

O SCRATCH COMO FERRAMENTA DE APRENDIZADO DE LÓGICA DE PROGRAMAÇÃO

**Bruno Gomes Rodrigues¹; Cleide Barbosa de Souza¹; Dienne Aparecida Pereira¹;
Pollyana de Queiroz Ribeiro²**

¹Discente do curso de Sistemas de Informação da UEG - Câmpus Santa Helena, brunnorodrigues@brunnorodrigues.jor.br; cleidebarbozadesousa@hotmail.com; dienneapereira11@gmail.com

²Docente do curso de Sistemas de Informação da UEG - Câmpus Santa Helena, pollyanaqueiroz@gmail.com

RESUMO: Este trabalho busca relatar o uso da linguagem de programação *Scratch*, como forma de auxiliar o desenvolvimento do pensamento e raciocínio lógico, que é um dos quesitos mais importantes para desenvolver sistemas. Essa linguagem trabalha a lógica de programação, a partir da formação de blocos visuais, personagens e animações, o que facilita o entendimento e o aprendizado por parte do programador.

Palavras-chave: Raciocínio lógico, *Scratch*, programação

SCRATCH AS A LEARNING TOOL FOR PROGRAMMING LOGIC

ABSTRACT: This work seeks to report the use of the Scratch programming language, as a way to aid the development of thinking and logical reasoning, which is one of the most important requirements for developing systems. This language works the logic of programming, from the formation of visual blocks, characters and animations, which facilitates understanding and learning by the programmer.

Key-words: Logical reasoning, Scratch, programming

INTRODUÇÃO

O aluno de curso superior na área de tecnologia da informação, normalmente, tem disciplinas relacionadas a programação. Essas disciplinas de programação, requer um bom raciocínio lógico, por parte dos estudantes. De modo geral, os alunos têm dificuldade no que tange a raciocínio lógico, e alguns autores, como MEDEIROS; MENEZES e PEREIRA (2012), confirmam que isso é um dos fatores do alto índice de reprovação e desistência dos cursos da área.

Sendo a programação o resultado do entendimento do problema e de sua resolução (a lógica) e o conhecimento da linguagem que será adotada, há algumas alternativas para introdução da programação. Entre elas, a linguagem *Scratch*, é uma das formas de tentativa de

minimizar ou sanar problemas no desenvolvimento do raciocínio lógico na construção de algoritmos.

O *Scratch* é um programa que simplifica o processo de desenvolvimento de algoritmo, por trabalhar com uma interface gráfica bastante intuitiva, que oferece o movimento e o som em imagens. Assim, a ideia de programar com inúmeras linhas de código, em um primeiro momento, não é abordada, significando que, a preocupação em aprender está somente na lógica em si, e não ela e linguagens ao mesmo tempo.

Nesse contexto, a discussão sobre a eficácia do *Scratch* quanto a compreensão de lógica de programação é importante, já que, como citado acima, ela é apontada como a que mais resulta na desistência e na reprovação de alunos em cursos da área, confirmada a necessidade de aplicações de formas mais simples e mais próximas do estudante para que ele compreenda lógica e só a partir daí, aprenda as linguagens que vão de fato resultar num produto final (um sistema, um *software*, e etc.).

Mas o *Scratch* de fato ajuda com que o futuro programador passe a ter um raciocínio ágil no momento de desenvolver um sistema ou um *software* em outras linguagens de programação? É o que este estudo tenta comprovar, apresentando a experiência de alunos do curso de Sistemas de Informação, em um projeto de extensão, que inclusive já passaram pelas disciplinas de lógica com algumas dificuldades e que agora, tendo a experiência com essa linguagem, vão verificar suas contribuições com relação ao aprendizado.

