

Programação de computadores com o Scratch: contando histórias construídas por alunos da escola de Hackers

Neuza Terezinha Oro (UPF)¹

Ariane Mileidi Pazinato (IMED)²

Amilton Rodrigo de Quadros Martins (IMED)³

Thaísa Leal da Silva (IMED)⁴

Resumo

Este artigo apresenta uma análise da programação envolvida em animações criadas por estudantes da Escola de Hackers, usando o software Scratch 2.0. A Escola de Hackers é um projeto interinstitucional entre a Secretaria Municipal de Passo Fundo, a Universidade de Passo Fundo, a Faculdade Meridional e o Instituto Federal Sul Rio Grandense, que oportunizam espaço para a aprendizagem de programação de computadores, desenvolvendo habilidades de autonomia e criatividade. Nesta análise procuramos evidenciar conhecimentos científicos aplicados na construção dos cenários, atores e códigos de programação de animações de histórias natalinas. As animações neste artigo foram elaboradas por alunos da escola municipal Notre Dame, como encerramento do ano letivo de 2014.

Palavras-chave: Programação; Scratch; Escola.

Abstract

This paper presents analysis of the programming involved in animation by students from the School of Hackers using Scratch 2.0 software. The School is an interagency project between the City Department of Passo Fundo, the University of Passo Fundo, the Faculdade Meridional and the Instituto Federal Sul Rio Grandense, which provides an opportunity space for learning computer programming, developing autonomy and creativity skills. This analysis seeks to show scientific knowledge applied in the construction of scenes, actors and programming code animations of Christmas stories. The animations in this article were prepared by students of the municipal school Notre Dame, in the 2014 activities year-end.

Keywords: Programming; Scratch; School.

¹ Contato: neuza@upf.br

² Contato: ariane.pazinato@imed.edu.br

³ Contato: amilton.martins@imed.edu.br

⁴ Contato: thaisa.silva@imed.edu.br

1. Contextualização

A Escola de Hackers (2014) é um projeto que consiste em um conjunto de ações que oportunizam a alunos do ensino fundamental de escolas públicas o aprendizado de técnicas de programação de computadores para crianças, utilizando o software Scratch, desenvolvido pelo MIT - Instituto de Tecnologia de Massachusetts. Este projeto é desenvolvido pela Secretária Municipal de Educação de Passo Fundo (SMEPF), Instituto Federal Sul Rio Grandense (IFSul), Faculdade Meridional (IMED) e Universidade de Passo Fundo (UPF), sendo gerenciado pelo Grupo de Estudos e Pesquisas em Inclusão Digital (GEPID), na qual fazemos parte, que se envolve com o planejamento e execução das atividades, tendo em vista o interesse deste em investigar temas relacionados com as tecnologias digitais aplicadas à escola básica.

Figura 1: Logo e Slogan do Scratch.



Fonte: <http://ict.gctaa.net/scratch4edu/>

A Escola de Hackers, além de ter foco na programação de computadores em escolas, promove publicações relevantes na área de Educação Tecnológica, e vários pesquisadores associados no tema da Educação e Tecnologia. Esse projeto já recebeu aporte da Fapergs - Fundação de Amparo à Pesquisa do Estado do Rio Grande do Sul com o edital PICMEL no valor total de R\$ 49.000,00, com objetivo de avaliar a viabilidade técnica do uso da Robótica Educacional como ferramenta de apoio à educação tecnológica, que usou como benchmark o Programa Líder da Lego.

Após 2 anos de execução do projeto, o mesmo já desenvolveu vários materiais livres, buscando publicar materiais e metodologias relevantes na área de Programação de Computadores para Crianças e Robótica Educacional, desenvolvendo conteúdo livre e

aberto para utilização em escolas públicas brasileiras e cumprindo seu papel de pesquisa aplicada na educação do nosso país.

No referido projeto, são realizadas atividades semanais nas escolas, na qual os estudantes da escola são orientados por monitores, que são estudantes de graduação em Ciências da Computação e Sistemas de Informação das IES parceiras, na realização de tarefas específicas com o intuito de possibilitar a aprendizagem dos recursos do Scratch, até adquirirem mais confiança e autonomia para seguirem por conta própria na programação. Depois disso, os alunos são desafiados a criarem jogos interativos sobre temas de seus interesses e, é neste momento, que a criatividade aflora e complexos algoritmos são criados.

Nesse viés, acreditamos que a programação de computadores se apresenta como uma ferramenta potencializadora da aprendizagem, quando possibilita a intensificação de relações multi ou interdisciplinares e o desenvolvimento do raciocínio lógico, da criatividade e da autonomia. Além disso, segundo Papert, a programação pode privilegiar a utilização de conhecimentos informais das crianças na construção de algoritmos, uma vez que o movimento de um objeto na tela do computador é programado pela criança a partir de sua noção de movimento na vida real (Papert, 2008).

Segundo John Dewey, filósofo e grande influenciador dos movimentos educacionais atuais, os alunos aprendem melhor realizando tarefas reais associadas aos conteúdos ensinados. Atividades manuais e criativas devem ganhar destaque no currículo, e as crianças precisam ser estimuladas a experimentar e pensar por si mesmas. Nesse contexto, a democracia ganha importância, por ser a ordem política que permite o desenvolvimento mais pleno dos indivíduos, no papel de decidir, em conjunto, o destino do grupo a que pertencem, sendo coautores da sua realidade (Dewey, 2010).

Dewey foi o maior difusor da corrente filosófica que ficou conhecida como pragmatismo, embora fosse denominada por ele de “instrumentalismo”, pois, em sua percepção, as ideias somente têm importância desde que sirvam de instrumento para a resolução de problemas reais. No campo específico da pedagogia, a teoria de Dewey é conhecida como “educação progressiva”, sendo um de seus principais objetivos educar a criança como um todo, valorizando o crescimento físico, emocional e intelectual em detrimento de conhecimentos puramente técnicos, adquiridos com base na repetição, que, de forma incerta, poderão servir para o seu futuro (Ramalho, 2015). Dewey defendia, também, a democracia não só no campo institucional, mas também no interior das escolas, onde o objetivo deveria ser ensinar a criança a viver no mundo, preparando-a para a vida ao

mesmo tempo em que vai vivendo. Com os problemas reais apresentados, o aprendizado vai sendo construído de forma natural e respeitando a individualidade de cada uma.

