

SCRATCH: UMA FERRAMENTA CONSTRUCIONISTA NO APOIO A APRENDIZAGEM NO SÉCULO XXI

SCRATCH: A CONSTRUCTIONIST TOOL TO SUPPORT LEARNING ON THE 21st CENTURY

Marize Lyra Silva Passos *

Instituto Federal do Espírito Santo – Campus Serra (*)
Coordenadoria de Informática
E-mail: marizes@terra.com.br

Resumo

Este artigo trata como o Construcionismo pode ser usado no contexto educacional utilizando como ferramentas a linguagem de programação Scratch, que é um ambiente de autoria multimídia que permite que os usuários, principalmente com menos idade, possam resolver problemas através de projetos pessoalmente significativos, tais como histórias animadas e jogos e também desenvolverem varias habilidades que são necessárias a aprendizagem no século XXI.

Palavras-chave: scratch. construcionismo. habilidades de aprendizagem para o século XXI. informática na educação.

Abstract

This article discusses how can constructionism be used in educational context using Scratch's programming language as a tool, which is a multimedia authoring environment that allows users, especially at younger ages, to be able to solve problems through personally meaningful projects such as animated stories and games and also to develop various skills that are required to learning in the twenty-first century.

Keywords: scratch. constructionism. learning skills. computers in education.

1. INTRODUÇÃO

A abordagem tradicional do processo ensino-aprendizagem baseada na transmissão do conhecimento acumulado ao longo do tempo tem sido por muito tempo a utilizada na educação. Nela os alunos são vistos como seres passivos, responsáveis apenas pela memorização dos conteúdos que lhes são transmitidos pelo professor, sem a apropriação desses, como descrito na concepção bancária de Freire (2011). Entretanto, gradativamente, essas abordagens vêm sendo substituídas por outras, como a abordagem Construtivista, que tem como principais representantes o suíço Jean Piaget e o norte-americano Jerome Bruner, e a abordagem Sociocultural, que tem como principais representantes o bielorusso Vygotsky e o brasileiro Paulo Freire.

Segundo a perspectiva freiriana, “ensinar não é transferir conhecimento, mas criar as possibilidades para a sua produção ou a sua construção” (FREIRE, 2011, p. 24). Mas, como conseguir essa transformação no processo de ensino-aprendizagem sem que haja uma transformação no modo de ensinar dos professores e sem o apoio de ferramentas pedagógicas motivadoras que levem os alunos a uma aprendizagem efetiva. O uso na prática pedagógica de diversas tecnologias como a televisão, o rádio, o computadores, a internet, entre outras, traz para a sala de aula o cotidiano dos alunos tornando o processo de ensino-aprendizagem mais atraente e significativo. É preciso ir além do cognitivo para motivar os alunos, deve-se, também, levar em conta a afetividade para que o ele possa aderir às propostas de ensino e, dessa forma, o aprendizado possa ocorrer.

O uso dessas novas tecnologias, no contexto escolar, pode ser de grande valia uma vez que os alunos mantêm uma relação afetiva com esses novos aparatos tecnológicos e podem ampliar seus conhecimentos por meio das interações com os amigos em rede. Mas, não se pode esquecer que essas ferramentas, por si, só não são capazes de tornar a aprendizagem efetiva se não houver um efetivo planejamento e uma intervenção pedagógica criativa para o seu uso.

Além disso, os estudantes de hoje estão crescendo em um mundo que é muito diferente do mundo de seus pais e avós. Para ter sucesso na Sociedade Criativa de hoje, os alunos devem aprender a pensar de forma criativa, planejar sistematicamente, analisar criticamente, trabalhar de forma colaborativa, comunicar de forma clara, atuando de forma iterativa, e aprendendo

continuamente. Infelizmente, a maioria dos usos de tecnologias nas escolas, hoje, não suportam essas habilidades de aprendizagem do século XXI. Em muitos casos, as novas tecnologias são simplesmente reforçar velhas formas de ensinar e aprender.

Nesse novo contexto, o professor, além de saber utilizar o computador, deve ter a capacidade de selecionar os melhores aplicativos a serem utilizados como apoio pedagógico a aprendizagem. Nessa perspectiva, esse artigo se propõe a apresentar o ambiente de autoria multimídia programável Scratch como uma ferramenta de apoio à educação no ambiente escolar, com o objetivo de relacionar o conteúdo programático com o cotidiano do aluno, facilitando, assim, o processo de aprendizagem baseado em uma abordagem construcionista, além de desenvolver nos alunos as habilidades necessárias à aprendizagem no século XXI.

2. O COMPUTADOR NO CONTEXTO EDUCACIONAL

No contexto educacional, o computador pode ser utilizado tanto para ensinar sobre computação como para ensinar praticamente qualquer assunto, sendo essa segunda visão o foco da informática educativa, que pode ter duas abordagens pedagógicas distintas, na primeira o computador ensina o aluno, e tem suas raízes nos métodos de instrução programa, já na segunda o aluno ensina ao computador e nesse caso o computador é visto como uma ferramenta que permite o aluno resolver problemas ou realizar tarefas. (VALENTE, 1998).