O SCRATCH

O portal *Scratch* Brasil (2014, s/p) define a ferramenta gratuita de estudos e facilitação da compreensão de linguagem de programação de forma didática, que, segundo ele, é utilizada em mais de 150 países ao redor do mundo, como:

O *Scratch* é um *software* que se utiliza de blocos lógicos, e itens de som e imagem, para você desenvolver suas próprias histórias interativas, jogos e animações, além de compartilhar de maneira online suas criações. [...] é um projeto do grupo *Lifelong Kindergarten* no *Media Lab* do MIT (Instituto de Tecnologia de *Massachusetts*), onde foi idealizado por *Mitchel Resnick*. Ele foi projetado especialmente para idades entre 8 e 16 anos, mas é usado por pessoas de todas as idades (*SCRATCH BRASIL*, 2014, s/p).

A utilização do programa, em números, está em um infográfico, disponibilizado pelo próprio *Scratch* Brasil, conforme identificação na Figura 1.

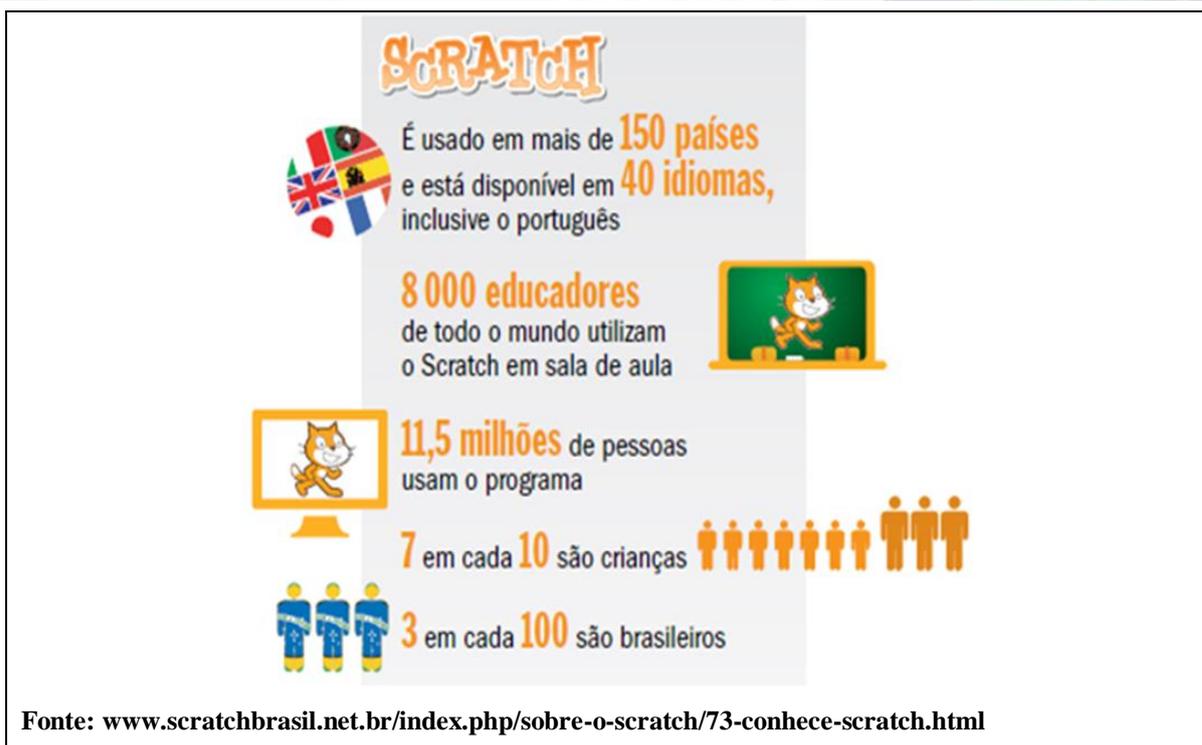


Figura 1: Números da utilização do *Scratch* por localidade, por público-alvo e por quantidade.

