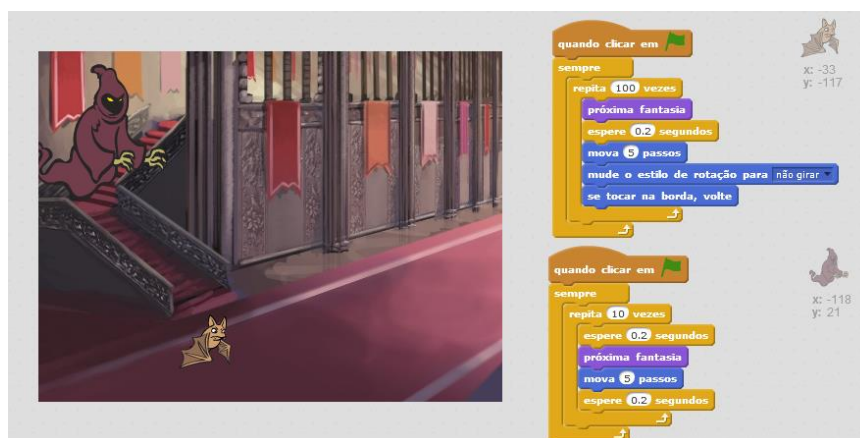


Figura 02. Palco e Blocos – Produção do Aluno B

Ao observar-se a estrutura construída pelo Aluno B percebe-se que ele cumpriu a atividade proposta, pois modificou o plano de fundo, inseriu dois atores e implementou a troca de suas fantasias por meio da utilização das Estruturas de Repetições “sempre” e “repita”.

Na animação abaixo, o aluno C, assim como o aluno B, concluiu o desafio proposto. O aluno C descobriu de forma autônoma a aplicação dos operadores na Estrutura de Seleção “se”, o que proporcionou o controle dos blocos a serem executados, e também descobriu que para utilizar o bloco “próxima fantasia” era necessário utilizar o bloco “mova x passos”. Além disso, ele utilizou blocos que não foram solicitados como “se tocar na borda, volte”, o que promoveu uma melhor execução da animação. Observamos aqui, portanto, a criatividade e a liberdade de criação proporcionadas pelo *Scratch*, definidas por Pinto (2010) como algumas das potencialidades desta linguagem de programação.

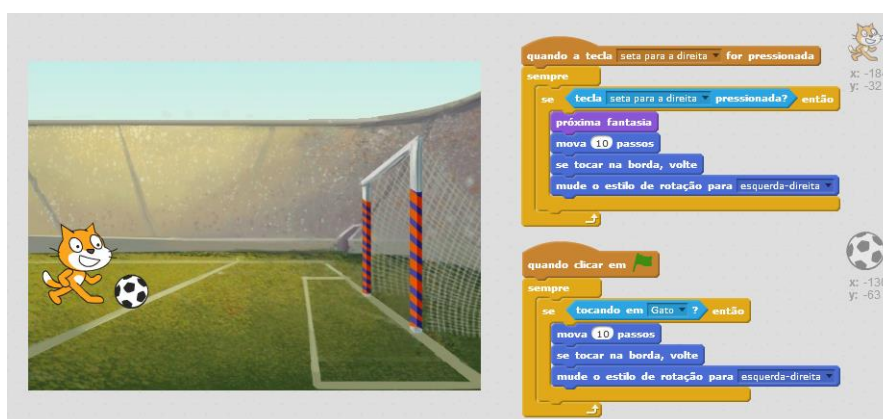


Figura 03. Palco e Blocos – Produção do aluno C

O que mais chamou atenção foi a postura criativa e autônoma dos alunos, quando se deparavam com os conflitos na organização da estrutura lógica dos blocos, e rapidamente encontravam as soluções mais adequadas e prosseguiram com seus projetos de animação.

Na sequência, após a apresentação teórica de conteúdos relativos a programação, partiu-se para a execução do Desafio 2, também descrito aqui na subseção 4.1. As figuras 04 e 05 a seguir apresentam os produtos obtidos através desta atividade.

