

caracteriza de transposição informática o processo que conduz a especificação, e posterior representação de um modelo de conhecimento, considerando os requisitos básicos da representação simbólica da computação. Em outras palavras, a transposição informática traduz-se num processo em que o conhecimento, tendo origem em um saber sábio de referência, e sendo dirigido para determinado saber aprendido pelo estudante, passa necessariamente por uma modelagem computacional (SILVA e RAUEN, 2010).

A transposição informática integra a dimensão didática e informática nos processos de ensino e aprendizagem, favorecendo repensar a estrutura educativa, o tipo de atividades e recursos didáticos utilizados em sala de aula, bem como os conteúdos ensinados, o papel do professor e da educação frente ao avanço tecnológico mundial. A partir da introdução dos recursos tecnológicos na educação, através de jogos e softwares educativos, tem-se a oportunidade de redimensionar as concepções sobre a aprendizagem, supondo-se alterações não só na organização habitual da sala de aula, como também, na maneira de se estruturar o processo educativo.

Talvez a grande contribuição dos aparatos tecnológicos para educação, além de oportunizar um maior acesso à informação, seja possibilitar novas formas de representação de objetos e a exploração dinâmica de diferentes situações educativas. A simulação parece ser a característica fundamental do recurso informatizado, pois compreende a experimentação, os ensaios e os erros, valorizando a operacionalidade e eficiência; enfatiza o papel dos modelos relativos e provisórios como forma de representação da realidade, em contraposição à imobilidade das teorias. (BITTENCOURT, 1998). Nesse sentido, o Scratch é um software que possibilita a exploração e representação de diferentes objetos e lógicas expressas pelo estudante, através dos comandos dados pelo mesmo.

A criação dos registros de representação no computador tem por objetivo facilitar a compreensão de conceitos. No entanto, segundo Balacheff (2000) a definição desses novos registros de representação precisa de um estudo epistemológico aprofundado, a fim de se conhecer melhor os sistemas de representações dos conceitos desenvolvidos, assim como suas propriedades e características. Esse mesmo autor defende que a criação de novos registros semióticos deve garantir a *vigilância epistemológica*, isto é, a necessidade de que o modelo representado por esse registro seja coerente. Ele entende que a questão da coerência tem duas dimensões: a completude e a adequação. A completude significa que qualquer objeto constituível a partir do modelo é representável. A adequação significa que qualquer representação é a representação de um objeto possível do modelo (BELLEMAIN, 2000).

4. Pensamento computacional: uma habilidade para todos

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, de acordo com as Diretrizes Curriculares Nacionais para os cursos de graduação em Computação, no Parecer nº 136/2012 (Brasil, 2012). Contudo, acompanhar e avaliar o desenvolvimento do pensamento computacional não é um processo trivial. Este é um dos tópicos mais discutidos por educadores e pesquisadores da área (MORENO-LEON e ROBLES, 2015). Brennan e Resnick (2012) apontam alguns conceitos a serem observados ao avaliar-se o desenvolvimento do pensamento computacional. Uma breve descrição de cada conceito é descrita a seguir:

- a) Sequências: expressar uma tarefa em uma série de passos.
- b) Laços (loops): mecanismo utilizado para repetição de sequências.
- c) Eventos: situações que dão início a outras.

- d) Paralelismo: sequências que acontecem ao mesmo tempo.
- e) Condicionais: habilidade de fazer decisões baseado em certas condições.
- f) Operadores: operadores matemáticos, relacionais e lógicos.
- g) Dados: elementos que o computador armazena, recupera e atualiza.

Brennan e Resnick (2012) também propõem algumas formas de abordar a avaliação do pensamento computacional. São elas a análise de portfólio de projetos, entrevistas, e a criação de cenários, onde o estudante deve resolver erros e outros tipos de atividades (cenários) planejadas. Outros métodos de avaliação são propostos em diversos trabalhos da área (WILSON, HAINEY, CONNOLLY, 2012), especialmente quando o pensamento computacional é desenvolvido e expresso no Scratch.