Arranhão, rabisco, risco são algumas traduções que o *Google Tradutor* sugere ao enviar a palavra *Scratch*, o que traz a ideia de que criar uma animação, um jogo, um gráfico qualquer é tão fácil quanto rabiscar, arranhar... E há sim uma extrema facilidade em construir um projeto, pois há apenas que arrastar as peças e ir montando-as logicamente em um bloco, para que uma ação ou comportamento aconteça. Medeiros; Menezes e Pereira (2012, s/p), sobre o termo, trazem:

O termo *Scratch* provém da técnica de *scratching* utilizada pelos *Disco-Jockeys* do *HipHop* que giram os discos de vinil com as suas mãos para frente e para trás de modo a fazer misturas musicais de forma criativa e inesperada. Com o *Scratch* é possível fazer algo de semelhante, misturando diferentes tipos de trechos de mídia (gráficos, fotos, músicas, sons) de formas criativas (MEDEIROS; MENEZES; PEREIRA, 2012, s/p).

O personagem principal de toda a ideia é um gato. O animal pode andar, correr, dançar, falar, miar, responder a perguntas, se algo for feito antes, ou alguma coisa pode ser feita após, enfim, fazer o que a imaginação do usuário propor. Há outros *sprites* (atores) padrões, além do felino e, com isso, muitas outras possibilidades, o que torna possível uma melhor compreensão, de modo geral, da ordem das coisas e de como elas acontecem em um *sistema* ou *software* funcional.

Agora, observando mais a parte que corresponde aos princípios de programação e de qualquer linguagem utilizada para essa finalidade, o *Scratch* também possibilita a criação e

utilização de variáveis; de operações relacionais (maior, menor ou igual) e de operadores lógicos (e, ou e não), no caso de trabalhar com comparações de valores; de operadores matemáticos, que vão possibilitar a soma, a subtração, a multiplicação e a divisão de números; e de condições se e senão, para que algo aconteça se atender a uma proposta ou não.

Há também como implementar a ideia de laços de repetição na linguagem, que ficam a cargo do “Sempre” e do “Sempre se”, do “Repita” e do “Repita até”, que verificarão se uma condição é verdadeira ou não, repetidamente. Vale lembrar que uma estrutura, pode ser implementada dentro da outra, fazendo múltiplas tarefas, a fim de retornar a ação desejada ao final.

O *Scratch* oferece a possibilidade de trabalhar com texto e números ao mesmo tempo; exibir mensagens na tela, para garantir a eficácia na interação com o usuário; controlar todas as solicitações de parada e de movimento (para cima, para baixo, para os lados...) do objeto visual com que está sendo trabalhado. As categorias Sensores e Caneta, ligadas diretamente a ação do personagem, respectivamente, faz algo, após outra coisa ser feita e faz com que ele desenhe algo na tela.

Tudo isto, colabora com o entendimento de uma lógica abstrata, muitas vezes em mente apenas, mas que, se desenvolvida corretamente vai tornar um programa, em qualquer linguagem (desde que o programador também conheça e domine a plataforma e suas possibilidades), totalmente funcional e útil e que realmente supra as necessidades do cliente ou do usuário.