A Educação, na visão deweyana, é uma constante reconstrução da experiência, de forma a dar-lhe cada vez mais sentido e a habilitar as novas gerações a responder aos desafios da sociedade. Educar, portanto, é mais do que reproduzir conhecimentos, é incentivar o desejo de desenvolvimento contínuo, preparar pessoas para transformar algo (Ramalho, 2015).

Para Dewey, o professor deve apresentar os conteúdos escolares na forma de questões ou problemas e jamais antecipar as respostas ou soluções. Em lugar de começar com definições ou conceitos já elaborados, deve utilizar procedimentos que instiguem o aluno ao raciocínio e à elaboração de seus próprios conceitos e, por conseguinte, ao confronto com o conhecimento sistematizado (Dewey, 2010).

Na visão de Papert (2008), estudioso seguidor dos princípios experimentalistas de Dewey, ainda nos anos 1980, o computador já se mostrava uma máquina sem precedentes para materialização do *continuum experiencial* com a finalidade da construção de conceitos nas crianças. Conforme o teórico:

Os cidadãos do futuro precisam lidar com desafios, enfrentar um problema inesperado para o qual não há uma explicação preestabelecida. Precisamos adquirir habilidades necessárias para participar da construção do novo ou então nos resignarmos a uma vida de dependência. A verdadeira habilidade competitiva é a habilidade de aprender. Não devemos aprender a dar respostas certas ou erradas, temos de aprender a solucionar problemas (Papert, 2008).

Papert, em seu livro *A máquina das crianças: repensando a escola na era da informática*, aborda as mais diversas formas de utilização dos computadores pessoais na educação. Por ter vivido na época histórica da computação, mais especificamente na década de 1950, pôde presenciar a evolução dos computadores, desde a criação das primeiras máquinas informatizadas de grande porte e de acesso limitado a poucos até os dias atuais, com as máquinas portáteis, já presentes nas residências e na vida de muitas pessoas das mais diversas classes sociais (Papert, 2008).

Papert explica ainda, que o computador é “um dispositivo técnico aberto que estimula pelo menos alguns estudantes a impelir seu conhecimento até o limite para realçar o projeto através de uma ilimitada variedade de efeitos” (Papert, 2008, p. 66). Assim, aprender mais sobre técnicas de computação torna-se parte do aprendizado de uma forma que não ocorrerá com o papel e o lápis.

O computador pode ser usado na educação como máquina de ensinar ou como máquina para ser ensinada. Seu uso como máquina de ensinar consiste na informatização dos métodos de ensino tradicionais, sem mudança de abordagem pedagógica, no modelo que vem sendo utilizado. Do ponto de vista pedagógico, esse é o paradigma instrucionista.

Papert denominou de construcionista a abordagem pela qual o aprendiz constrói, por intermédio do computador, o seu próprio conhecimento e não é meramente “ensinado”. O computador deve ser utilizado de modo adequado e significativo, pois informática educativa nada tem a ver com aulas de computação. Nas palavras de Papert:

A questão central da mudança na Educação é a tensão entre a tecnicização e a não tecnicização, e aqui o professor ocupa a posição central. Desde a criação da máquina de imprimir não houve tão grande impulso no potencial para encorajar a aprendizagem tecnicizada. Há, porém, outro lado: paradoxalmente, a mesma tecnologia possui o potencial de destecnicizar a aprendizagem. Se isto ocorresse, eu contaria com uma mudança muito maior do que o surgimento, em cada carteira, de um computador programado para conduzir o estudante através de passos do mesmo velho currículo. Contudo, não é necessário sofismar sobre que mudança tem o maior alcance. O que é necessário é reconhecer que a grande questão no futuro da Educação é se a tecnologia fortalecerá ou subverterá a tecnicidade do que se tornou o modelo teórico e, numa grande extensão, a realidade da Escola (Papert, 2008, p. 55).

Na visão do construcionismo de Papert, existem duas ideias centrais que contribuem para esse tipo de construção do conhecimento: Primeiro, o aprendiz constrói alguma coisa, ou seja, é o aprendizado por meio do fazer. Segundo o aprendiz constrói algo do seu interesse e para o qual está bastante motivado e determinado – motivação intrínseca, por isso tende a potencializar o pensamento criativo. Em síntese, o envolvimento afetivo tende a tornar a aprendizagem mais significativa, rica e cheia de experiências.

Um dos grandes legados de Papert, enquanto ainda era pesquisador no MIT, buscando por em prática os princípios construcionistas, foi a linguagem Logo, que é uma linguagem de programação interpretada, voltada, principalmente, para crianças, jovens e até adultos. Foi utilizada, com grande sucesso, como ferramenta de apoio ao ensino regular e por aprendizes em programação de computadores. Ela implementa, em certos aspectos, a filosofia construtivista, segundo a interpretação de Papert e Wally Feurzeig.

A linguagem foi desenvolvida para permitir que crianças programassem a máquina, em vez de serem programadas por ela, criando seu próprio micromundo, que para Papert, é um universo simbólico criado pelas crianças em suas brincadeiras, onde tudo é possível, e sua imaginação e criatividade ditam as regras do mundo, inclusive físicas e sociais. Nesse espaço de criação, são executados processos ricos de autoria, assimilação e acomodação,

e as regras do micromundo são testadas pela própria criança para fortalecer e legitimar a sua realidade (Papert, 2008).

A linguagem Logo é de fácil compreensão e manipulação por crianças ou por pessoas leigas em computação e sem domínio em matemática, usada por centenas de escolas nos anos 1980, inclusive no Brasil. Embora tenha sido feita também para os leigos, o Logo envolve as linguagens de programação profissionais e parte, basicamente, da exploração de atividades espaciais, desenvolvendo conceitos numéricos e geométricos.

Pesquisador do MIT e seguidor da teoria do construcionismo de Papert, Mitchel Resnick busca aliar criatividade e aprendizagem, partindo do princípio que, na educação infantil, as crianças constroem brinquedos, instrumentos criativos que lhes possibilitam pensar, testar e aprender. Também, no ensino fundamental e médio, o caminho do aprender não deveria ser diferente. Embora os estudantes não sejam ouvintes passivos, permanecem inativos em sala de aula, quando o modelo educativo é baseado na transmissão de uma grande massa de dados, muitas vezes desconectada do interesse dos estudantes, que os sobrecarrega de dados e teorias (Resnick, 2006).