Um recurso de avaliação específico para o Scratch foi desenvolvido por pesquisadores da Universidad Rey Juan Carlos, em Madri, Espanha. Este recurso, batizado de Dr. Scratch³ está disponível online. O Dr. Scratch utiliza como método a avaliação da presença de certos elementos no projeto, através de algoritmos específicos para avaliar projetos Scratch. Os critérios de avaliação do software estão listados na tabela 1 (traduzida de MORENO-LEÓN e ROBLES, 2015). O Dr. Scratch avalia os projetos quanto a estes 7 conceitos (tabela 1), atribuindo uma nota de 0 (não está presente) a 3 (proficiente). Ele considera a qualidade do projeto de acordo com pontuação feita (até um total de 21 pontos).

Tabela 1: Nível de desenvolvimento para cada conceito

Conceito	Básico	Em Desenvolvimento	Proficiente
Abstração e decomposição de problemas	Mais que uma programação e mais de um objeto	Definição de blocos	Uso de clones
Paralelismo	Dois programas são executados quando a bandeira verde é pressionada	Dois programas são executados quando uma tecla é pressionada ou quando um objeto é clicado	Dois programas são executados na verificação da ocorrência de um evento (exemplo, ao criar clone, ao mudar fundo, etc)
Pensamento Lógico Condicional	Se	Se, senão	Uso de expressões lógicas
Sincronização	Esperar	Anunciar, receber mensagem, parar tudo, parar programa, parar programa do objeto	Esperar até, quando o plano de fundo mudar para, anunciar e esperar
Controle de Fluxo	Sequência de blocos	Repetir, sempre	Repetir até
Interatividade com o Usuário	Bandeira verde	Tecla pressionada, objeto clicado, pergunte e espere, blocos de mouse	Quando %s é > %s, vídeo, áudio
Representação de Dados	Modificador de propriedades de objetos	Operações com variáveis	Operações em listas

5. Materiais e Método

A fim de contribuir com a formação docente, desenvolveu-se um trabalho orientado que

3 Site web: <http://drscratch.programamos.es/>

envolveu utilizar e avaliar o software Scratch seguindo critérios tecnológicos e pedagógicos. O desenvolvimento deste estudo ocorreu em três etapas descritas a seguir.

5.1 Primeira etapa: constituição do grupo amostral e introdução aos Scratch

Na primeira etapa foi constituída a amostra de professores avaliadores voluntários para o estudo. Os 6 professores participantes atuam no ensino de Ciências ou Matemática no ensino fundamental ou médio. Ainda na primeira etapa do trabalho os professores foram apresentados ao Software Scratch. Durante o período de 4 horas os professores receberam explicações sobre o uso, realizaram testes e posteriormente produziram exemplares de projetos. Mesclou-se atividades individuais e em grupo, com a finalidade de que uns auxiliassem aos outros nesta etapa. Assim, todo o grupo pode avançar de forma homogênea.

5.2 Segunda etapa: avaliação técnico-pedagógica do Scratch

Na segunda etapa do trabalho foi realizada a avaliação *técnico-pedagógica* do software. Para esta avaliação foi utilizado um instrumento para avaliação de software educacional (Webber et al., 2009). O instrumento compreende vinte e quatro questões, sendo que 5 avaliam aspectos pedagógicos e 19 aspectos técnicos. Ao final do instrumento há um espaço para comentários. Todos os professores participantes do estudo realizaram a avaliação do software. Posteriormente foram formados grupos entre os participantes para discussão em grupo e análise sobre os resultados das avaliações.

5.3 Terceira etapa: avaliação epistemológica do Scratch

Na terceira etapa do trabalho partiu-se para a avaliação epistemológica do software. Para esta avaliação se considerou o conceito de transposição informática proposto por Balacheff (2013) que trata da validade epistemológica do que pode ser ensinado por meio de um software educacional. Posteriormente foram formados grupos entre os participantes para discussão em grupo e análise sobre os resultados das avaliações.

5.4 Quarta etapa: autoavaliação

Ao término das avaliações foi realizado um encontro para reflexões sobre usos e percepções do software Scratch no ensino das Ciências e da Matemática.