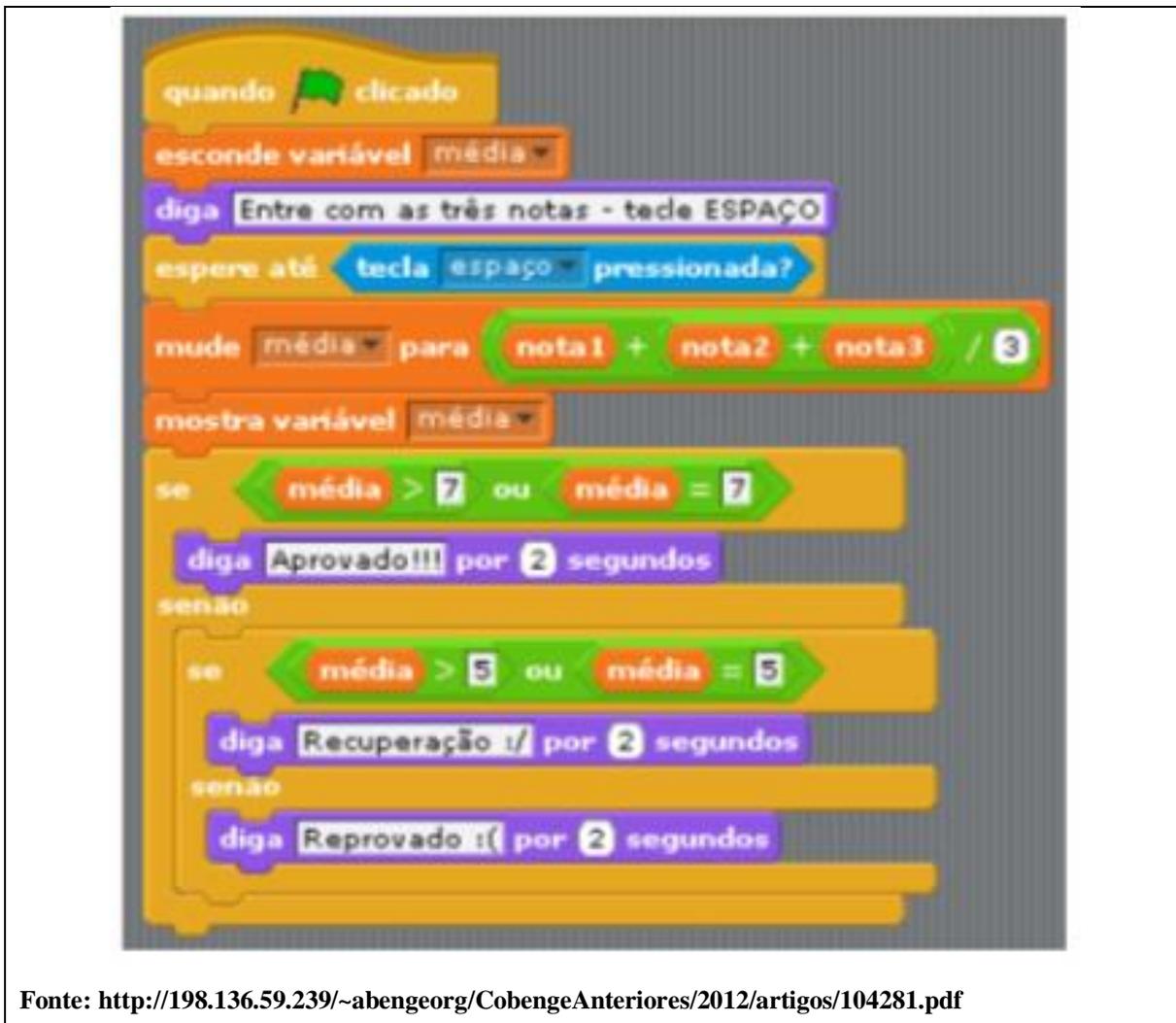
O USO DO SCRATCH NO APRENDIZADO DE LÓGICA DE PROGRAMAÇÃO

MEDEIROS; MENEZES e PEREIRA (2012) relataram a aplicação e os resultados de um teste de um problema comum a ser resolvido como exercício dentro de sala de aula em disciplinas de lógica de programação: o de obter a média aritmética após a entrada de três notas, ou seja, será feita a entrada de três valores, correspondentes às notas de três provas, respectivamente, e deverá ser mostrado de o aluno foi aprovado, caso sua média for maior ou igual a 7,0; se ficará de recuperação, caso tenha obtido menos de 7,0 e mais de 5,0 e se foi reprovado, caso a média seja menor que 5,0.