Valente (1998) afirma que, atualmente, o uso do computador na educação não é mais visto como uma “‘maquina de ensinar’ mas, como uma nova mídia educacional: o computador passar a ser uma ferramenta educacional, uma ferramenta de complementação, de aperfeiçoamento e de possível mudança na qualidade do ensino” (p. 6). Nessa perspectiva o computador não é mais visto como o instrumento que ensina o aprendiz, mas como uma ferramenta com a qual o aluno desenvolve algo, fazendo com que o aprendizado ocorre pelo fato do aluno estar executando uma tarefa por intermédio do computador.

O computador como ferramenta de apoio ao ensino pode ser usado para diversos fins como: para o uso de aplicativos por professores e alunos; para a resolução de problemas; para a produção de

música; para o controle de processos; para facilitar a comunicação etc. Mas todos esses usos devem ter como foco a aprendizagem, ou melhor, uma aprendizagem verdadeiramente inovadora e efetiva, que segundo Moram (2009) ocorre ao

[...] descobrimos novas dimensões de significação que antes se nos escapavam, quando vamos ampliando o círculo de compreensão do que nos rodeia, quando como numa cebola, vamos descascando novas camadas que antes permaneciam ocultas à nossa percepção, o que nos faz perceber de uma outra forma. Aprendemos mais quando estabelecemos pontes entre a reflexão e a ação, entre a experiência e a conceituação, entre a teoria e a prática; quando ambas se alimentam mutuamente (p. 22).

Na Informática Educativa os aplicativos devem permitir a realização de atividades que possibilitem a interação dos alunos com os computadores para, a partir de suas ações, obterem o conhecimento esperado e desenvolverem habilidades necessárias a aprendizagem no século XXI. Além disso, para o uso dos computadores como ferramenta de apoio pedagógico no auxílio ao processo de construção do conhecimento é necessário que professores e alunos aprendam a usá-lo no desenvolvimento de componentes curriculares como um meio e não como um fim, transformando o computador em um poderoso recurso de suporte à aprendizagem abrindo inúmeras possibilidades pedagógicas (LOVATE e NOBRE, 2011; RUSK, RESNIK E MALONEY, 2013).

3. O CONSTRUCIONISMO E O CONSTRUTIVISMO

Aqui será dada ênfase à abordagem construtivista de Piaget, predominantemente interacionista, na qual o conhecimento é considerado como uma construção continua na qual “[...] a passagem de um estado de desenvolvimento para o seguinte é sempre caracterizada por formação de novas estruturas que não existiam anteriormente no indivíduo” (MIZUKAMI, 2011, p. 63-64) Sendo o conhecimento humano essencialmente ativo.

[...] Conhecer um objeto é agir sobre ele e transformá-lo, apreendendo os mecanismos dessa transformação vinculados com as ações transformadoras. Conhecer é, pois, assimilar o real às estruturas de transformações, e são as

estruturas elaboradas pela inteligência enquanto prolongamento direto da ação (PIAGET, 1970a, p. 30).

O processo educacional segundo a teoria de Piaget

[...] tem um papel importante, ao provocar situações que sejam desequilibradoras para o aluno, desequilíbrios esses adequados ao nível de desenvolvimento em que se encontram, de forma que seja possível a construção progressiva das noções e operações, ao mesmo tempo em que a criança vive intensamente (intelectual e afetivamente) cada etapa de seu desenvolvimento (MIZUKAMI, 2011, p. 70).

Mizukami (2011) afirma que para Piaget o ensino está baseado “no ensaio e no erro, na pesquisa, na investigação, na solução de problemas por parte do aluno, e não em aprendizagem de fórmulas, nomenclaturas, definições etc.” (p. 76). E para ele a aprendizagem verdadeira só ocorrerá, realmente, quando o aluno elabora seu conhecimento. Esta autora ao descrever a abordagem pedagógica construtivista afirma que “as experiências não devem ser feitas na frente do aluno; Devem ser feitas pelos alunos” (p. 81).

Segundo Valente (1998), o uso do computador como máquina de ensinar tem sido, erroneamente, associada por alguns educadores como uma abordagem construtivista, mas na verdade continua sendo uma abordagem instrucionista. Para evitar esse tipo de engano “Papert denominou de construcionista a abordagem pela qual o aprendiz constrói, através do computador, o seu próprio conhecimento” (p. 40)

No construcionismo de Papert, a construção do conhecimento acontece quando o aluno constrói um objeto de seu interesse, e essa noção se difere do construtivismo piagetiano por duas ideias principais, primeiramente é o aprendiz que ao fazer, “colocar a mão na massa” constrói algum objeto real (ou virtual) com o apoio de computadores, e segundo o fato do aprendiz estar muito motivado por estar construindo algo que realmente é de seu interesse. Por isso no construcionismo o envolvimento afetivo torna a aprendizagem mais significativa (VALENTE, 1998). As interações que ocorrem entre alunos e computador propiciam um ambiente bastante rico e efetivo do ponto de vista da construção do conhecimento.

Na concepção de Papert, para que um aplicativo seja considerado construcionista, é necessário que ele tenha algumas características que facilitem as atividades de descrição, execução e depuração visando à solução de problemas (VALENTE, 1998).