Mitchel Resnick foi graduado em Haverford High School na Pensilvânia, bacharel em física na Universidade de Princeton em 1978, PhD em ciência da computação pelo MIT em 1988. Trabalhou por cinco anos como jornalista de ciência para a revista Business Week, pesquisando amplamente sobre o uso de computadores na educação. Resnick foi premiado com um National Science Foundation Young Investigator Award, em 1993. Ele é co-editor da obra “Construcionismo na Prática: a concepção, Pensamento e Aprendizagem em um Mundo Digital” de 1996, e co-autor de “Aventuras em Modelagem: Explorando sistemas complexos e dinâmicos com StarLogo” em 2001.

Em suas pesquisas sobre aprendizagem, Resnick cita a lógica do “jardim da infância para toda a vida”. A ciência da computação deve desenvolver materiais para que as crianças possam crescer, construir soluções criativas que envolvam as várias áreas do conhecimento. Brinquedos vão adquirindo formas mais complexas, como pequenos robôs e engenhocas a serem montados e programados pelas crianças. Nesse processo, o computador torna-se um grande aliado, uma nova ferramenta para o aprendizado criativo (Resnick, 2006).

O propósito colocado pelo pesquisador é aprender por toda a vida, pesquisando, movimentando-se, deixando de lado as atividades estanques. Assim, criam-se atividades que se integram às diversas áreas do currículo, para se chegar a uma maior compreensão do assunto em questão. As crianças vão evoluindo no processo e descobrem soluções para problemas do cotidiano.

A tecnologia deve fazer parte da vida de crianças e jovens de modo que lhes faça sentido, e o mais importante é que eles podem mudar os sistemas que criam, fazendo isso de acordo com seus próprios interesses e necessidades, empregando as novas tecnologias na busca de soluções de seus problemas. O aprendizado, por meio de projetos e experiências, deve ultrapassar o sentido de sociedade de informação, evoluindo para o conceito de sociedade do conhecimento (Resnick, 2006).

Neste sentido, é fundamental evoluir para uma concepção de “sociedade criativa”, sendo preciso, para tanto, saber dar uso à informação. “As pessoas precisam continuar aprendendo a vida toda e dando soluções criativas para seus problemas e necessidades”, conclui o pesquisador.

A difusão de novas ideias na escola é fundamental, para que as novas metodologias e tecnologias reflitam em mudanças significativas na aprendizagem. O trabalho de Mitchel Resnick fala, justamente, na dificuldade em atingi-las, prevendo que as mudanças serão lentas, mas que devem ocorrer com maior facilidade ao longo das próximas gerações. “As crianças de hoje é que estarão melhor preparadas para as mudanças sistêmicas” (Resnick, 2006).

Vivemos o momento da inclusão digital maciça, de onde se originam estudos do potencial de novas tecnologias na educação. Por meio delas, pode-se melhorar e ampliar o aprendizado, usando-se músicas, esportes, laboratórios de ciências, ou, mesmo, observando-se a natureza. É preciso criar alternativas de uso das tecnologias de forma inovadora e produtiva, promover experiências criativas, abrindo portas para essas crianças às novas e infinitas possibilidades de aprender. Com isso, é possível operar a busca pelo desenvolvimento de um pensamento criativo sistêmico e intencional.

Baseado em todo o legado histórico de Papert, e conhecendo o trabalho de Mitchel Resnick, optamos por trabalhar com o ambiente de programação Scratch, criado por seu grupo de pesquisa no MIT, em virtude de sua interface dinâmica e intuitiva na qual os comandos estão visíveis ao usuário, na forma de blocos, bastando fazer a organização e sequenciamento destes. Como a programação em blocos é mais visual e menos textual, não exige que saibamos “escrever” os comandos na tela, pois estes já estão escritos. Também, segundo Resnick (2013), as crianças não só aprendem a programar, mas sobretudo, programam para aprender tanto conhecimentos de computação como de matemática, como estratégias de resolução de problemas e com isso, desenvolvimento da criatividade.

Papert (1985, 2008) e Resnick (2007) afirmam que na interação da criança com o computador, delinea-se um processo em espiral no qual a criança programa uma ideia

inicial – a hipótese, e testa esta hipótese no seu código. Assim, se algo não sai conforme previsto, a criança tem a possibilidade de refletir, propor novas hipóteses e testá-las novamente, em um processo cíclico que busca a melhoria contínua e intencional. Isto caracteriza o modelo de espiral, proposto pelos autores, na qual o “erro” surge como algo intrínseco ao processo e não algo ruim.

Ainda, Resnick e Rosenbaum (2013) afirmam que um projeto de construção criativa precisa partir do interesse do estudante e não de um conteúdo escolar pré-determinado. Os conteúdos específicos das áreas científicas podem ou não ser trabalhados dentro de um projeto com tais particularidades. Este é um fato importante a ser destacado, pois o professor, ao determinar que ensinará certos conteúdos de diversas áreas do conhecimento por meio do Scratch, deverá ter cuidado para não tornar a atividade de programação algo desinteressante e sem sentido para o estudante. É importante que eles queiram programar e que o projeto tenha relevância pessoal para eles, gerando com isso motivação intrínseca no estudante, ou seja, a motivação de querer fazer algo por vontade própria.

Sabemos que a programação de computadores utiliza conhecimentos de diversas áreas, no entanto, muitas vezes estes conhecimentos não aparecem da mesma forma como é trabalhado na escola. O estudante, ao programar computadores, aplica seus conhecimentos prévios, aprende e desenvolve outros conhecimentos necessários à programação, sem saber necessariamente que o está fazendo. Neste sentido, surge o interesse em analisar quais são estes conhecimentos “informais” que estão na programação e a quais conhecimentos científicos “formais” estão relacionados.

Para tanto, no tópico seguinte apresentaremos duas animações elaboradas por estudantes participantes do projeto com discussões a respeito dos conhecimentos científicos que foram possíveis de serem identificados nos códigos de programação.