Antes da implementação e construção de um algoritmo no *Scratch*, os autores sugerem uma espécie de identificação do que serão dados de entrada, do cálculo que será feito e aqui, possivelmente entra a parte de mais raciocínio lógico e da saída de dados, escrevendo tudo isso em forma de respostas para perguntas como: “Quais são os dados de entrada?” “Qual

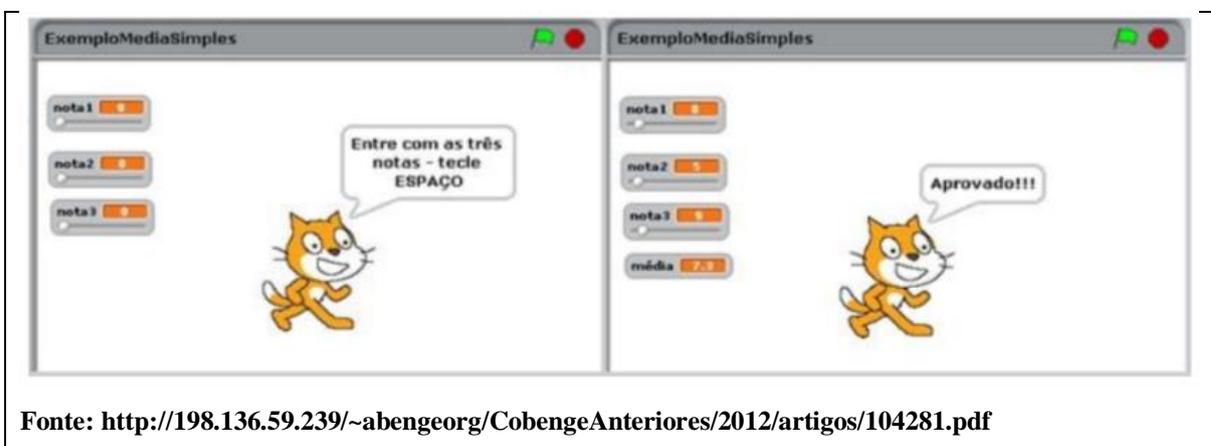
11^a JORNADA ACADÊMICA

será o processamento a ser utilizado?” e “Quais serão os dados de saída?”. A Figura 2, apresenta os resultados:



Fonte: <http://198.136.59.239/~abengeorg/CobengeAnteriores/2012/artigos/104281.pdf>

Figura 2: Estrutura do programa para obter a média.



Fonte: <http://198.136.59.239/~abengeorg/CobengeAnteriores/2012/artigos/104281.pdf>

Figura 3: Execução do programa.

MENDONÇA NETO (2013), aplicou uma pesquisa de campo com alunos de 16 e de 17 anos, de um curso técnico em informática de uma instituição de ensino de Maranhão, e ressalta que:

Em 2012, a instituição iniciou quatro turmas, 63 alunos, nos turnos manhã e tarde. Nessas, a disciplina de Lógica de Programação foi ministrada da seguinte forma: 15 aulas com exposição da teoria, resolução de exercícios, com o aumento gradativo da dificuldade, utilizando o papel e/ou o aplicativo *Visualg*. Durante as aulas identificou-se que os alunos, em quantidade bastante expressiva, apresentavam grandes dificuldades na compreensão e aplicação dos conceitos de programação (MENDONÇA NETO, 2013, s/p).

Com resultados comprovando desmotivação e dificuldades em utilizar o *Visualg* e em compreender a matemática por trás dos algoritmos, houveram altos números de reprovações e desistência. No ano seguinte, o *Scratch* foi utilizado nas primeiras aulas de lógica de programação nas turmas do curso.

Nestas, 36 alunos se submeteram a essa nova abordagem. Assim a disciplina foi estruturada da seguinte forma: das 15 aulas, 5 foram destinadas ao ensino da ferramenta, dos conceitos básico (operadores, variáveis, comandos de entrada e saída, etc.) e comandos de seleção e repetição e 10 destinadas ao ensino da disciplina com o *Visualg* (MENDONÇA NETO, 2013, s/p).