6. Resultados

6.1 Primeira etapa: constituição do grupo amostral e introdução aos Scratch

A primeira etapa do trabalho iniciou com a apresentação do software, realização de testes, uso de simulações e consulta a material de referência. Em seguida partiu-se para a realização de uma atividade prática envolvendo o planejamento de uma tarefa com o software Scratch. Cada professor foi convidado a refletir e escrever sobre os seguintes pontos: a) qual a sequência de atividade(s) prevista(s)? b) qual o objetivo desta(s) tarefa(s)? c) quais os conhecimentos prévios necessários ao estudante? d) o que você espera que ele construa, reflita, analise e conclua com esta tarefa? e) como a tarefa será avaliada?

Concluída a elaboração da tarefa, cada professor foi convidado a testar a tarefa, se colocando no papel de estudante, e avaliando os seguintes pontos: a tarefa está bem descrita? O nível de dificuldade está adequado? O conhecimento visado está sendo utilizado? As reflexões esperadas emergirão ao longo da realização da tarefa? A realização desta primeira etapa sensibilizou os professores para os potenciais da ferramenta, bem como das dificuldades no seu uso.

6.2 Segunda etapa: avaliação técnico-pedagógica do Scratch

A avaliação do software Scratch foi realizada por meio de um instrumento de avaliação de software educativo, composto por questões considerando aspectos pedagógicos e computacionais. Esse instrumento de avaliação foi proposto para o

desenvolvimento da FASE, uma ferramenta especialista para avaliação de software educacional (WEBBER; BOFF; BONO, 2009).

Cada professor atuou com um *avaliador* do software. Inicialmente, com relação aos aspectos pedagógicos, a primeira questão do instrumento de avaliação indaga sobre a proposição, por parte do software analisado, de situações-problemas que envolvam a formulação de hipóteses, a investigação e/ou a comparação. Metade dos professores pesquisados respondeu que o Scratch atende parcialmente a essa questão, enquanto a outra metade considerou que não se aplica ao Scratch.

Nas questões sobre a adequação do programa ao nível do público-alvo e o favorecimento da utilização interdisciplinar, todos os avaliadores consideraram ser adequado. Outra questão pedagógica busca verificar se o software motiva o usuário a buscar outras informações em diferentes fontes de pesquisa. Dos participantes, 50% consideraram que o Scratch motiva essa busca enquanto a outra metade dos avaliadores considerou que instiga parcialmente. A última questão desse grupo busca saber se o software oferece atividades variadas, diversificando os níveis de complexidade das tarefas. A maioria dos avaliadores (67%), respondeu que essa questão não se aplica ao Scratch enquanto que 33% considerou que o software não apresenta essas características. Houve portanto dificuldade na compreensão do que foi solicitado.

Na etapa seguinte as questões abordaram aspectos técnicos do software. Todos os professores que responderam o questionário afirmaram que o usuário possui controle sobre o Scratch, podendo interromper as tarefas e retomá-las posteriormente, possibilitando a correção de erros. Também foram unânimes com relação à adequação da apresentação de textos à sua leitura. Já na questão que aborda a utilização de imagens pelo software, sua relação e adequação aos conteúdos, 50% dos participantes consideraram que as imagens são adequadas e bem empregadas enquanto a outra metade dos avaliadores considerou que as imagens são apenas parcialmente adequadas.

Os participantes foram unânimes quanto à qualidade gráfica, adequação do layout da tela e bom emprego dos recursos de animação, já que os mesmos estimulam a utilização do programa pelos usuários. Também concordam que o Scratch possui ajudas e dicas que facilitam a utilização do software e esclarecimento de dúvidas.

Quando o instrumento de avaliação questiona sobre a clareza e precisão das instruções fornecidas pelo software, metade dos pesquisados considerou que essa questão não se aplica ao Scratch. Por outro lado, 33% dos participantes consideraram as instruções claras e precisas, e 17% as consideraram parcialmente claras e precisas.