2. Análise dos códigos de programação

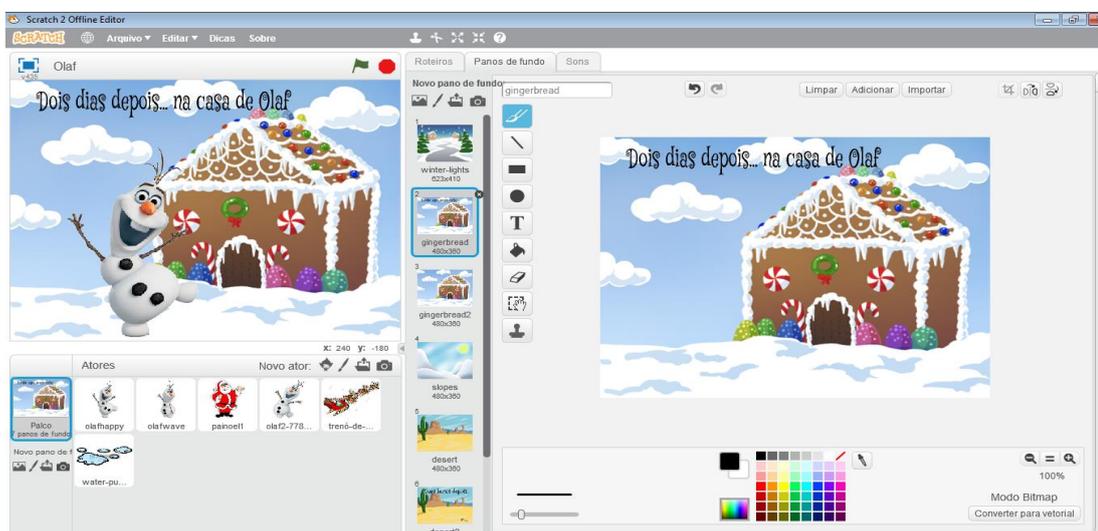
A Lei de Diretrizes e Bases da Educação, art. 26, diz que: “os currículos do ensino fundamental e médio devem ter uma base nacional comum a ser complementada, em cada sistema de ensino e estabelecimento escolar, por uma parte diversificada, exigida pelas características regionais e locais da sociedade, da cultura, da economia e da clientela”. O parágrafo primeiro do mesmo artigo especifica que os currículos “devem abranger, obrigatoriamente, o estudo da língua portuguesa e da matemática, o conhecimento do mundo físico e natural e da realidade social e política, especialmente do Brasil” (Brasil, 1996).

Com base neste propósito, neste artigo vamos analisar os códigos de programação da animação de Natal realizada por dois grupos de alunos com 10 a 11 anos, da Escola Municipal Notre Dame, Passo Fundo, nas oficinas semanais da Escola de Hackers. A atividade escolhida para análise envolveu sequenciamento e produção de histórias animadas, aliando conhecimentos de língua portuguesa, artes e de lógica matemática, além de habilidades de comunicação, liderança e gerenciamento de conflitos, inerentes de uma atividade em grupo. Os grupos eram compostos por dois alunos, que serão denominados Grupo A e Grupo B. Cada grupo elaborou a sua história, desde a escolha dos personagens, do enredo, dos cenários e dos diálogos até a finalização na programação da animação.

O grupo A elaborou a história “O pedido que o boneco de neve Olaf fez ao Papai Noel: conhecer um lugar ensolarado”, pois segundo os alunos, o boneco estava cansado de viver na neve. Ele foi atendido pelo Papai Noel, porém o seu pedido não deu muito certo, pois Olaf acabou virando uma poça d’água.

Nesta animação o Grupo escolheu os seguintes atores: três bonecos, representado o boneco de neve Olaf em diferentes posições; o Papai Noel; o trenó e a poça d’água. Esses personagens e os cenários estão mostrados na Figura 2. O grupo optou por criar três atores para o personagem Olaf, ao invés de criar um único ator com três trajes diferentes para ele.

Figura 2: Atores e Cenários da animação do Grupo A



Fonte: Autores

Os roteiros para cada um dos personagens que compõem a animação estão apresentados nas Figuras de 3 a 5. Na Figura 3 são apresentadas as programações realizadas pelo grupo para os atores poça d’água e trenó.

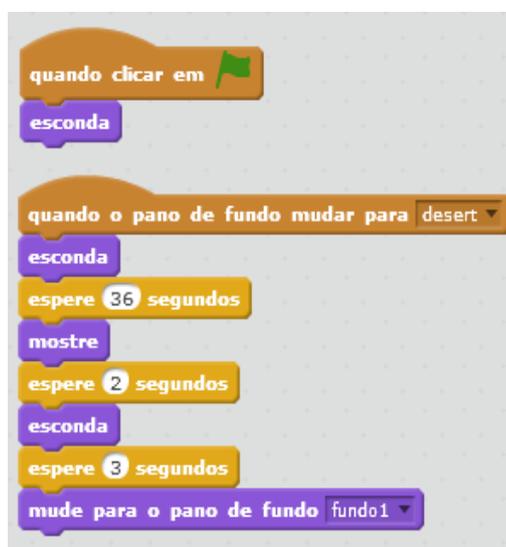
Na Figura 3 (a), o Grupo A utilizou uma programação simples, com comandos de troca de pano de fundo, escondendo e mostrando o ator “poça d’água” conforme o tempo indicado na figura pelo comando “espere __segundos”.

Já na Figura 3 (b), na programação do ator “trenó”, além da troca de pano de fundo, e dos comandos: “espere”, “mostre” e “esconda”, são utilizados comandos de movimento, como o comando “vá” no qual é necessário indicar as coordenadas x e y da tela do Scratch para indicar para onde se deseja que o ator se mova.

Além disso, na programação do trenó um outro comando de movimentação muito útil é utilizado, o comando “deslize por __segundos”, no qual se define por quantos segundos o ator deverá se mover, informando também as coordenadas x e y para onde se deseja que o ator se mova (deslize).

Ao invés de utilizar o comando “deslize por __segundos”, o Grupo A poderia ter usado o comando “vá” em conjunto com o comando “mova __passos”, assim o ator iria da mesma forma de um ponto a outra da tela só que se movendo através do comando de contagem de passos. No entanto, o comando “deslize por __segundos”, escolhido pelo grupo, possui um efeito mais adequado à proposta de um trenó deslizando (voando), durante a execução do programa.