Como pode ser visto na figura a seguir, o aluno D cumpriu o Desafio 2 ao inserir todos os elementos solicitados. O conceito de paralelismo é aplicado nos atores que possuem evento comum para serem executados, no caso, “quando clicar em bandeirinha”. Esses atores ficam estáticos na tela, e somem da tela ao serem tocados pelo ator “girl”. Para tanto o aluno D descobriu de forma autônoma o bloco “esconda” e utilizou a mesma sequência de blocos para os atores estáticos, que pode ser vista no ator “pomba”, disposto na Figura 5. Além disso, um fato relevante foi que o aluno D agregou os conceitos aprendidos por meio do Desafio 1 ao implementar blocos de Estruturas Controle, utilizando-se de operadores.

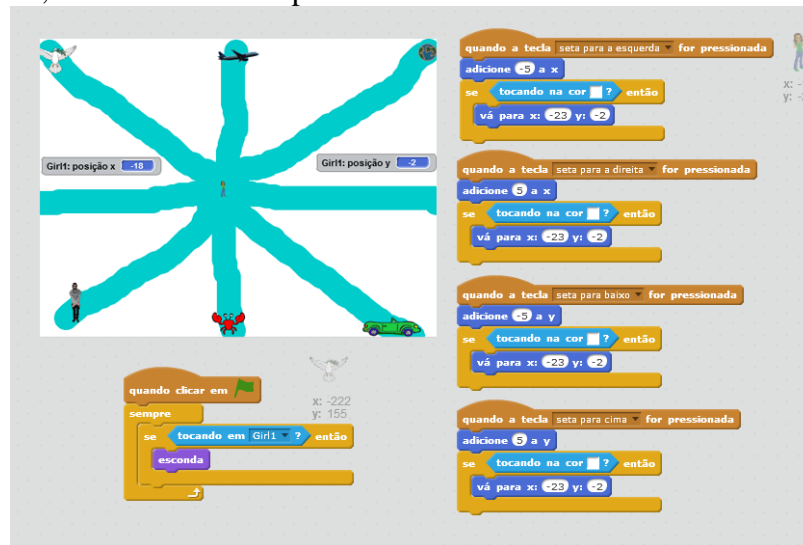


Figura 04. Palco e Blocos – Produção do aluno D

Ao desenvolver o Desafio 2, o aluno E, assim como o aluno D, utilizaram-se do conceito de paralelismo também fizeram uso de mecanismos aprendidos anteriormente ao utilizar os blocos “próxima fantasia” e Estruturas Condicionais e Estruturas de Repetição. O que difere a produção do primeiro, para a do segundo foi a forma com que ele utilizou a Estrutura de Controle “se”, e agrupou todos os blocos de comandos em eventos distintos.



Figura 05. Palco e Blocos – Produção do aluno E

O Aluno E, diferentemente do Aluno D, conseguiu implementar o bloco “mudar estilo de rotação”, e também uma variável capaz de armazenar o número de atores, a fim de exibi-la na tela e informar o número de atores tocados pelo ator que se movimenta. Dessa forma o aluno E concluiu o desafio proposto em sua totalidade. O aluno D, apesar de ter utilizado de uma estrutura lógica diferente também alcançou êxito em sua produção.

CONCLUSÃO

Conclui-se que a realização desse minicurso proporcionou uma rica experiência interventiva aos alunos que foram contemplados, as atividades desenvolvidas foram voltadas a exploração do Pensamento Computacional no ensino médio, a contextualização constante dos conceitos e conteúdos apresentados contribuíram para a aprendizagem significativa.

Através da análise foi possível perceber que os alunos de turmas do Ensino Médio de uma escola pública são capazes de interagir e produzir objetos digitais num ambiente de linguagem de programação, mesmo sem conhecimentos prévios específicos da área da computação, e ainda num espaço reduzido de tempo, visto que todas as atividades desenvolvidas ocorreram em 4 (quatro) dias de trabalho que totalizaram 16 horas de minicurso.

