



Programação de Computadores e Matemática: potencializando a aprendizagem

Neuza Terezinha **Oro**
Universidade de Passo Fundo
Brasil

neuza@upf.br

Ariane Mileidi **Pazinato**
Universidade de Passo Fundo
Brasil

arianepazinato@upf.br

Valéria Espíndola **Lessa**
Instituto Federal do Rio Grande do Sul (IFRS), câmpus Erechim
Brasil

valeria.lessa@erechim.ifrs.edu.br

Franciele Meinerz **Forigo**,
Instituto Federal de Educação, Ciência e Tecnologia Farroupilha, câmpus Santa Rosa
Brasil

francieleforigo@gmail.com

Resumo

Este trabalho apresenta o projeto Escola de Hackers, em sua perspectiva metodológica, bem como uma discussão a respeito dos conceitos e habilidades matemáticas presentes na transição da linguagem narrativa para a linguagem de programação. A Escola de Hackers é um projeto interinstitucional, coordenado pela Universidade de Passo Fundo (UPF- Brasil), que oportuniza espaço para desenvolvimento de conhecimentos na área de programação de computadores e de matemática para estudantes do Ensino Fundamental da rede pública de Passo Fundo. Para tanto, utiliza-se o *software* Scratch, pois acredita-se que sua linguagem de programação no formato de blocos facilita a compreensão da lógica algorítmica pelos estudantes.

Palavras Chave: Programação, Matemática, Scratch, Aprendizagem.

Introdução

A programação de computadores pode vir a intensificar relações interdisciplinares da computação com outras áreas do conhecimento, em especial a matemática, e também a desenvolver o raciocínio lógico e a criatividade. Nesta perspectiva, a Escola de Hackers é um projeto que consiste em um conjunto de ações que oportunizam a alunos do ensino fundamental o aprendizado de técnicas de programação, utilizando o *software* Scratch (<http://scratch.mit.edu>). O projeto é desenvolvido por Instituições de Ensino Superior da cidade de Passo Fundo sendo elas: Universidade de Passo Fundo (UPF), Instituto Federal Sul Rio Grandense (IFSul) e Faculdade Meridional (IMED), com apoio da Secretária Municipal de Educação de Passo Fundo (SMEPF), Brasil.

Na Escola de Hackers participam alunos de 6º ao 9º ano do Ensino Fundamental com idades entre 11 e 14 anos. As atividades na Escola de Hackers são realizadas em oficinas semanais por um monitor orientado pela equipe executora do projeto. Estas atividades procuram desafiar os estudantes a transpor a linguagem narrativa de um problema em linguagem de programação, na qual esse elo está permeado de conhecimento implícito de matemática.

Além disso, a programação tende a agir de forma intermediária entre o problema narrativo e a formalização matemática, “podendo promover uma transição mais ‘suave’ para compreensão da linguagem matemática” (Barcelos e Silveira, 2012).

Neste sentido, apresentaremos, nesta comunicação, o projeto Escola de Hackers sob o viés metodológico, bem como uma discussão a respeito dos conhecimentos matemáticos informais presentes nos códigos de programação do *software* Scratch, que possibilitam esta transição "suave" da linguagem narrativa dos problemas que queremos resolver, para o entendimento da matemática formal da escola.

Escola de Hackers

A Escola de Hackers foi idealizada por meio de diversas ações que aconteciam na UPF em 2013, mais especificamente no Grupo de Pesquisa em Inclusão Digital (GEPID - <http://gepid.upf.br>). A ação responsável por esta promoção foi a Olimpíada de Programação de Computadores para Estudantes de Ensino Fundamental, que já viabilizava a utilização do *software* Scratch pelas equipes participantes e, a partir de seu destaque na região, encontra-se em sua terceira edição.

A partir disso, no início de 2014, iniciou-se as atividades da Escola de Hackers, cujo principal objetivo é que os estudantes destas escolas aprendam programação e realizem projetos no *software* Scratch com o auxílio de monitores, durante todo o ano letivo de 2014. Este é um projeto piloto, cuja pretensão é dar continuidade nos próximos anos. Dessa forma, cada instituição envolvida - UPF, IFSul, IMED e SMEPF - ficou encarregada de realizar atividades específicas do projeto nas 21 escolas municipais da cidade de Passo Fundo que participam.

O projeto da Escola de Hackers está organizado em quatro etapas: preliminar, execução, formatura e avaliação. A etapa preliminar consiste na definição dos conteúdos de programação, elaboração de material didático pedagógico; contato com as escolas municipais; realização das inscrições; e formação de monitores, que são acadêmicos dos cursos de Matemática e Ciências da Computação da UPF, bem como a formação dos professores das Escolas do município, realizadas pelo IFSul.