Figura 3: Programação da poça d’água e do trenó.



a) Poça d’água



(b) Trenó

Fonte: Autores

A Figura 4 apresenta a programação dos atores que representam o Boneco de Neve Olaf. E, a programação do Papai Noel está mostrada na Figura 5.

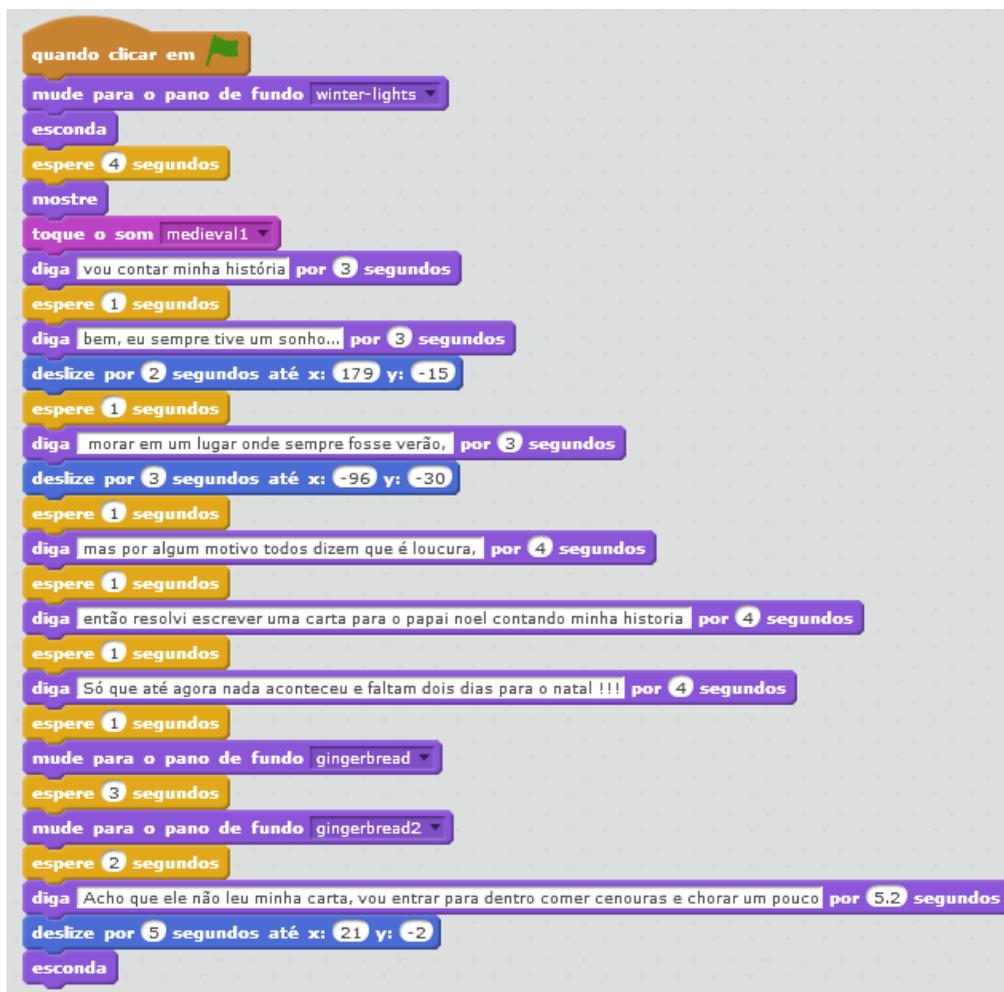
A programação apresentada na Figura 4 (a) indica que ao clicarmos na bandeira verde para executar a animação, o pano de fundo é alterado para “winter lights”, um som é ouvido e o ator Olaf 1 se apresenta dizendo uma mensagem de “Olá” aos usuários. Na Figura 4 (b), ao clicarmos na bandeira verde, o pano de fundo selecionado mais uma vez é o “winter lights”, um novo som, chamado “medieval 1” é tocado, e o ator Olaf 2 começa a contar uma história, na qual diz que sempre teve o sonho de morar em um lugar que seja sempre verão, e diz que escreveu uma carta ao Papai Noel fazendo esse pedido, mas que faltando dois dias para o Natal nada aconteceu, e Olaf acha que o Papai Noel não leu sua carta.

Com isso, o pano de fundo é alterado novamente e o ator Olaf 2 “desliza” como se estivesse entrando para sua casa e diz que vai comer algumas cenouras e chorar um pouco. Na Figura 4 (c), o ator Olaf 3 comemora dizendo que sabia que seu sonho iria realizar, desliza de um lado do pano de fundo a outro e logo depois o pano de fundo é alterado para “desert2”.

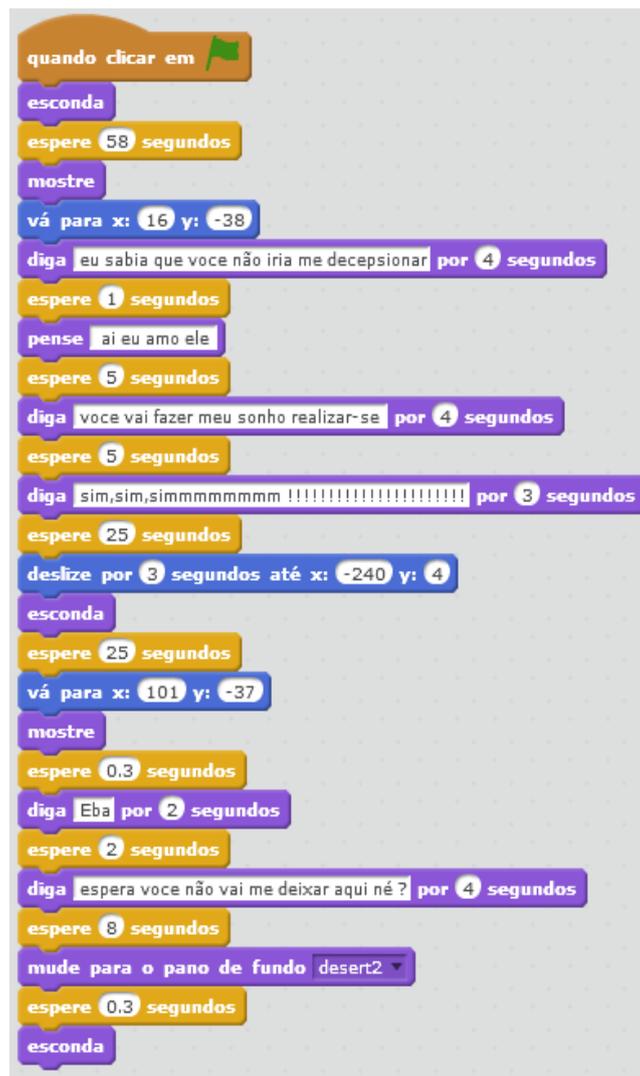
Figura 4: Programação dos atores que representam o Boneco de Neve