Todos os participantes consideraram que o software avaliado opera de acordo com as instruções fornecidas na documentação. Já a questão de adequação da linguagem ao público-alvo, ao tópico/conteúdo e características regionais, 50% dos participantes consideraram a linguagem adequada, enquanto a outra metade dos participantes a considerou parcialmente adequada, justificando que alguns termos não foram traduzidos do inglês e na ajuda do programa vários blocos estão em inglês.

Todos os participantes consideraram usuais as convenções e símbolos utilizados para guiar o usuário na interface. Também foram unânimes em afirmar que o Scratch possui bom tratamento de erros, ou seja, quando um erro ocorre o sistema consegue contorná-lo e possibilita ao usuário cancelar a operação ou retornar a um ponto anterior. Outra questão de consenso foi com relação à utilização de recursos multimídias (textos imagens, filmes e sons) para apresentar os assuntos.

Quando o instrumento de avaliação questiona sobre a adequação do vocabulário ao público-alvo, metade dos participantes considerou o vocabulário adequado, enquanto que 50% considerou apenas parcialmente adequado ressaltando

que alguns termos não são usuais para crianças e algumas palavras estão em inglês.

Foram unânimes em afirmar que o conteúdo é adequado ao nível do estudante. Já com relação à observação da correção ortográfica por parte do software, 83% dos participantes consideraram que o Scratch observa a correção ortográfica enquanto que 17% considerou que essa questão não se aplica ao software analisado. Provavelmente a forma como a pergunta foi interpretada resultou nessa escolha.

Outro aspecto muito importante a ser avaliado é sobre a existência de manuais disponíveis sobre o software. Nesse sentido, foram considerados o manual do usuário, o manual de instalação, e o manual pedagógico para o professor. Todos os avaliadores consideram que o software disponibiliza de manual de usuário e também de manual pedagógico para o professor, como o Guia Curricular ScratchMIT, que tem como título “Computação criativa: uma introdução ao pensamento computacional baseada no conceito de design”. Apenas sobre a existência de um manual de instalação que a minoria (17%) dos avaliadores considera parcialmente, enquanto a maioria (83%) considera que sim.

Quando questionados se os alunos conseguem utilizar o software sem a interferência do professor, a minoria (33%) considera que é possível, enquanto a maioria (67%) considera que parcialmente, tendo em vista alguns aspectos como a idade e o conhecimento prévio dos alunos. Além disso, cabe considerar que o estudante pode pesquisar materiais como apostilas, tutoriais, vídeos explicativos, além de exemplos que podem guiar o estudante na utilização do software.

6.3 Terceira etapa: avaliação epistemológica do software

A avaliação em termos epistemológicos do software permitiu ampla e profunda reflexão por parte dos professores. As observações e percepções foram anotadas e são aqui transcritas. Foi consenso afirmar que a grande contribuição dos aparatos tecnológicos para educação, além de oportunizar um maior acesso à informação, é de possibilitar novas formas de representação de objetos e a exploração dinâmica de diferentes situações educativas. A simulação parece ser a característica fundamental do recurso informatizado, pois compreende a experimentação, os ensaios e os erros, valorizando a operacionalidade e eficiência; enfatiza o papel dos modelos relativos e provisórios como forma de representação da realidade, em contraposição à imobilidade das teorias (BITTENCOURT, 1998). Nesse sentido, o Scratch foi considerado um software que possibilita a exploração e representação de diferentes objetos e lógicas expressas pelo estudante, por meio dos comandos dados pelo mesmo.

O domínio de validade epistemológica dos dispositivos informáticos relaciona-se às condições de aprendizagem que o dispositivo oferece, isto é, quais conhecimentos o software possibilita ao estudante construir. Nesse sentido, um dos aspectos relevantes a ser considerado consiste em verificar se as ferramentas tecnológicas permitem a interação e a identificação dos caminhos percorridos pelo estudante, fornecendo respostas sobre as atividades realizadas, bem como o *feedback* em relação ao seu desempenho. Pode-se dizer que o Scratch atende ao requisito de interação e identificação dos caminhos percorridos, uma vez que permanecem registrados e visíveis todos os comandos feitos pelo estudante no decorrer do uso do software, permitindo que se reavalie, a todo momento, suas ações e os resultados dela decorrentes.