A **descrição** é a fase na qual a solução do problema é descrita através de procedimentos. A fase de **execução** ocorre quando o aluno observa está sendo executado pelo computador e analisa os resultados obtidos. Essa fase gera reflexões que podem produzir diversos níveis de abstração nos alunos. O nível de abstração mais simples é a abstração empírica, que permite ao aluno extrair informações do objeto, tais como a cor e a forma do objeto. A abstração pseudo-empírica permite ao aluno deduzir algum conhecimento da sua ação ou do objeto. “A abstração reflexiva permite a projeção daquilo que é extraído de um nível mais baixo para um nível cognitivo mais elevado ou a reorganizado desse conhecimento em termos de conhecimento prévio (abstração sobre as próprias ideias dos alunos)” (VALENTE, 1998, p. 41-42). Quando os resultados não são os esperados para a solução do problema, o aluno deve realizar a **depuração** de sua solução.

Segundo Valente (1998) essas fases não ocorrem simplesmente colocando os alunos de frente de um computador, é necessário que a interação entre aluno computador seja mediada por um profissional que conheça o aplicativo, tanto do ponto de vista computacional, pedagógico e psicológico, não se esquecendo de que esse aluno é um ser social e está inserido, também, em um ambiente social. O mediador só conseguirá ser efetivo se agir na zona de desenvolvimento proximal do aluno que foi definida por Vygotsky como

[...] a distância entre o nível de desenvolvimento cognitivo real do indivíduo, tal como medido por sua capacidade de resolver problemas independentemente, e o seu nível de desenvolvimento potencial, tal como medido por meio da solução de problemas sob orientação (de um adulto, no caso de uma criança) ou em colaboração com companheiros mais capazes (VYGOTSKY, 1988, p. 97).

Para Valente (1998) o suporte teórico que ampara a teoria construcionista basea-se, não somente na teoria construtivistas de Piaget, mas também na teoria sócio interacionista de Vygostky.

4. O SCRATCH

O projeto iniciou em 2003 e, em 2007, o ambiente multimídia Scratch e o primeiro site foram lançados. O Scratch é gratuito e está disponível em quase 50 idiomas, inclusive no português. Ele foi desenvolvido pelo Grupo *Lifelong Kindergarten* no Media Lab do *Massachusetts Institute of Technology* (MIT), com o apoio financeiro da *National Science Foundation*, Microsoft, Intel Foundation, Fundação MacArthur, Google, Iomega e MIT consórcios de pesquisa do *Media Lab* (SCRATCH, 2013).

Mais de dois milhões de cópias do aplicativo já foram baixados do site oficial do Scratch (MALONEY *et al.*, 2010), nele, hoje, há mais 1,5 milhões de membros registrados, 459 mil criadores do projeto, 3,3 milhões de projetos enviados, 58 milhões de scripts e 20 milhões de objetos. Cerca de 1.500 novos projetos são enviados para o site todos os dias, em média, mais do que um novo projeto a cada minuto. Seus membros encontram-se espalhados por todo o mundo como se pode ver no mapa da figura 1.

Figura 1. Locais de acesso ao site oficial do Scratch no mundo.



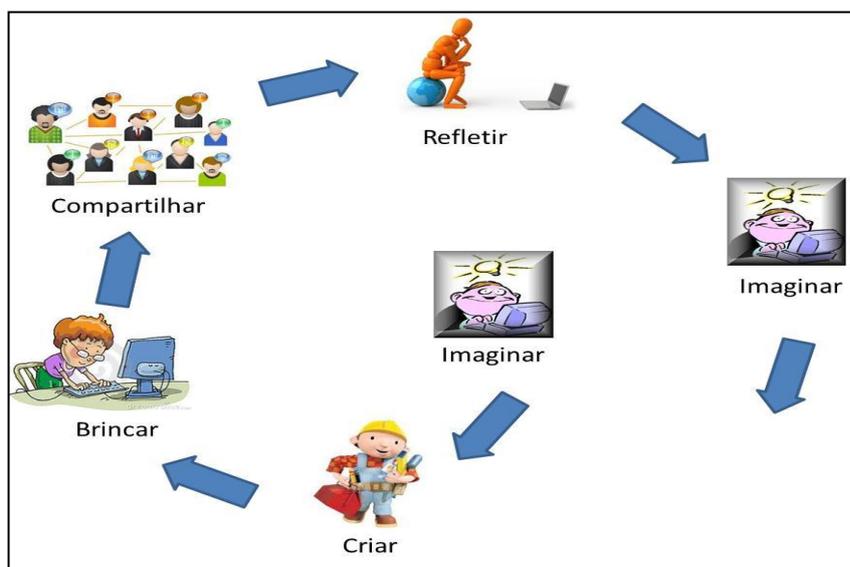
Fonte: site oficial do Scratch: <http://stats.scratch.mit.edu/community/geography.html>.

O Scratch foi criado para dar suporte ao que Resnick (2007) chamou de “Espiral do Pensamento Criativo”, vista na Figura 2 a seguir. Neste processo, as pessoas **imaginam** o que eles querem fazer, **criam** um projeto com base em suas ideias, **brincam** com as suas criações, **partilham** suas ideias e criações com os outros, **refletem** sobre as suas experiências e tudo isso os levam a imaginar novas

ideias e novos projetos. Quando os alunos passam por esse processo, mais e mais, aprendem a desenvolver suas próprias ideias, a testá-las, a testar seus limites, a experimentar alternativas e a gerar novas ideias baseadas em suas experiências.