A etapa de execução se refere à implementação das ações da Escola de Hackers. Dentre as ações desenvolvidas pela Escola destacam-se a realização de oficinas semanais com duração de 2 horas, conduzidas pelos monitores, nos laboratórios de informática das escolas inscritas. Nessa etapa, está se utilizando o material elaborado na etapa preliminar. Estas ações são acompanhadas por professores da UPF e da IMED e, em dezembro de 2014, será realizada a formatura com a entrega de certificados aos participantes.

De forma a avaliarmos o desenvolvimento do Projeto, estão sendo construídos relatórios mensais, a partir de reuniões e seminários realizados. Após a finalização deste projeto piloto, o desdobramento de dará em dois momentos: a continuidade das oficinas nas escolas para novos estudantes, com o auxílio daqueles que já participaram do projeto, e, a realização de oficinas avançadas de programação, na UPF, para estudantes destaques desta primeira experiência, envolvendo a robótica educacional.

O conteúdo das oficinas foi elaborado em formato de módulos. No primeiro módulo, trabalhou-se a interface do *software* Scratch, com atividades que envolveram a troca de palco e de *sprites*¹, cores, sons, trajes ou fantasias, entre outras. No segundo módulo (o projeto encontra-se nesta etapa), está sendo trabalhado a programação, utilizando atividades que envolvem todos os comandos ou roteiros, inclusive as noções de variáveis, condicionais e listas. A partir das instrumentalizações obtidas no que já foi feito, no módulo 3 serão propostas atividades na qual os estudantes podem construir projetos de jogos, animações e histórias. Além desses módulos, também estamos elaborando um Caderno Didático, no qual serão apresentadas propostas pedagógicas para que possam ser aplicadas ou adequadas às futuras oficinas de programação.

Educação, Matemática e Programação de Computadores

Segundo Papert (2008), as crianças possuem, antes de ir para a escola, estilos orais de aprender e quando chegam lá precisam desenvolver estilos "leterados"² de aprender. Para o autor, esta transição é menos aplicável à matemática, uma vez que o nome "matemática" é reservado apenas para o tipo de matemática "leterada" da escola e exclui uma base de conhecimentos "orais" que poderiam ser usados de alicerce para a matemática formal. Nesse sentido, a programação pode privilegiar a utilização destes conhecimentos informais (orais) na construção de algoritmos necessários para resolução dos problemas, em especial da Escola de Hackers.

Acreditamos que esse processo sequencial utilizado na elaboração de algoritmos contribua para a compreensão de conceitos, teoremas, conteúdos, etc., da matemática numa situação formal de ensino desta ciência. A partir disso, compreendemos que é possível criarmos estratégias inovadoras dentro do Sistema Escolar privilegiando processos de aprendizagem pela invenção e criação. Papert e Freire convergem neste sentido, na qual Papert diz que para Piaget "entender é inventar. Ele (Piaget) estava pensando nas crianças. O princípio, porém, aplica-se a todos nós." (Papert, 2008). Ainda, Freire (2005) afirma que

“Fora da busca, fora da práxis, os homens não podem ser. Educadores e educandos se arquivam na medida em que, nesta distorcida visão da educação, não há criatividade, não há transformação, não há saber. Só existe saber na invenção, na reinvenção, na busca

¹ Sprite: objeto a ser programado no software Scratch.

² Papert (2008) utiliza a palavra "leterado" para se referir "à habilidade de ler palavras formadas com as letras do alfabeto".

inquieta, impaciente, permanente, que os homens fazem do mundo, com o mundo e com os outros” (2005, p.66).

Na busca de uma educação mais criativa, a Escola de Hackers utiliza o *software* de programação Scratch, elaborado no Massachusetts Institute of Tecnology (MIT - <http://web.mit.edu/>) por Mitchel Resnick³. O Scratch é um ambiente gráfico de programação que possibilita o desenvolvimento de aplicativos que integram recursos de multimídia de forma intuitiva. Tem por objetivo facilitar a introdução de conceitos de matemática e de computação, bem como oportunizar o pensamento criativo, o raciocínio e o trabalho colaborativo. Segundo informações que constam no portal EduScratch (<http://eduscratch.dge.mec.pt/>), a ferramenta Scratch foi concebida e desenvolvida como resposta ao problema do crescente distanciamento entre a evolução tecnológica no mundo e a fluência tecnológica dos cidadãos.