Balacheff (2000) afirma que o processo de construção do conhecimento ocorre na interação entre os alunos e o computador, de modo que somente a leitura da tela, sem a manipulação e a exploração dos conteúdos, não será relevante para a aprendizagem. Ou seja, o processo de transposição informática, além de modificar as

relações entre o estudante e o objeto do conhecimento, através da interação com a máquina, pode possibilitar que outros processos cognitivos sejam desenvolvidos, desencadeando, assim, a construção de novas estruturas de raciocínio. Considerando este aspecto, o Scratch é extremamente válido por permite a manipulação de seus objetos na tela e a simulação de múltiplas execuções.

Alguns softwares educativos denominados micromundo⁴ têm se apresentado como um importante recurso didático, no sentido de oferecer aos estudantes, formas diferentes de atribuir significado aos assuntos desenvolvidos em aula. Ainda segundo Balacheff (2000), eles modificam as estratégias didáticas, o modo de ensinar e de avaliar, como também, possibilitam ampliar o conhecimento do estudante pela oportunidade de investigação e raciocínio que o software oferece. O Scratch faz parte dessa categoria de softwares educativos, uma vez que favorece ao estudante a possibilidade de construir uma sólida ponte entre as representações simbólicas dos objetos (do próprio software) e a representação gráfica que se mostra na tela do computador. Conforme aponta Balacheff (2000, p. 94, tradução nossa), “o conhecimento não pode simplesmente ser lido na tela, ele é resultado de uma construção no processo de interação com a máquina.”

6.4 Quarta etapa: autoavaliação

Nesta etapa voltou-se ao cerne das discussões: a aprendizagem das Ciências e da Matemática. Na percepção dos professores ficou clara a importância dos conhecimentos prévios sobre linguagem de programação para o planejamento das tarefas com o Scratch. Os professores da área de Matemática tiveram maior facilidade no manuseio do Scratch e maior aderência de seus conteúdos ao software. Contudo, professores de outras disciplinas (Ciências, Biologia e Química) também vislumbraram um grande número de situações de aprendizagem que poderiam ser desenvolvidas no Scratch. Um fato observado foi que por meio da programação os estudantes podem desenvolver simulações, testando parâmetros (variando valores) e assim assimilar de forma real comportamentos que em sala de aula são trabalhados de forma abstrata. Com a mediação do professor, os estudantes podem elaborar várias hipóteses para explicar um dado fenômeno. Em seguida, podem construir um ou mais modelos do fenômeno no Scratch. A partir das sucessivas execuções, os estudantes podem testar, comparar e inferir qual dos modelos (oriundos das hipóteses) é o mais correto para explicar o fenômeno estudado. Como exemplos dados pelos professores pode-se citar problemas em Física (trajetória de um projétil), Química (movimento das partículas) e nas Ciências (estados físicos da água). Estes fenômenos e tantos outros podem ser explorados no Scratch na forma de projetos de jogos ou simulações interativas.

7. Conclusões

Este trabalho apresentou um estudo avaliativo sobre o Software Scratch aplicado ao ensino visando auxiliar professores na inclusão deste recurso em suas atividades docentes. Ao professor pode ser uma opção por meio da qual ele pode promover a aprendizagem, avaliações, atividades variadas e diversificar suas aulas. Além dos conteúdos curriculares, os estudantes podem por meio do Scratch desenvolver o pensamento computacional adquirindo aptidão para o desenvolvimento de aplicações, e

4 “[...] o termo micromundo foi inicialmente usado para definir um sistema que permite simular ou reproduzir um domínio do mundo real, e que tem como objetivo abordar e resolver uma classe de problemas.” (BELLEMAIN, 2002, p. 54). O micromundo contempla a criação de atividades que estimulem o indivíduo a explorar diferentes conceitos, através da manipulação dos objetos criados.