Todas essas fases são suportadas pelo Scratch, que permite a criação e a aplicação de soluções para os mais diversos tipos de problemas de forma fácil através do uso de blocos de comando. Esses projetos podem ser compartilhados com toda a comunidade Scratch, através de sua disponibilização no site oficial do projeto e os alunos podem individualmente ou em grupos refletir sobre a solução e a partir da qual propor melhorias em seu projeto ou até mesmo a construção de novos projetos.

Figura 2. Espiral do Pensamento Criativo de Resnik.

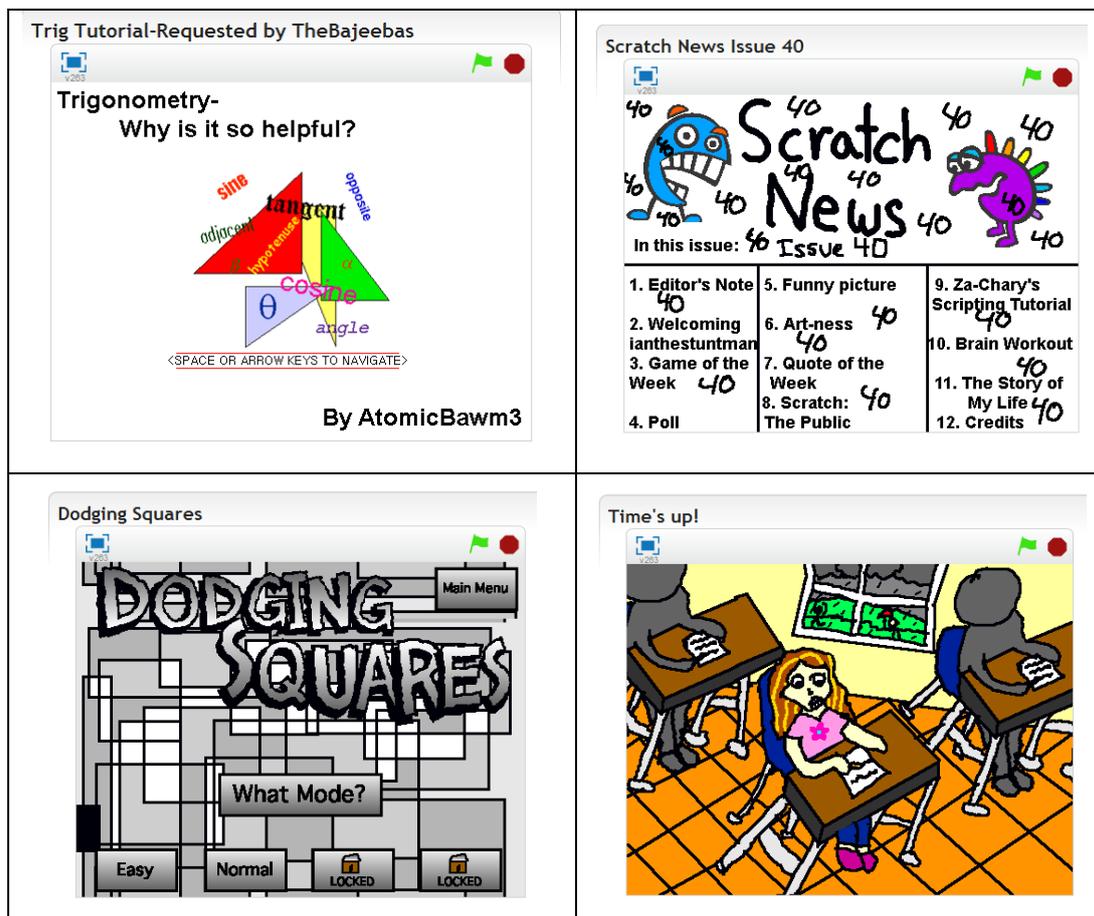


Fonte: adaptado de Resnik (2007).

O Scratch dá aos alunos o poder de criar e controlar as coisas no mundo virtual. Para muitos estudantes, a Web é essencialmente um lugar para navegar, clicar, e bater papo etc. Com o Scratch, eles mudam de consumidores de mídia para produtores de mídia, passando, então, a criar seus projetos que são muito diversificados, incluindo jogos de vídeo, boletins informativos interativos, simulações científicas, passeios virtuais, cartões de aniversário, concursos de dança animadas e tutoriais interativos, e em seguida, compartilham suas criações na Internet e podem aprender e desenvolver ideias matemáticas e computacionais, além de aprender a pensar de forma criativa, a

raciocar de forma sistemática, e a trabalhar de forma colaborativa. (RESNICK, 2007; RESNIK e MALONEY, 2009; SCRATCH, 2013).

Figura 3. Imagens de projetos Scratch criados por usuários, incluindo um tutorial sobre trigonometria, um show de notícias, um jogo e uma história interativa.



Fonte: projetos disponíveis no site: <http://scratch.mit.edu/channel/>.