O termo “Scratch” provém da técnica de *scratching* utilizada pelos disco-jockeys do hip-hop, que giram os discos de vinil com as suas mãos para frente e para trás de modo a fazer misturas musicais originais. Com o Scratch, é possível fazer algo semelhante, misturar diferentes tipos de clipes de mídia de modo criativo e usar uma programação matemática similar à feita nos programas de computadores reais, conforme mostra a Figura 1, porém de forma lúdica, simples e intuitiva (EduScratch, 2014).

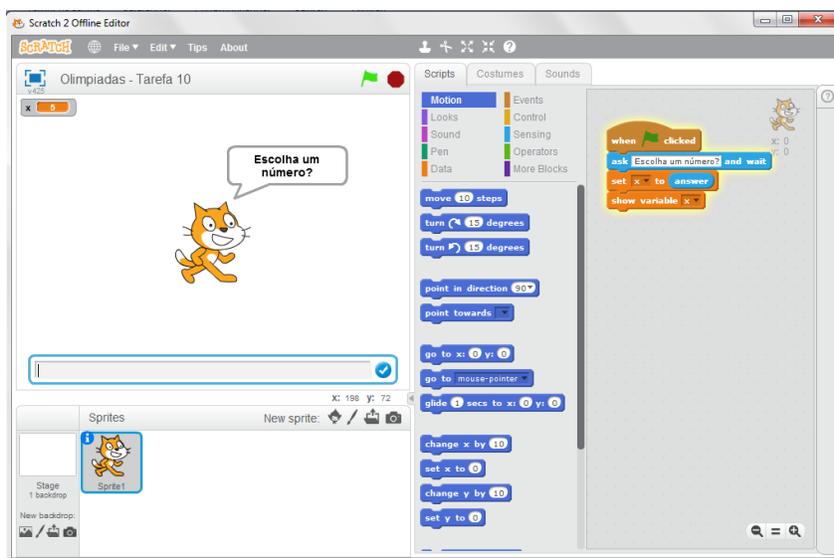


Figura 1. Tela do Scratch 2.0.

Entre os recursos do Scratch, destacam-se as competências para a resolução de problemas e para a concepção de projetos com raciocínio lógico, decomposição de problemas complexos em partes mais simples, identificação e eliminação de erros, desenvolvimento de ideias, desde a compreensão até a concretização do projeto, concentração e perseverança (Marques, 2012).

O Scratch é voltado, principalmente, ao usuário infantil e jovem, mas já com adeptos da idade adulta. Oferece uma linguagem de programação simples, em que é possível criar projetos

³Dirigi o grupo Lifelong Kindergarten no Media Laboratory do Instituto de Tecnologia de Massachusetts na qual desenvolve novas tecnologias para envolver as pessoas (especialmente crianças) em experiências de aprendizagens criativas. Trabalhou com Seymour Papert. Recebeu o Prêmio McGraw em Educação para 2011. (Fonte: <http://web.media.mit.edu/~mres/>)

que auxiliam na aprendizagem e no desenvolvimento de habilidades matemáticas e computacionais, de modo a complementar e enriquecer o pensamento criativo e aprender a trabalhar de maneira colaborativa. Segundo Resnick (2013).

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

Vemos que na interação da criança com o computador, delineia-se um processo em espiral no qual a criança programa uma ideia inicial (hipótese) e testa. Assim, se algo não sai conforme previsto, a criança tem a possibilidade de refletir, propor novas hipóteses e testá-las novamente. Resnick (2007) propõe um modelo de espiral que apresenta o processo sequencial de criação do estudante. Tal modelo inicia com “Imagine”, imaginar o que quer fazer, seguido por “Create”, criar o projeto a partir de suas ideias, “Play”, divertir-se com suas criações, “Share”, compartilhar suas ideias com outros e “Reflect”, refletir sobre suas experiências para depois reiniciar o processo, conforme ilustra a Figura 2.

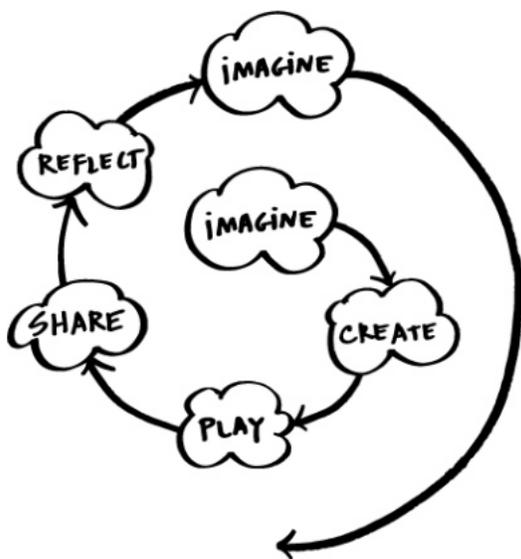


Figura 2. Modelo de Espiral de Resnick(2007).