(a) Olaf 1



(b) Olaf 2



(c) Olaf 3

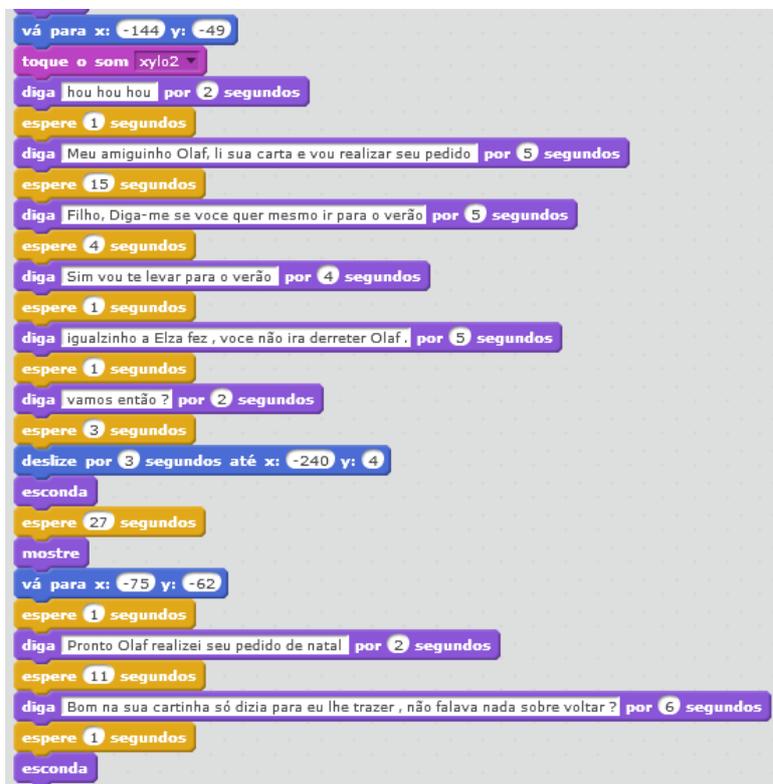
Fonte: Autores

Na Figura 5, são utilizados dois novos comandos em relação aos comandos usados nos atores “poça d’água” e “trenó”, são eles: o comando de Som “toque o som___”, no qual o som utilizado pode ser do próprio Scratch ou um arquivo de som importado para o Scratch: e o comando “diga___por___segundos”, o qual faz com que uma mensagem apareça em um balão de fala acima do ator correspondente por um determinado tempo, indicado em segundos.

O grupo também poderia ter usado o comando “diga___”, em conjunto com o comando “espere___segundos” o que teria o mesmo efeito na programação do que o comando “diga___por___segundos”. Além do comando “diga”, os alunos poderiam fazer uso do comando “pense___” ou do comando “pense___por___segundos” nos quais uma

mensagem apareceria em um balão de pensamento (com formato de nuvem) acima dos atores que estivessem sendo programados.

Figura 5: Programação do Papai Noel



Fonte: Autores

Na Figura 5, a programação do ator Papai Noel é feita de modo que ao clicar na bandeira verde, um novo som é tocado, e o Papai Noel aparece dizendo “Hou hou hou”. Logo após, o Papai Noel diz a Olaf que leu a sua carta e que vai levá-lo para o verão, dizendo a Olaf que ele não irá derreter. O Papai Noel leva Olaf até o verão (deslizando pelo pano de fundo através do uso de coordenadas da tela) e depois desaparece, afirmando que na carta de Olaf apenas era feito o pedido para levá-lo até o verão, mas não falava nada sobre voltar. E a história é concluída assim.

O outro grupo, Grupo B, elaborou o seguinte roteiro para a programação de sua animação de natal: “O ajudante do Papai Noel que se perdeu na entrega dos presentes às crianças”. O narrador da história solicita que o espectador ajude-o a encontrar o Papai Noel, o que não acontece e ele volta para o Pólo Norte.

Os personagens representados pelos atores: Natal, boneco de neve e morcego, bem como os cenários escolhidos para animação estão mostrados na Figura 6.

Figura 6: Cenários e atores da animação de natal do Grupo B.



Fonte: Autores

Para fazer a animação do enredo da história de natal, o Grupo B construiu a programação de cada ator, conforme ilustrado nas Figuras 7 e 8.

Figura 7: Programação dos atores Natal e Morcego.

```

quando clicar em [bandeira]
  esconda

quando o pano de fundo mudar para winter-lights
  espere 15,2 seg
  vá para x: -110 y: 99
  mostre
  espere 0,2 seg
  diga HO por 2 segundos
  deslize por 5 seg até x: -70 y: 108
  adicione -10 ao tamanho
  diga HO por 2 segundos
  deslize por 1 seg até x: 72 y: 122
  adicione -10 ao tamanho
  diga HO por 2 segundos
  esconda
  adicione 20 ao tamanho
    
```

(a) Natal

```

quando clicar em [bandeira]
  esconda

quando o pano de fundo mudar para winter-lights
  esconda

quando o pano de fundo mudar para iMAAAAGEEEEEM
  espere 10,2 seg
  mostre
  vá para x: 240 y: 180
  deslize por 10 seg até x: 28 y: 108
  mude para a fantasia bat1-b

quando o pano de fundo mudar para Casa-de-Santa-Claus-Polo-Norte
  esconda
    
```

(b) Morcego

Fonte: Autores

Na Figura 7 (a) é apresentada a programação do ator Natal, o qual é um Papai Noel em seu trenó puxado por renas, conforme apresentado na Figura 6. Na programação do ator Natal, o Papai Noel aparece dizendo “HO HO HO”, e deslizando pela tela, conforme as coordenadas apresentadas na Figura 7 (a).