Os resultados com a introdução da utilização do *Scratch*, o autor concluiu que:

Identificamos que alguns estudantes tiveram maior desempenho na disciplina no primeiro momento, programando com o *Scratch*, que no segundo utilizando o *Visualg*. Questionados a respeito alegaram que com a ferramenta era mais simples e divertido criar programas. Outro ponto observado pelos alunos é o fato de poder utilizar ferramentas como som e imagem, por exemplo, associado ao ambiente gráfico, com blocos coloridos que são encaixados e a possibilidade de, utilizando esses recursos, criar jogos, animações e histórias animadas. Estes aspectos geraram um clima mais divertido e descontraído durante as aulas e curiosidade e frustração por parte dos alunos de outras turmas, que estão finalizando o curso técnico em informática este ano, que desejaram ter aprendido a programar utilizando esta nova abordagem. Quando perguntados se a ferramenta ajudou na compreensão da disciplina de Lógica de Programação, apenas 1 aluno marcou a opção não na pesquisa. Ao ser questionado sobre a negativa nas respostas, o mesmo justificou dizendo que não gostou de desenvolver jogos, por não ter afinidade para tal, e que se não se adaptou a forma de programar do software (MENDONÇA NETO, 2013, s/p).

Para MENDONÇA NETO (2013), a ferramenta *Scratch* como uma nova abordagem, auxiliou no processo de aprendizagem dos alunos nos conceitos iniciais da disciplina de lógica de programação e trouxe resultados bastante favoráveis, graças ao ambiente interativo e descontraído no ato de programar. O programa torna a atividade mais atraente, fácil e completamente intuitiva, ainda segundo ele, já que os alunos podem inserir em seus programas, elementos como sons e imagens que o próprio programa traz ou criados por eles.

MATERIAIS E MÉTODOS

Foram utilizados como materiais para este estudo, as conclusões e opiniões de autores, especialistas em *Scratch*, como Marji (2014), Júnior; Lopes e Martinelli (s/d); Medeiros, Menezes e Pereira (2012) e Mendonça Neto (2013), publicadas em livros, artigos científicos e apostilas que ensinam a utilização da ferramenta, além da resolução de exercícios práticos:

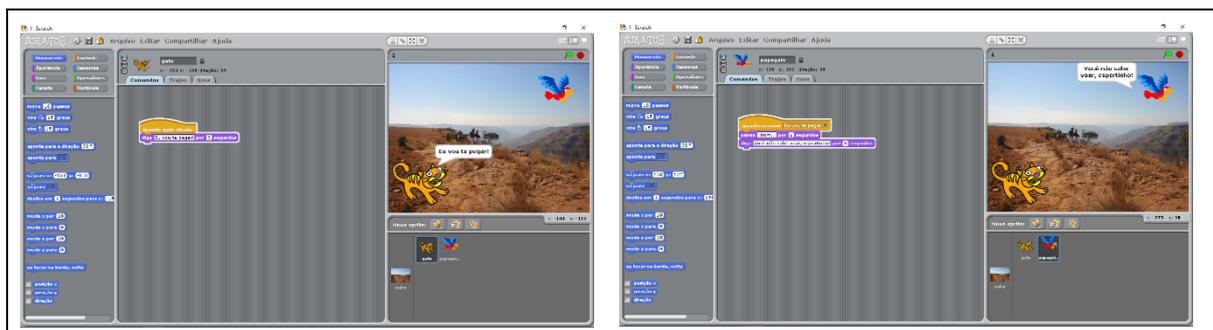


Figura 4: Diálogo entre dois atores, em que um espera a fala do outro para então responder.

Na Figura 4, é entendida a lógica que há em um diálogo, em que, um personagem só fala após o término da fala do outro, o que, neste caso, representou também a maior dificuldade.