Dessa forma, antes de iniciar o processo em espiral, o estudante necessita do apoio do professor para apropriar-se dos recursos básicos do programa e conhecer alguns exemplos de jogos e criações já prontas para que, a partir daí, consiga imaginar o que deseja fazer. Na criação, o estudante utilizará tais recursos em seu projeto e o papel do professor é de mediador do processo, dando liberdade de criação ao estudante e, só interferindo se for solicitado.

A diversão é um dos fatores mais relevantes para a aprendizagem, pois leva o estudante a querer aprender. Segundo Resnick (2007), divertir-se e aprender são ações que podem e devem estar intimamente ligadas, pois incentiva a busca por novos conhecimentos, a desenvolver suas ideias e a testar seus limites e, talvez o mais importante, a gerar novas ideias a partir de suas experiências. O compartilhamento do projeto é uma parte fundamental do processo criativo,

embora a habilidade de compartilhar e colaborar seja pouco enfatizada nas atividades escolares. Mostrar suas criações para outras pessoas e conhecer outros trabalhos facilita a troca de informações e de experiências e contribui para a reformulação dos projetos e à geração de novas ideias. Por conseguinte, a reflexão leva o estudante a analisar se o resultado obtido foi o esperado e como poderia ser modificado para melhorar. Assim, novamente haverá a etapa de imaginar o que poderá ainda ser feito, criar, se divertir, compartilhar e refletir novamente. E a espiral continua.

Com base nisso, 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 de ser destacado, pois, o professor, ao determinar que ensinará certo conteúdo matemático por meio do Scratch, deverá ter cuidado para não tornar a atividade de programação algo desinteressante e sem sentido ao estudante. É importante que eles queiram programar.

Na verdade, a programação de computadores utiliza muita matemática, mas esta, muitas vezes, não se apresenta da mesma forma que na escola. O estudante, ao programar computadores, aplica seus conhecimentos prévios de matemática, aprende e desenvolve outros conhecimentos necessários à programação, sem saber necessariamente que o está fazendo. Ele tem a compreensão de uma matemática aplicada ao problema que está resolvendo pela programação, mas pode não ter clareza da relação disto com a matemática escolar.

Para tanto, acreditamos que a programação de computadores possa facilitar a aprendizagem de conhecimentos matemáticos formais da escola, quando for possível a criação de vínculos entre o que se faz no computador e o que se aprende na sala de aula. Portanto, no tópico seguinte, apresentamos algumas atividades propostas na Escola de Hackers, assim como reflexões sobre os conhecimentos matemáticos encontrados.

Atividades de Programação na Escola de Hackers: discussão de conhecimentos informais de matemática

Segundo Marji (2014), as linguagens de programação mais comuns são baseadas em textos, que "parecem uma forma enigmática de inglês" (p.22), possuindo algumas regras de sintaxe, que a princípio são desafiadoras aos estudantes. Por exemplo, se você quiser escrever "Olá!", na linguagem Python terá que digitar: `print("Olá!")`; na linguagem C++ terá que digitar: `std::cout << "Olá!" << std::endl`; na linguagem java: `System.out.print("Olá!")`; e na linguagem php: `echo "Olá!"`. Já o Scratch, é uma linguagem de programação visual, na qual os comandos são representados por blocos a serem encaixados uns nos outros na implementação dos algoritmos, conforme a Figura 3.

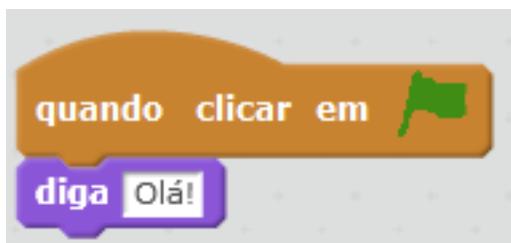


Figura 3. Programação em Scratch 2.0.

Os Parâmetros Curriculares Nacionais (Brasil, 1998), afirmam que comunicar-se em diferentes formas de linguagem é uma competência que exige do aluno fazer diferentes representações matemáticas para um determinado problema ou questão. O mesmo problema escrito na língua materna (forma narrativa) pode ser representado numa linguagem simbólica, como gráfico, equações, fórmulas, e inclusive numa linguagem de programação. Conforme já mencionado, a linguagem de programação pode servir de ponte entre a linguagem narrativa e a linguagem formal simbólica da matemática.