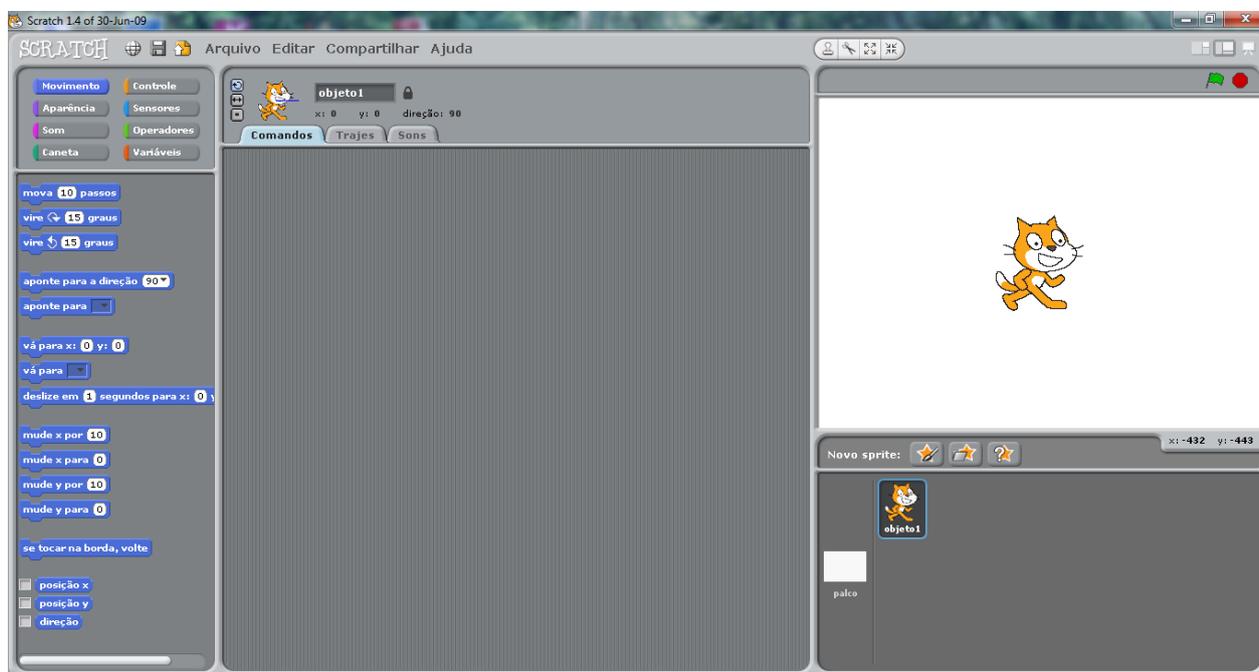
O Scratch foi concebido baseado nas ideias construcionistas dos aplicativos Logo e Etoys (MALONEY *et al.*, 2010) e é uma linguagem de programação que torna fácil a criação de projetos contendo mídia e objeto, que além de serem importados podem ser criados através de ferramenta própria para criação de imagens e gravador de som. A programação é feita pela junção de blocos de comandos coloridos para controlar objetos gráficos 2-D que se movendo em um fundo chamado de palco. Projetos Scratch podem ser salvos localmente no computador ou compartilhado on-line no site do Scratch.

5. O AMBIENTE DE DESENVOLVIMENTO DO SCRATCH

No Scratch, a interface de usuário se esforça para tornar a navegação mais fácil, usando para isso uma única janela, com múltiplos painéis para assegurar que os componentes principais estarão sempre visíveis, evitando paletas flutuantes, que podem ficar escondidas, e minimiza o uso de painéis que são mostrados somente quando demandado.

A Figura 4 mostra a janela do Scratch, que tem quatro painéis principais. O painel da esquerda representa a paleta de comandos, que serão incluídos em seu projeto, com botões para selecionar categorias. O painel do meio mostra os eventos (scripts) relacionados ao objeto (sprite) selecionado no momento, contém abas que permitem visualizar e editar as imagens (**trajes**) e os sons pertencentes ao objeto em edição. O painel no canto superior direito é o palco, onde a ação acontece. O painel inferior direito mostra miniaturas de todos os objetos do projeto, com o objeto selecionado destacado.

Figura 4. Tela da interface com o usuário do Scratch.



Fonte: Dados de pesquisa.

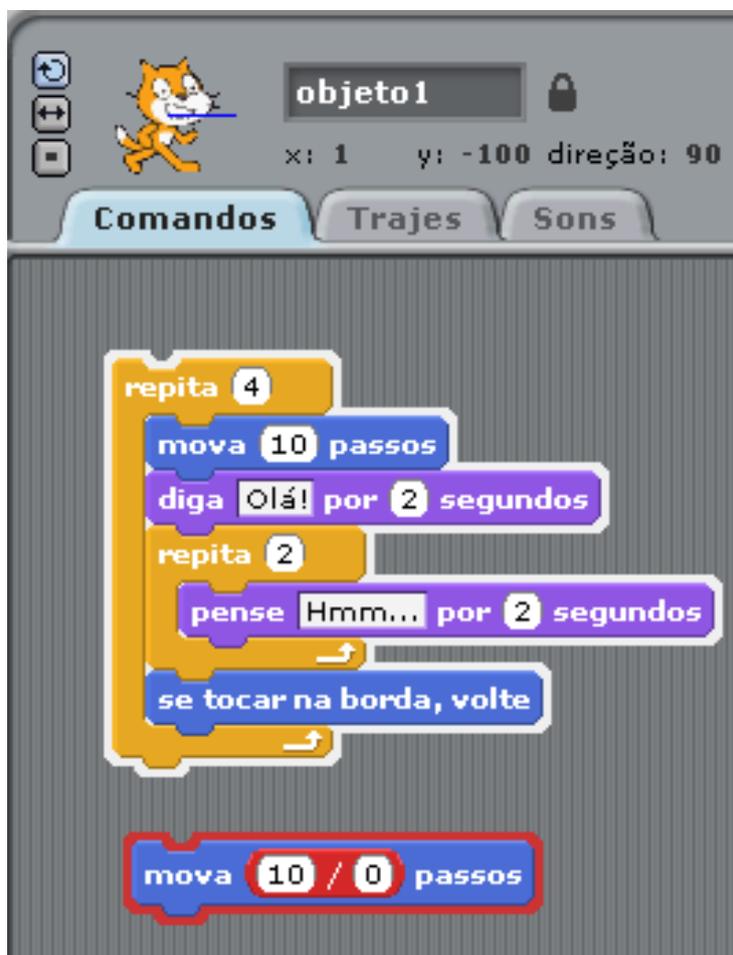
No ambiente multimídia Scratch, os alunos programam suas criações por junção de blocos gráficos, sem que nenhum dos sinais de pontuação obscuros ou sintaxe das linguagens de programação tradicionais. Desta forma, ele torna a programação acessível a um público muito mais amplo, em uma faixa etária mais jovem. (RESNICK, 2007)