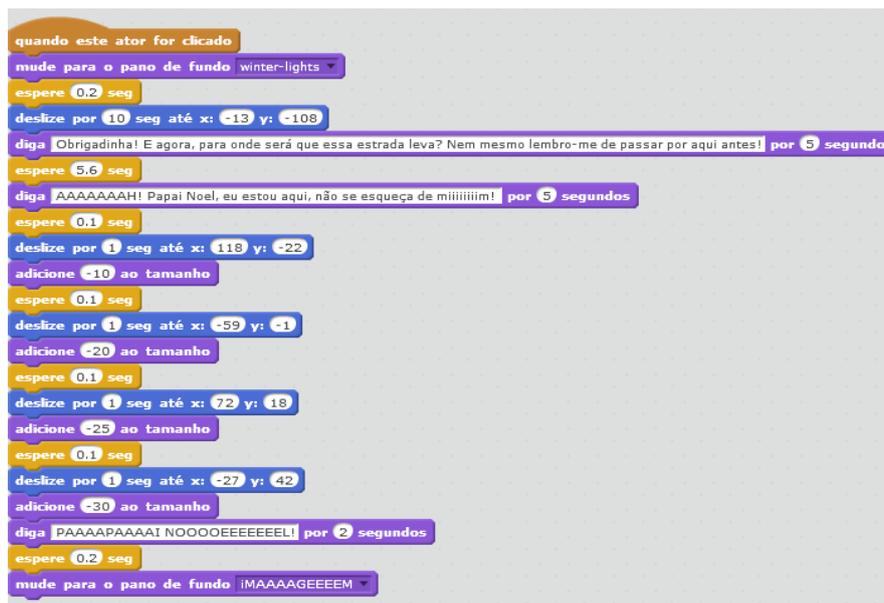
Na Figura 7 (b), é feita a programação do ator Morcego, na qual quando o pano de fundo é clicado ele é alterado de “winter lights” para “IMAGEMMM”, e a imagem indicada como pano de fundo na Figura 6 é apresentada. Logo após a fantasia do ator Morcego é alterada para “bat1-b”, e novamente quando o pano de fundo for clicado, ele é alterado para “Casa de Santa Claus Pólo Norte”.

Na Figura 7 (a) e Figura 7 (b) podemos verificar que além do Evento “quando clicar na bandeira verde”, que possui a função de acionar a execução do código programado, também foi utilizado o Evento “quando o pano de fundo mudar para_____”, o qual dispara uma ou mais ações, toda a vez que determinado pano de fundo for trocado durante a execução do programa.

Além disso, em relação aos comandos utilizados na programação apresentada pelo Grupo A, o Grupo B acrescentou dois comandos principais, o comando “adicione___ao tamanho”, que tem a função de aumentar o tamanho do autor conforme o número indicado; e comando “mude para a fantasia_____” que faz com que a aparência do ator seja alterada quando o comando é executado.

Figura 8: Programação do ator Boneco de Neve





Fonte: Autores

Na Figura 8, é apresentada a programação do ator Boneco de Neve, na qual é possível verificar que quando a bandeira verde é clicada o pano de fundo é alterado para “imagens natalinas”, o Boneco de Neve pergunta aos usuários se eles viram o Papai Noel, dizendo que é ajudante dele e que se perdeu enquanto entregava presentes. Ele pede a ajuda dos usuários, dizendo para clicar nele, com o intuito de ajudá-lo a achar o Papai Noel.

A seguir, na programação, é possível notar que quando o ator do Boneco de Neve é clicado, o pano de fundo é alterado novamente para “winter lights”, o Boneco agradece ao usuário por levá-lo até aquele novo cenário, mas diz que não se lembra de ter estado ali. Ele fica chamando pelo Papai Noel, deslizando pela tela e reduzindo de tamanho, como se estivesse se distanciando do usuário. Por fim, o pano de fundo é alterado para o cenário chamado de “IMAGEMMM”, conforme indicado na Figura 6.

A partir dos projetos de construção de uma história de Natal elaborados pelos dois grupos escolhidos para este artigo, percebemos que ambos se utilizaram do bloco “espere ___ seg” encontrado no Script “Comando” do *Scratch 2.0* para sincronizar os diálogos entre os atores.

Cabe salientar que a mesma sincronização poderia ser realizada pelos blocos enviar e receber mensagens, que se encontra no script “Eventos” da mesma versão do *Scratch*. Esta não foi utilizada porque na Escola Notre Dame a versão 2.0 havia sido instalada há poucos dias do término das atividades da Escola de Hackers para aquele ano e os alunos optaram em finalizar suas histórias com os conhecimentos de programação no *Scratch* que já haviam adquirido.

A transposição dos conhecimentos de sala de aula, em específico de Redação, Artes e Matemática, foi realizada no momento que os alunos organizaram as programações dos atores a partir das redações das histórias por eles elaboradas, construíram cenários que auxiliaram na sequência da história - início, meio e fim e elaboraram sequenciamentos lógicos para a programação e sincronização dos diálogos entre atores e atores e cenário.

A programação permite concretizar conceitos de diversas áreas do conhecimento na idealização e concretização de projetos multidisciplinares. Neste caso, usando o *Scratch* para construção de histórias ficou perceptível a relação entre as áreas de Matemática e Linguagens.

3. Considerações finais

Neste artigo foram apresentados projetos de dois grupos de alunos da Escola Notre Dame. Em seus projetos os grupos deveriam construir uma história de Natal, utilizando programação no ambiente *Scratch 2.0*. Ambos os grupos demonstraram grande criatividade e criaram histórias bastante distintas. A complexidade da programação envolvida foi semelhante nos dois projetos, desde que os grupos usaram principalmente comandos encontrados nos Scripts: Movimento (“vá” e “deslize”), Aparência (“diga”, “mostre”, “esconda”), Controle (“espere”). No entanto, alguns comandos foram usados especificamente por cada grupo de acordo com sua criatividade e imaginação, como, por exemplo, o comando “toque o som” usado pelo Grupo A; e o comando “adicione ao tamanho” utilizado pelo Grupo B.