A matemática está intrinsecamente presente em qualquer linguagem de programação sob uma forma diferente da aprendida na escola, conforme já foi dito. Com exceção de alguns algoritmos que solucionam certos tipos de exercícios de matemática escolar, como por exemplo: faça o *sprite* desenhar um triângulo retângulo, ou faça o *sprite* calcular as raízes de uma equação do 2º grau, dados seus coeficientes; e que, portanto, estão "cheios" de matemática visível aos estudantes, todos os outros algoritmos que podem ser criados lidam, necessariamente, com a lógica matemática e com conhecimentos matemáticos informais, nem sempre visíveis aos estudantes.

Nas três atividades descritas a seguir, apresentamos uma solução possível no Scratch 2.0 e uma análise dos conhecimentos matemáticos percebidos nos códigos de programação. Estes conhecimentos, no entanto, podem ser a aplicação de algum conteúdo formal específico já compreendido pelos estudantes ou, conforme viemos argumentando, poderão dar subsídios para a compreensão da matemática formal futuramente.

Atividade 1: Crie um programa que permita que um retângulo gire de modo perceptível, quando clicada a bandeira verde.

Esta é uma atividade aplicada à um conteúdo específico da matemática. Com ela, o professor pode ter a pretensão de trabalhar conceitos de geometria e recursos do Scratch. A Figura 4, apresenta uma forma de solução, realizada no Scratch 2.0. Com isso, obtemos diversas possibilidades de giro do retângulo, considerando o seu ponto de rotação. As figuras 5 e 6, mostram duas posições diferentes do ponto de rotação.

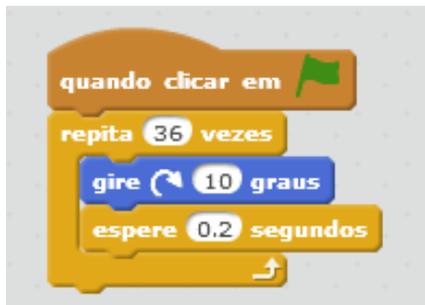


Figura 4. Programação da Atividade 1.

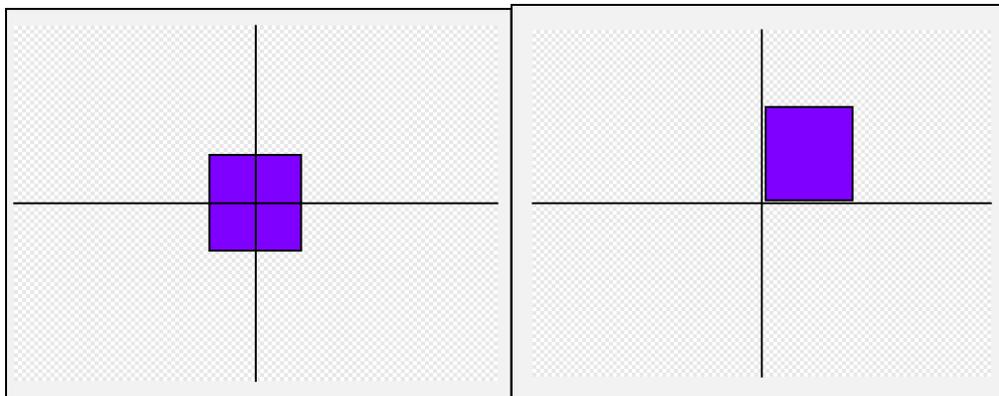


Figura 5. Ponto de rotação no centro e num dos vértices do retângulo.

Esta atividade é simples, mas contém conhecimentos interessantes. Além de trabalhar o conceito de retângulo, desenvolve a noção de centro como uma referência para o movimento, o que é importante para a construção do conceito de centro de massa, centro de gravidade da Física. Mudando o centro (Figura 5), o movimento do retângulo muda. Também são trabalhados noções de ângulos - um giro completo tem 360° - e, para o movimento ser visível no programa, é necessário que ele gire aos poucos (esperando para girar novamente alguns segundos depois). Neste caso, escolhemos a repetição de 36 vezes um giro de 10° a cada 0,2 segundos.

Atividade 2: Crie um programa que faça um objeto se mover de modo a desenhar um octógono regular.