No Scratch não há distinção entre as etapas de edição, compilação e execução. Os usuários podem clicar em um bloco de comando ou conjunto de blocos de comando a qualquer momento para ver o que ele faz. Na verdade, eles podem até mesmo alterar os parâmetros ou adicionar blocos a um script enquanto ele estiver sendo executado. Ao eliminar as pausas para compilação o Scratch ajuda os usuários a se envolverem mais com os testes, a depuração e conseqüentemente com a melhora de seus projetos (MALONEY et al., 2010).

Ele permite que os usuários alterem blocos de comandos e conjuntos de blocos de comandos da forma “tinkerable”, como se mexe em componentes mecânicos ou eletrônicos, essa característica incentiva a aprendizagem do tipo “mãos-a-obra” apoiada em uma abordagem *bottom-up* para a criação de scripts, onde pequenos pedaços de código são montados e testados, e posteriormente combinados em unidades maiores não exigindo que os usuários criem scripts completos antes de executar seus projetos. Fragmentos de programa podem ser deixados no painel de scripts e são salvos com o projeto. Tais fragmentos desempenham um papel semelhante ao código comentado em uma linguagem baseada em texto. Ao solucionar problemas, um script longo pode ser dividido em blocos de comandos e cada bloco testado de forma independente (MALONEY et al., 2010).

O Scratch fornece *feedback* visual para mostrar a execução do script. Quando um script é executado, ele está rodeado por uma borda branca brilhante (Figura 5). Este *feedback* ajuda o usuário a entender quando os scripts são acionados e por quanto tempo foram executados. Se ao executar um script ocorrer um erro (por exemplo, divisão por zero), a borda ao redor do bloco de comando fica vermelho e o bloco que provocou o erro é destacada em vermelho (Figura 5). Ele também pode mostrar os comandos e o fluxo de controle de forma sequencial, este efeito visual, mesmo que apenas como um breve flash na tela que fornece importantes pistas visuais para a solução de problemas do projeto.

Figura 5. Exemplos de blocos rodeados de branco quando executados e blocos cercados de vermelhos quando apresentam erros.



Fonte: Dados de pesquisa.

O Scratch não possui mensagens de erro, os erros de sintaxe são eliminados porque os blocos se encaixam apenas da forma que façam sentido, mas o aplicativo também se esforça para eliminar erros durante a execução, fazendo todos os blocos serem *failsafe*. Claro, eliminar mensagens de erro não significa eliminar erros. O usuário deve ainda pensar com cuidado antes de escrever os scripts para a solução do problema, mas mesmo quando um script não faz o que é esperado, ele faz algo, e isso é um bom começo. Essa característica do Scratch ajuda os alunos a se sentirem mais próximos da solução do problema, pois um programa que é executado, mesmo que não da forma correta, é mais motivador do que um que não funciona devido a erros de compilação (MALONEY et al., 2010).

Na maioria das linguagens de programação baseadas em texto, as variáveis são invisíveis, abstratos e difíceis de entender, já o Scratch transforma variáveis em objetos concretos que o usuário pode ver e manipular, tornando-os mais fáceis de entender através de ajustes e observação.

O Scratch se esforça para minimizar o número de blocos de comandos, pois diferente de outras linguagens de programação os blocos de comando ocupam espaços na paleta de comandos e podem dificultar a visualização da solução sem que para isso aja a necessidade de rolagem da tela. O Scratch enfatiza a simplicidade, o ambiente e o modelo de objeto e foi projetado para funcionar sem problemas, sem explicação prévia, no entanto, fazendo sentido em um exame mais detalhado. Ele tem um número surpreendentemente pequeno de comandos, especialmente levando em conta a grande variedade de tipos de projetos que apoia. Um sistema de comunicação Inter objeto de baixo acoplamento permite que objetos possam ser trocados, sem quebrar as dependências, promovendo a colaboração e compartilhamento de códigos.