A Escola de Hackers apresenta-se como uma possibilidade para os estudantes de Escolas Municipais de Passo Fundo aprenderem programação e conseqüentemente, desenvolverem atividades inter ou multidisciplinares, conforme relatado neste artigo. Segundo Resnick (2013):

No processo de aprender a programação, as pessoas aprendem muitas outras coisas. Eles não estão apenas aprendendo uma programação, eles estão programando para aprender. Além de aprender noções matemáticas e computacionais (como variáveis e condições), eles também estão aprendendo estratégias para a resolução de problemas, elaboração de projetos e comunicar idéias. Essas habilidades são úteis não apenas para os cientistas da computação, mas para todos, independentemente da idade, interesses ou ocupação” (Resnick, 2013, p. s/n, Tradução dos autores).

Neste contexto, utilizando-se da programação de computadores como ferramenta educacional, o projeto viabilizou muitas aprendizagens informais e formais da escola, mediante um trabalho dinâmico realizado nos laboratórios de informática por cada grupo.

Além disso, a programação de computadores pode permitir que os estudantes deixem de ser somente usuários de tecnologias e passam a ser produtores de conhecimento pela ação de programar. Tornam-se cidadãos com perspectivas para um futuro inovador.

Para tanto, o software de programação *Scratch*, usado nas oficinas, oferece a possibilidade para que o aluno programe por meio de blocos coloridos, sem a necessidade de lidar com códigos e linguagem de programação de difícil compreensão e em outra língua. Dessa forma, rapidamente aprendem a lidar com os recursos do software. Assim, Sobreira, Takinami e Santos (2013) destaca o potencial pedagógico e significativo do *Scratch* ao verificar que o contato com múltiplas linguagens favorece o desenvolvimento crítico na análise de mídias pelos alunos e que o incentivo ao pensamento criativo e a curiosidade na busca de soluções inovadoras para problemas inesperados exige, além do conhecimento, postura reflexiva, autônoma, crítica e colaborativa.

Estas potencialidades do uso do *Scratch* levam os autores a afirmar que desenvolver atividades com esta linguagem contextualizada ao currículo escolar possibilita o desenvolvimento de competências e habilidades necessárias à formação do cidadão no século XXI. Esta autonomia para o aluno aprender, criar e inovar pode ser observada no desenvolvimento das histórias descritas nesse artigo, sendo que esta foi uma das atividades desenvolvidas na Escola de Hackers.

Considerando as relações entre o pensamento computacional e as disciplinas já presentes no currículo escolar, Barcelos (2014) confirma que existem benefícios no desenvolvimento de estratégias didáticas conjuntas, na medida em que é necessária à mobilização de conceitos ao programar com *Scratch*.

Neste contexto, Valente (1999) salienta que ao compreender uma nova linguagem os alunos dispõem de um recurso que, ao ser utilizado na realização de tarefas que podem envolver desde a criação de figuras e textos até a movimentação de personagens, permitindo-lhes que construam seus conhecimentos através da resolução de situações problemas para a qual necessitam explicitar seu raciocínio, em termos de uma linguagem precisa e formal.

O Projeto Escola de Hackers, que vem sendo realizado desde 2014, e possibilitou a concretização de estudos sobre programação de computadores realizados no GEPID, mobilizando estudantes em TCCs de Graduação, Dissertações de Mestrado e duas Teses de Doutorado do PPGEdU da UPF, pois em dois anos de projeto já participaram em torno de 600 alunos, 30 professores, de 25 escolas da rede municipal de ensino de Passo Fundo e Região.

A comunidade escolar participante do Projeto avaliou de forma positiva as ações desenvolvidas e percebeu significativas mudanças na aprendizagem, nas atitudes e na criatividade e desenvoltura dos estudantes que participaram do projeto por meio da liberdade de aprender criando.

4. Referências

BARCELOS, T. S. (2014). **Relações entre pensamento computacional e a matemática em atividades didáticas de construção de jogos digitais**. 2014. Disponível em <http://goo.gl/IHGQd3>. Acesso em 23 nov. 2015.

BRASIL **Senado Federal**. Lei de Diretrizes e Bases da Educação Nacional: nº 9394/96. Brasília, 1996.

DEWEY, J. **Experiência e educação**: textos fundantes de educação. Rio de Janeiro: Vozes, 2010.

PAPERT, S. **Logo**: computadores e educação. Tradução de José A. Valente, Beatriz Bitelman e Afira V. Ripper. São Paulo: Editora Brasiliense, 1985.

PAPERT, S. **A máquina das crianças**: repensando a escola na era da informática. Ed.rev. Porto Alegre: Artmed, 2008.

RAMALHO, P. **John Dewey**: educar para crescer. Disponível em: <<http://educarparacrescer.abril.com.br/aprendizagem/john-dewey-307892.shtml>>. Acesso em: 19 out. 2015.

RESNICK, M. O computador como pincel. **VEJA**. Limpeza de Alto Risco. Especial: um guia do mundo digital, São Paulo: Abril Cultural, (41), 2006.

RESNICK, M. All I Really Need to Know (About Creative Thinking) I Learned (By Studying How Children Learn) in Kindergarten. **Anais**, ACM Creativity e Cognition conference, Washington DC, June, 2007.

RESNICK, M. **Learn to code, code to learn**. MIT Media Lab. 2013. Artigo on-line. Disponível em: < <http://web.media.mit.edu/~mres/papers/L2CC2L-handout.pdf>>. Acesso em: 18 nov. 2015.

RESNICK, M.; ROSENBAUM, E. Designing for Tinkerability. In: HONEY, M.; KANTER, D. (eds.), **Design, Make, Play**: Growing the Next Generation of STEM Innovators, pp. 163-181. Londres: Routledge, 2013.

SOBREIRA, E. S. R.; TAKINAMI O. K.; SANTOS, V. G. Programando, criando e inovando com o Scratch: em busca da formação do cidadão do século XXI. **Anais**, Jornada de Atualização em Informática na Educação, 2, 126-152, 2013.

VALENTE, J. A. Análise dos diferentes tipos de softwares usados na Educação. In: VALENTE, J. A. (Org.) **O computador na sociedade do conhecimento**. Campinas, SP: NIED, UNICAMP, p. 71-85, 1999.