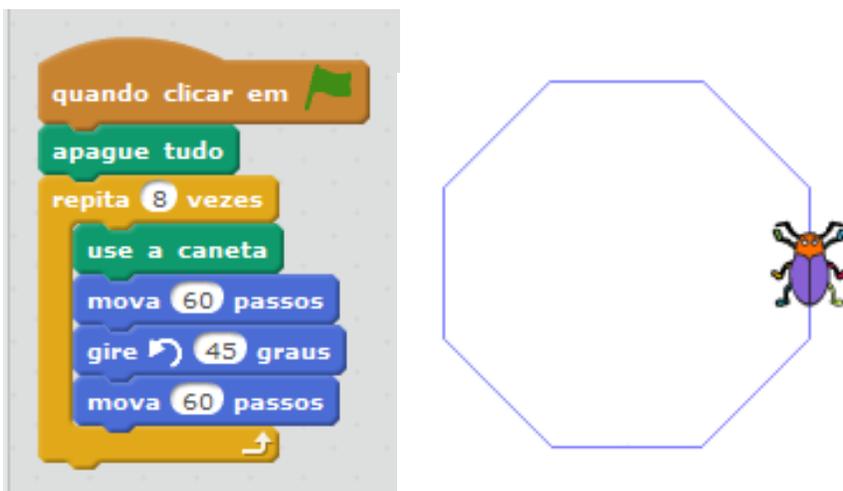


Figura 6. Programação e visualização da solução da Atividade 2.

Nesta atividade, assim como a anterior, é necessário a compreensão prévia do conceito de octógono regular para definir o ângulo de rotação do objeto. Também é necessário desenvolver o conhecimento de que o objeto da tela precisa girar conforme o ângulo externo do octógono, mas este conhecimento pode ser aplicado de forma implícita (o estudante escolhe o ângulo, mas não sabe o que é ângulo externo de polígonos). E, para formar a figura, é preciso definir o número de repetições. Com isso, o estudante poderá generalizar a construção de qualquer polígono regular.

Atividade 3: Crie um programa que solicite, via teclado, as dimensões de um retângulo, que desenhe o tal retângulo e forneça a medida de sua área.

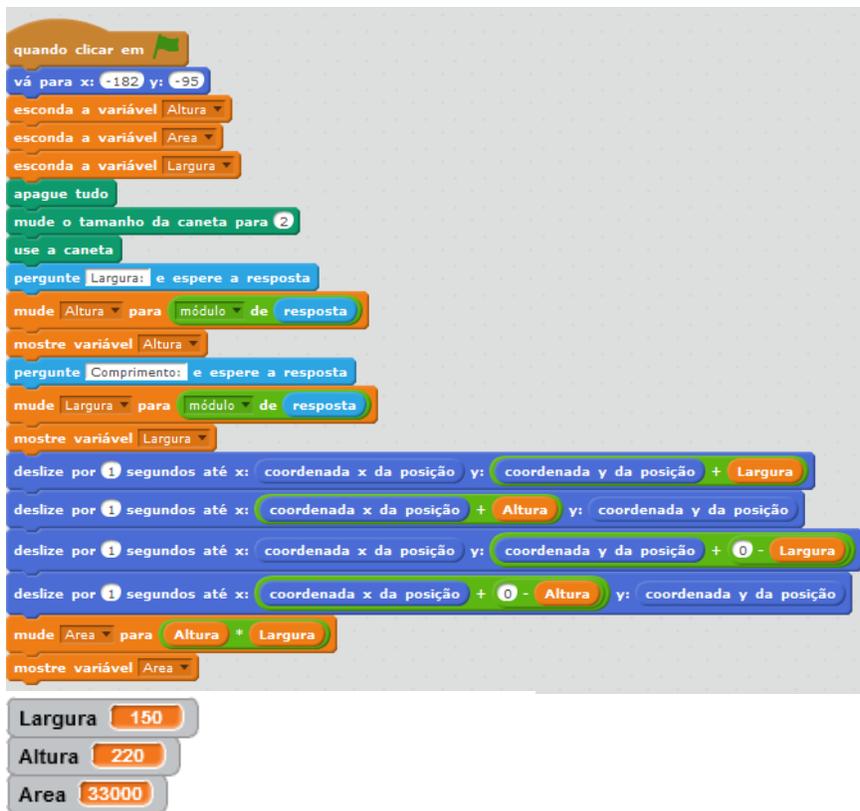


Figura 7. Programação e visualização da atividade 3.