Observou-se ainda, que o *Scratch* é potencialmente significativo, e também que os alunos que participaram do minicurso o frequentaram de iniciativa própria, de forma extraclasses, e estavam predispostos a conhecer os novos conteúdos apresentados. Dessa forma, acredita-se que a realização deste trabalho proporcionou aos alunos a aprendizagem do Pensamento Computacional por meio do *Scratch*, de forma significativa.

Ratificando o que afirma Phillips (2009), constatou-se que os indivíduos que desenvolvem o Pensamento Computacional adquirem aptidão para o desenvolvimento de aplicações, e também competências como o pensamento abstrato, o pensamento algorítmico, o pensamento lógico e o pensamento dimensionável. Consequentemente, se continuarem a explorar o *Scratch*, poderão tornar-se sujeitos com competências e habilidades nem sempre alcançadas por sujeitos que nunca tiveram contato com ambientes de linguagem de programação.

REFERÊNCIAS

- Ausubel, D. P. (1968). *Educational psychology: a cognitive view*. New York and Toronto: Holt, Rinehart and Winston.
- Brasil, Diretrizes Curriculares Nacionais para os cursos de Graduação em Computação. Parecer nº 136/2012. Conselho Nacional de Educação. Disponível em < <http://bit.ly/1nheV4e>>. Acesso em: 20 abr. 2015.
- Carvalho, M. L. B. *et al.* (2013) *Pensamento Computacional no Ensino Médio Mineiro*. Belo Horizonte, MG- 2013. Disponível em < <http://bit.ly/1I9eqoC>>. Acesso 20 abr. 2015.
- Marques, D. L. *et al.* (2011) *Atraindo alunos do ensino médio para a computação: Uma Experiência Prática de Introdução à Programação utilizando Jogos e Python*. In:

- Anais Do XVII Workshop sobre Educação Informática. Disponível em <<http://bit.ly/1J5NSFz>>.
- Malan, D. J. e Leitner, H. H. (2007) Scratch for budding computer scientists. Proceedings do 38th SIGCSE'07, Kentucky, USA, 2007, p. 223-227.
- Maloney, J. *et al.* (2010) The scratch programming language and environment. In: Communications of The ACM, 2010. v. 10, n. 4, Article 16. D
- MIT - Massachusetts Institute of Technology. Para os Pais. s.d. Disponível em: <<https://scratch.mit.edu/parents/>>. Acesso em: 01 abr. 2015.
- Minayo, Maria Cecília de Souza *et al.* Pesquisa social: teoria, método e criatividade. – 30 ed. Petrópolis, Rio de Janeiro: Vozes, 2011.
- Moreira, M. A. (1999) Aprendizagem significativa. Editora Universidade de Brasília. Brasília, 1999.
- Oliveira, E. C. de L. (2009) O Uso do Software Scratch no Ensino Fundamental: Possibilidades de Incorporação Curricular Segundo Professoras dos Anos Iniciais. PUC-Minas, Belo Horizonte, 2009.
- Pinto, António Sorte. Scratch na aprendizagem da matemática no 1.º ciclo do ensino básico: estudo de caso na resolução de problemas. Braga: Universidade do Minho, 2010.
- Souza, R. M. e Lencastre, J. A. (2013). Desenvolvimento do Pensamento Computacional com Recurso ao Scratch: Uma Experiência com Alunos do 8º Ano. Universidade do Minho. Atas do XII Congresso Internacional Galego-Português de Psicopedagogia. Braga: Universidade do Minho, 2013.
- The Royal Society. (2012). Desligar ou reiniciar? O caminho a seguir para a Computação no Reino Unido Escolas. Disponível em: <<http://bit.ly/1KbIXkX>>
- Wing, J. M. Computational thinking. In: Communications of The ACM, Março 2006. Vol. 49, nº 3, p. 33-35.