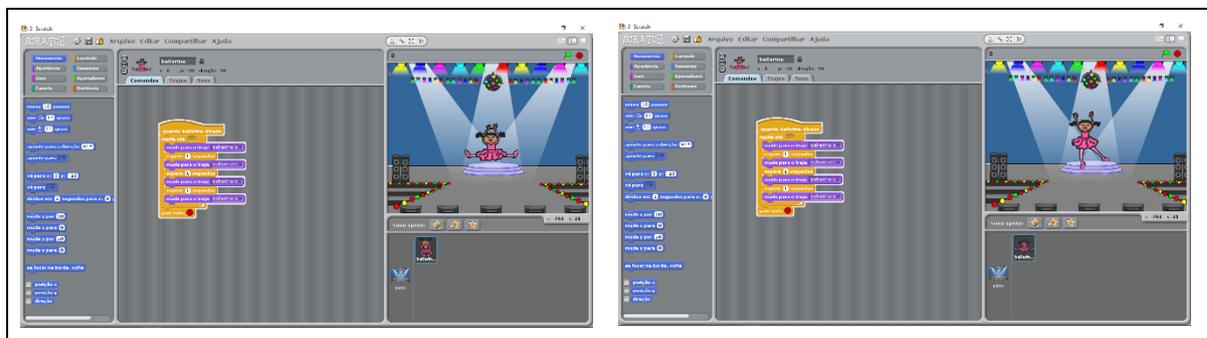


Figura 5: Cenário em que o ator se movimenta, em um *looping*, simulando dança.

Na Figura 5, é entendido facilmente como funciona a estrutura de um laço de repetição e, na prática, como ele pode ser utilizado. Na Figura 6, o aprendizado é relacionado ao comando de decisão quando, ou seja, somente quando o mago diz as palavras mágicas, o sapo desaparece, sendo a única condição capaz de realizar tal tarefa.

11^a JORNADA ACADÊMICA



Figura 6: O mago diz “Abracadabra” e o outro ator (sapo) some da tela por 5 segundos.

Na Figura 7, inclusive o mais fácil na implementação, não houve nenhuma dificuldade, já que, como percebido, com apenas duas linhas, o efeito acontece, sem contar que o comando de decisão utilizado anteriormente, também foi aplicado aqui.

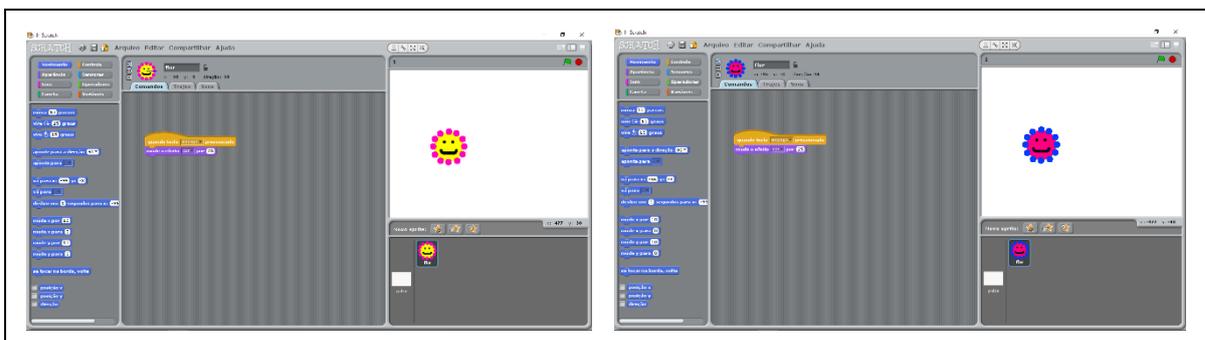


Figura 7: Efeito de mudança de cor da flor ao pressionar a tecla Espaço no teclado.

Na Figura 8, ainda trabalhando com a condição de decisão, foi apresentada também a estrutura de repetição sempre, ou seja, para que o dançarino mude de posição ao som do tambor, toda a instrução para tal tarefa deve vir dentro do laço de repetição.



Figura 8: Ao som de uma música, ator muda de posição no cenário, simulando dança.