Esta atividade desenvolve a noção de coordenadas, no momento em que iniciamos o desenho de um lugar específico na tela. Também é preciso aplicar o conceito de retângulo e de área de retângulo. Na Figura 7, vemos que foi utilizado o recurso "variável" do Scratch. Este recurso tem o sentido semelhante do que se entende de "variável" na matemática, ou seja, é um "lugar" ou "símbolo" que representará diferentes valores numéricos que escolhemos. Neste código, é necessário informar um valor numérico para a variável "largura", depois para a variável "comprimento", na qual é armazenado na memória. Depois, outra variável é criada, para guardar a resposta da operação de multiplicação entre as duas variáveis anteriores. Temos, então, diferentes conhecimentos matemáticos envolvidos, sem contar o raciocínio lógico presente na superposição das categorias (cada categoria possui uma cor) que formam os blocos, conforme pode ser observado na programação.

Atividade 4: Crie um jogo que contenha pontuação

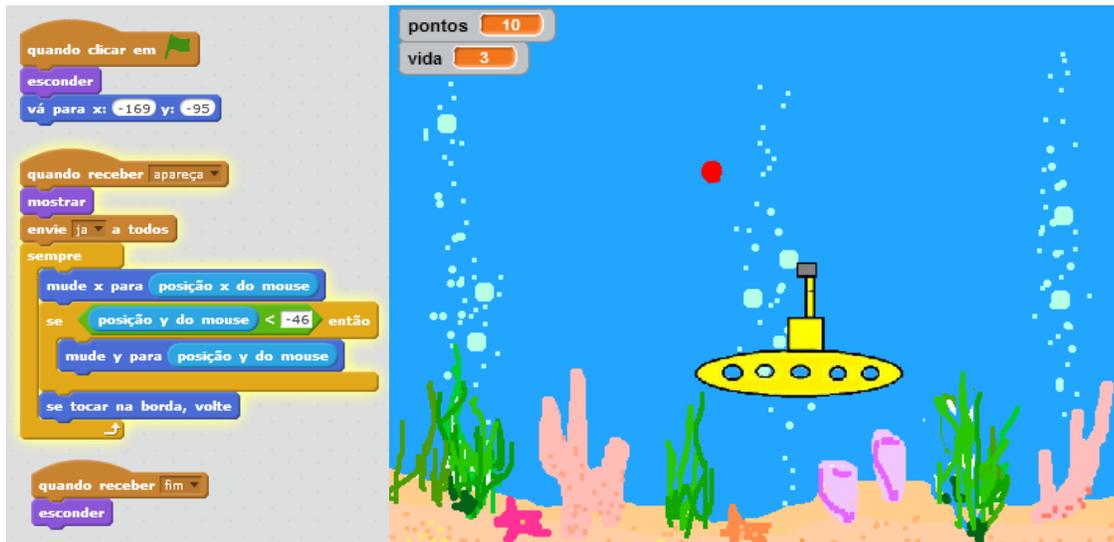


Figura 8. Programação do submarino e visualização da Atividade 4.

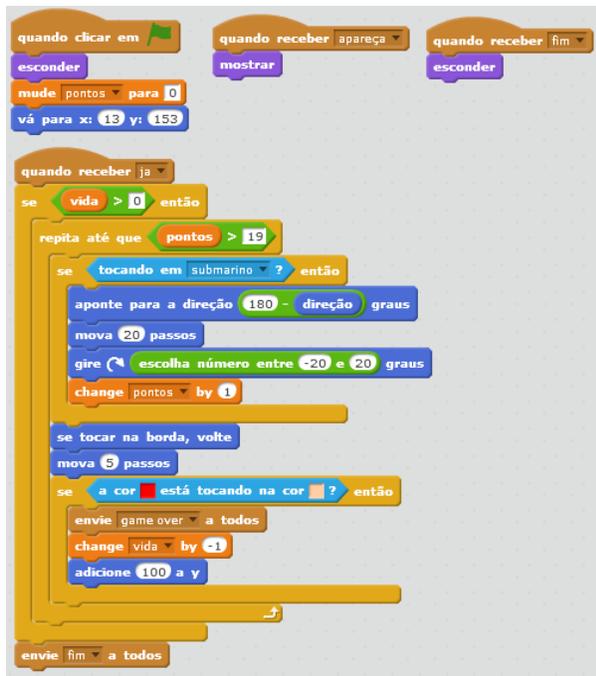


Figura 9. Programação da bolinha.

O objetivo da Atividade 4 é de não direcionar os conhecimentos envolvidos e, portanto, muitas ideias inesperadas e surpreendentes podem surgir. Na Figura 8 e 9, temos um exemplo construído na Escola de Hackers para introduzir a ideia de construção de jogos, na qual apresenta uma programação mais complexa e com dois objetos em movimento, portanto, duas telas de programação.