O ambiente de programação e a linguagem do Scratch trabalham juntos para criar um sistema que é excepcionalmente rápido de aprender, no qual os usuários podem aprender a utiliza-lo em 15 minutos, já com alguma profundidade, e mete-lo interessado por anos.

5. SCRATCH E AS HABILIDADE DE APRENDIZAGEM DO SÉCULO XXI

Segundo o relatório “Aprendizagem para o século XXI”, existem nove tipos de habilidades de aprendizagem, divididas em três áreas-chaves que são importantes de serem desenvolvidas pelos alunos. Segundo Rusk, Resnik e Maloney (2013) ao criar seus projetos em Scratch os alunos estão desenvolvendo essas nove habilidades.

As três áreas-chaves descritas no projeto são: habilidades ligadas à informação e comunicação; habilidades de pensamento e resolução de problemas e habilidade de relacionamento interpessoal e de se auto-direcionar. As habilidades ligadas à área da informação e comunicação são (RUSK, RESNIK e MALONEY, 2013):

- *habilidades de tratar informações e trabalhar com mídias* - ao trabalhar em projetos Scratch, os alunos aprendem a selecionar, criar e gerenciar múltiplas formas de mídia, incluindo texto, imagens, animações e gravações de áudio. Como os estudantes ganham experiência com a criação de meios de comunicação, tornam-se mais perspicazes e críticos ao analisar os meios de comunicação que eles vêm no mundo ao seu redor;
- *habilidade de comunicação* - uma comunicação eficaz no mundo de hoje exige mais do que a capacidade de ler e escrever um texto. No Scratch os jovens têm que escolher, manipular e se integrarem com uma variedade de meios, a fim de se expressar de forma criativa e persuasiva.

As habilidades ligadas à área do pensamento e da resolução de problemas são (RUSK, RESNIK e MALONEY, 2013):

- *habilidade de pensar criticamente e sistemicamente* - ao aprender a programar em Scratch, os jovens tornam-se envolvidos em raciocínio crítico e pensamento sistêmico. A fim de construir projetos, os alunos precisam coordenar o tempo e as interações entre vários "sprites" (objetos em movimento). A capacidade programar as entradas interativamente e oferece aos alunos a experiência com o sensoriamento, *feedback*, e outros conceitos de sistemas fundamentais;
- *habilidade de identificar, formular e solucionar problemas* – o Scratch suporta a busca e solução de problemas contexto significativo. A criação de um projeto em Scratch requer primeiro que se pense sobre a ideia, em seguida, descubra como dividir o problema em etapas menores e como implementa-las usando os blocos de comando. O Scratch é projetado para ser "tinkerable": nele os estudantes podem alterar dinamicamente partes do código e ver imediatamente os resultados (por exemplo, dobrando um número para ver como ele muda um efeito gráfico);
- *habilidade de ser criativo e ter curiosidade intelectual* – o Scratch encoraja o pensamento criativo, uma habilidade cada vez mais importante no mundo de hoje em constante mudança. O Scratch envolve os jovens na busca de soluções inovadoras para problemas inesperados não só aprendendo a resolver um problema predefinido, mas estando preparado para chegar a novas soluções quando surgirem novos desafios.

As habilidades ligadas a área do relacionamento interpessoal e capacidade de se auto-direcionar são (RUSK, RESNIK e MALONEY, 2013):

- *habilidades interpessoais e colaborativas* - como os programas em Scratch são construídas com blocos gráficos, o código de programação é mais legível e de fácil compartilhamento. O suporte a objetos visuais e a códigos modulares apoia a colaboração, permitindo que os alunos trabalhem em conjunto em projetos e a troca de objetos e de código;
- *habilidade de se auto-direcionar* - ter uma ideia e descobrir como programá-lo em Scratch requer persistência e prática. Quando os jovens trabalhar em ideias de projetos que tenham um significado pessoal, eles têm uma motivação interna para superar os desafios e frustrações encontradas no processo de concepção e de resolução dos problemas;
- *habilidade de ser responsável e ser adaptável* - quando os alunos criam projetos Scratch, eles têm um público em mente, e precisam pensar em como os outros vão reagir e responder a seus projetos. Como os projetos Scratch são fáceis de mudar e rever, os alunos podem modificar seus projetos com base nos feedbacks recebidos;
- *habilidade de ser responsável socialmente* - como os programas Scratch são compartilháveis, os alunos podem usa-lo para provocar a discussão de questões importantes com outros membros de sua comunidade escolar, bem como com a comunidade Scratch de âmbito internacional.

5. CONSIDERAÇÕES FINAIS

Como mencionado anteriormente, para que um aplicativo seja considerado construcionista, deve ter características que facilitem as atividades de descrição, execução e depuração para a resolução de problemas (VALENTE, 1998).

A **fase de descrição** da resolução de um problema no Scratch é realizada através da junção de blocos gráficos, sem a necessidade de linguagens formais ou sinais de pontuação obscuros, fazendo com que seja de fácil aprendizagem e atendendo a um amplo público em uma faixa etária mais jovem. Ao descrever a solução de um problema longo o ambiente permite que a descrição

seja dividida em partes menores e testas de forma independente. Segundo Maloney et al. (2010) o ambiente gráfico Scratch incentiva a aprendizagem através da resolução dos problemas no qual o aluno coloca mão-a-obra em uma abordagem *bottom-up*.