também competências como o pensamento abstrato, algorítmico, lógico e dimensionável. Explorar programas como o *Scratch*, pode fazer com que o estudante se torne um sujeito com competências e habilidades que só podem ser alcançadas por quem tem contato com ambientes de linguagem de programação. Uma das habilidades é a capacidade de resolver problemas complexos pela decomposição em problemas menores. Segundo os professores envolvidos no estudo, esta habilidade por si só já justifica o uso do Scratch. Para finalizar, destaca-se a orientação das diretrizes curricular que reforça que a introdução do pensamento computacional e algorítmico desde a educação básica fornece os recursos cognitivos necessários para a resolução de problemas, transversal a todas as áreas do conhecimento (BRASIL, 2012).

7. Referências Bibliográficas

- BALACHEFF, N. **cKc, a Model to Reason on Learners Conceptions**. In: Martinez, M. & Castro Superfine, A (Eds.) Proc of the 35th annual meeting of the North American Chapter of the Intl. Group for the Psychology of Mathematics Education. Chicago, IL: University of Illinois at Chicago, 2013.
- _____. **Entornos informáticos para la enseñanza de las matemáticas: complejidad didáctica y expectativas**. 2000. Disponível em: http://www.cvrecursosdidacticos.com/download_file.php?file=1288115856_U2_Balacheff.pdf. Acesso em: 24 maio 2016.
- BELLEMAIN, F. A transposição informática na Engenharia de Software Educativos. 2000. I SIPEM, 22 a 25 de novembro, Serra Negra (SP).
- BITTENCOURT, J. **Informática na educação?** Algumas considerações a partir de um exemplo. Revista da Faculdade de Educação, São Paulo, vol. 24, n.1, Jan./Jun. 1998.
- BRASIL, Parâmetros Curriculares Nacionais. Matemática. **Secretaria de Educação Fundamental Brasília**, 1998.
- BRASIL, **Diretrizes Curriculares Nacionais para os cursos de Graduação em Computação**. Parecer nº 136/2012. Conselho Nacional de Educação .
- BRASIL, **Diretrizes Curriculares Nacionais Gerais da Educação Básica** / Ministério da Educação. Secretaria de Educação Básica. Diretoria de Currículos e Educação Integral, Brasília, 2013.
- BRENNAN, Karen; RESNICK, Mitchel. New frameworks for studying and assessing the development of computational thinking. In: **Proceedings of the 2012 annual meeting of the American Educational Research Association, Vancouver, Canada**. 2012. p. 1-25.
- FERNANDES, G. W. R. **Práticas pedagógicas mediatizadas: delineando caminhos para a formação de professores de física na modalidade à distância**. Dissertação (Mestrado). UFSC, Florianópolis, 2007.
- 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
- MORENO-LEÓN, Jesús; ROBLES, Gregorio. Analyze your Scratch projects with Dr. Scratch and assess your computational thinking skills. In: **Scratch Conference**. 2015. p. 12-15.
- SCAICO, P.D., HENRIQUE, M.S., CUNHA, F.O.M., ALENCAR, Y.M. Um Relato de Experiências de Estagiários da Licenciatura em Computação com o Ensino de Computação para Crianças. **RENOTE - Revista Novas Tecnologias na Educação**, v. 10, n. 3, 2012.
- SILVA, C. R. da; RAUEN, F. J. **Desenvolvimento de aplicativos para o ensino de função com base na teoria de registro de representação semiótica**. Anais do X Encontro Nacional de Educação Matemática. Salvador/BA, 2010.
- WEBBER, C.G.; BOFF, E.; BONO, F. **“Ferramenta Especialista para Avaliação de Software Educacional”**. In XX Simpósio Brasileiro de Informática na Educação, 2009.
- WILSON, Amanda; HAINEY, Thomas; CONNOLLY, Thomas. Evaluation of computer games developed by primary school children to gauge understanding of programming concepts. In: **European Conference on Games Based Learning. Academic Conferences International Limited**, 2012. p. 549.
- WING, J.M. Computational Thinking. COMMUNICATIONS OF THE ACM March 2006/Vol. 49, N. 33.