Na programação do submarino, percebe-se a utilização de coordenadas para estabelecer a posição de início tanto do submarino quanto da bolinha. E a condição é usada para estabelecer a posição do submarino, conforme a posição do mouse. É interessante observar que há uma limitação da altura que o submarino pode atingir. No código, temos a restrição desta posição determinada pela condição "se posição y do mouse < -46, então mude y para posição y do mouse". Logo, se a posição do mouse for maior do que -46, ele não se mexe na direção do mouse.

Na programação da bolinha, na Figura 9, percebe-se o uso de: coordenadas; operadores "-" e ">"; variáveis para guardar a pontuação e as vidas; noções de ângulos quando a bolinha toca o submarino (ângulos de incidência e reflexão); e a ideia de condição "se, então".

Conclusão

A Escola de Hackers, dessa forma, apresenta-se como uma possibilidade para os estudantes de Escolas Municipais de Passo Fundo aprenderem programação e conseqüentemente, matemática. Conforme já foi dito, a programação possibilita muitas aprendizagens e pode potencializar a compreensão da matemática formal da escola, mediante um trabalho paralelo nos laboratórios de informática das escolas com o uso de matemática informal nos códigos de programação. 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” (p. s/n, Tradução dos autores).

Para uma criança programar em Scratch, ela não precisa saber a matemática formal da escola. O *software* permite que ela conheça seus recursos brincando e teste muitas vezes suas hipóteses e, portanto, a espiral do conhecimento proporciona uma visão do processo, na qual a reflexão sobre seus códigos de programação gera as aprendizagens.

Como vimos na descrição das atividades da Escola de Hackers, o pensamento matemático referente às variáveis, por exemplo, não precisa, necessariamente ter sido trabalhado na escola para que o estudante consiga programar. E isso, pode também, ser validado para outros conhecimentos. Aliás, há uma grande possibilidade de o estudante não conseguir relacionar os dois tipos de variáveis, mesmo já tendo visto um deles na escola. O que observamos é um distanciamento dos conhecimentos formais dos aplicáveis fora da escola.

Nesta perspectiva, o GEPID, do qual os autores deste artigo fazem parte, está engajado em estudos teóricos sobre o software Scratch, assim como, na concretização da Escola de Hackers e das Olimpíadas de Programação, como possibilidades de inclusão dos estudantes na sociedade cada vez mais digital, promovida por meio do conhecimento.

Referências

Barcelos, T., & Silveira, I. F. (2012). Pensamento Computacional e Educação Matemática: Relações para o Ensino de Computação na Educação Básica. In *XX Workshop sobre Educação em Computação*. Curitiba. Anais do XXXII CSBC.

- Brasil. (1998). *Secretaria de Educação Fundamental. Parâmetros Nacionais Curriculares Nacionais: Matemática*. Brasília: MEC.
- EduScratch. (2012). *Site do Scratch para Educadores*. Disponível em: <<http://eduscratch.dgfdc.min-edu.pt>>. Acesso em: 10 jun. 2012.
- Freire, P. (2005). *Pedagogia do Oprimido*. (41ª ed.). Rio de Janeiro: Paz e Terra.
- Marji, M. (2014). *Aprenda a programar com scratch: uma introdução visual à programação com jogos, arte, ciência e matemática*. São Paulo: Novatec.
- Marques, Maria T. P. M. (2012). *Recuperar o engenho a partir da necessidade, com recurso às tecnologias educativas: contributo do ambiente gráfico de programação Scratch em contexto formal de aprendizagem*. Universidade de Lisboa, 2009. Disponível em: <http://eduscratch.dgfdc.min-edu.pt/index.php?option=com_docman&task=doc_download&gid=43&Itemid=40>. Acesso em: 10 jun. 2012.
- Papert, S. (2008). *A máquina das crianças: repensando a escola na era da informática* (Ed.rev). Porto Alegre: Artmed.
- Resnick, M., & Rosenbaum, E. (2013). Designing for Tinkerability. In M. Honey, & D. Kanter (Eds.), *Design, Make, Play: Growing the Next Generation of STEM Innovators* (pp. 163-181). Routledge.
- Resnick, M. (2007). All I Really Need to Know (About Creative Thinking) I Learned (By Studying How Children Learn) in Kindergarten. *ACM Creativity e Cognition conference*, Washington DC, June.
- Resnick, M. et al. (2009). *Scratch: programming for all*. Communication of the ACM, 52(11), 60-67.
- Resnick, M. (2013). *Learn to code, code to learn*. MIT Media Lab. Artigo on-line. Disponível em: <<http://web.media.mit.edu/~mres/papers/L2CC2L-handout.pdf>> Acesso em: 18 maio 2014.