O Scratch não apresenta distinção entre as fases de edição, compilação e execução o que facilita o acompanhamento da **fase de execução** e reflexão sobre o que esta sendo executado pelos blocos ou conjunto de blocos de comandos, inclusive é possível inclusive alterar parâmetros e incluir novos blocos de comandos nessa fase. Durante a execução dos seus scripts é possível acompanhar a sua execução, pois os blocos que estão sendo executados no momento ficam marcados com a borda externa em branco, o que permite saber o que, e por quanto tempo, cada bloco de comando está sendo executado.

O ambiente Scratch não possui mensagens de erros de sintaxe, pois os blocos de comandos se encaixam somente da forma que façam sentido, isso não significa que a resolução de problemas no ambiente não tenha erros. Então a **fase de depuração** no Scratch é facilitada por uma serie de recursos do ambiente como, por exemplo, a eliminação da fase de compilação que faz com que os usuários se envolvam mais com os testes e depuração da solução que passa a ser mais interativa melhorando assim os projetos propostos pelos alunos (MALONEY, et. al, 2010). Outro recurso é a capacidade de testar e depurar scripts individualmente e quando durante a execução de um script ocorra um erro o bloco de comando com erro fica rodeado de vermelho o que chama a atenção para o mesmo.

Todas essas características do Scratch fazem dele um ambiente no qual os alunos se sentem mais próximos da solução, pois mesmos que ele não esteja resolvendo o problema proposto, ele já esta fazendo algo o que diminui a frustração dos alunos e os leva a refletirem sobre o que está errado e repensarem sua solução para chegarem ao resultado esperado.

Os estudantes de hoje estão crescendo em um mundo que é muito diferente do mundo de seus pais e avós. Para ter sucesso na Sociedade Criativa de hoje, os alunos devem aprender a pensar de forma criativa, planejar sistematicamente, analisar criticamente, trabalhar de forma colaborativa, comunicar de forma clara, design de forma iterativa, e aprender continuamente. Infelizmente, a maioria dos usos de tecnologias em escolas de hoje não suportam essas habilidades de

aprendizagem do século 21. Em muitos casos, as novas tecnologias são simplesmente reforçar velhas formas de ensinar e aprender.

O Scratch foi projetado para dar suporte ao processo criativo dos alunos, como descrito pela “Espiral do Pensamento Criativo” proposta por Resnik (2007) é faz parte de uma nova geração de tecnologias projetadas para ajudar os alunos a se preparar para uma Sociedade Criativa. Mas eles são apenas o começo, sendo necessário repensar continuamente as abordagens pedagógicas utilizadas pelos professores e o uso de tecnologias educacionais. Assim como os alunos precisam se engajar na espiral pensamento criativo para se preparar para a Sociedade Criativa, os educadores devem fazer o mesmo.

REFERÊNCIAS

- FREIRE, P. **Pedagogia do Oprimido: saberes necessários à prática educativa**. 50ª ed. rev. São Paulo: Paz e Terra, 2011.
- LOVATE, E. P.; NOBRE, I. A. M. **A importância do uso de recursos computacionais do século XXI**. In.: Informática na Educação: um caminho de possibilidades e desafios. Serra, ES: Instituto Federal de Educação Ciência e Tecnologia do Espírito Santo, 2011, p.41-66.
- MALONEY, J., RESNICK M., RUSK N., SILVERMAN, B., EASTMOND E. **The Scratch Programming Language and Environment**. ACM Transactions on Computing Education, Vol. 10, No. 4, Nov. 2010.
- MIZUKAMI, M. G. N. **Ensino: as abordagens do processo**. [Reimpr.] São Paulo: E.P.U., 2011.
- MORAN, J. M. **Caminhos para a aprendizagem inovadora**. In.: Novas Tecnologias e Mediação Pedagógica, 15ª ed. SP: Papirus, 2009, p.22-24
- PIAGET, J. **Psicologia e Pedagogia**. Rio de Janeiro, Forense, 1970a.
- RESNICK, M., MALONEY, J., MONROY-HERNANDEZ, A., RUSK, N., EASTMOND, E., BRENNAN, K., MILLNER, A., ROSENBAUM, E., SILVER, J., SILVERMAN, B., AND KAFI, Y. **Scratch: Programming for All**. Communications of the ACM 52, 11, 60–67, 2009.
- RESNICK, Mitchel. **Sowing the Seeds for a More Creative Society**. Learning and Leading with Technology, 18-22, 2007.
- RUSK, N., RESNICK, M, MALONEY, J. **21st Century Learning Skills – Learning with scratch**. Lifelong Kindergarten Group MIT Media Laboratory. Disponível em: <http://ilk.media.mit.edu/projects/scratch/papers/Scratch-21stCenturySkills.pdf>. Acesso em: abr. 2013.
- SCRATCH. **ABOUT Scratch** (Scratch Documentation Site). Disponível em: http://info.scratch.mit.edu/About_Scratch>. Acesso em: maio 2013.
- VALENTE, J. A. **Computadores e Conhecimento: repensando a educação**. 2. ed. Campinas, SP: UNICAMP/NIED, 1998. 501. p.
- VYGOSTKY, L. S. **A formação social da mente**. 2ª. Ed. brasileira. São Paulo: Martins Fontes, 1988, 168 p.