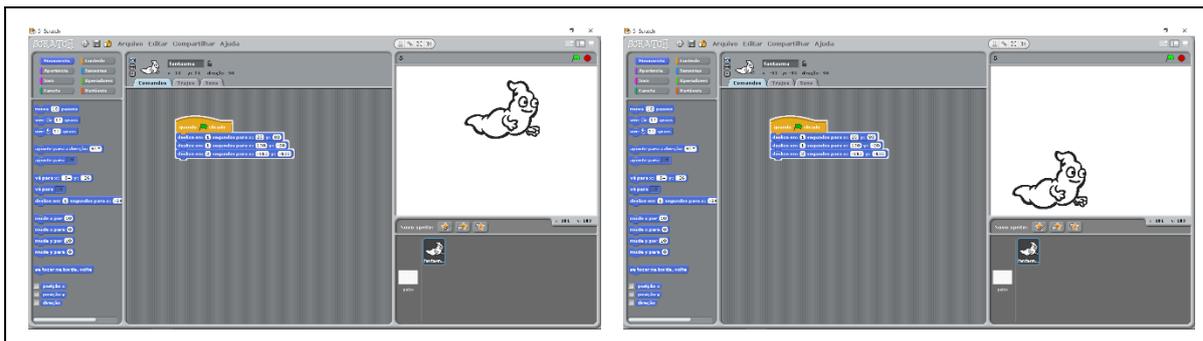


Figura 9: Fantasma se move no cenário, de um lado ao outro.

Neste exercício um fantasma “voa” no cenário, o que não representou dificuldade nem na implementação nem na construção do cenário.

Neste último exemplo apresentado, foi utilizado o mesmo conceito utilizado no exercício representado na Figura 6, portanto sem grandes dificuldades.

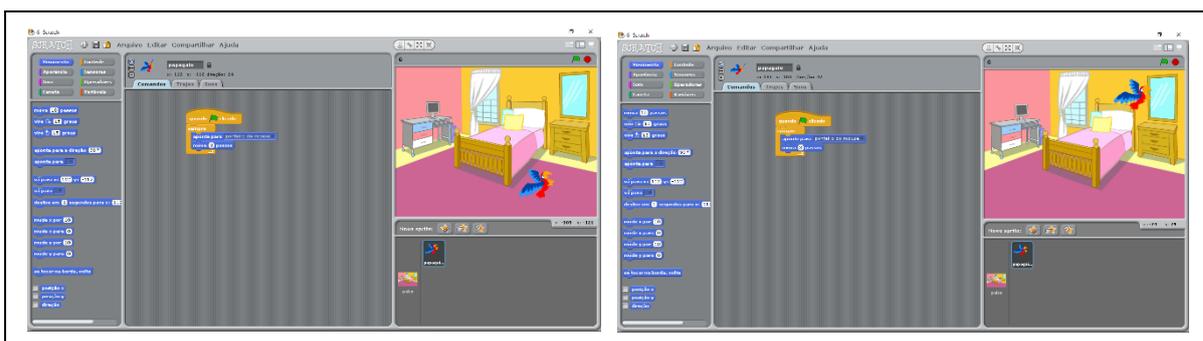


Figura 10: Papagaio segue voando, o ponteiro do mouse dentro do cenário.

CONCLUSÕES

Conclui-se, com este Relato de Experiência, que de fato o *Scratch* auxilia e proporciona uma noção de lógica de programação a quem se propõe ou precisa adquirir esse tipo de conhecimento. Sua simplicidade e possibilidades, torna a compreensão dessa técnica muito mais didática, além de a ferramenta poder ser utilizada por pessoas de qualquer idade, o que estimula ainda a criatividade.

A noção de organização e ordem de tarefas ou da execução de algum passo, é completamente compreendida por estudantes de programação ou por até mesmo por quem quer se divertir, já que o *Scratch* também possibilita que o usuário aprenda de uma forma leve e até brincando. Por fim, com base em pesquisas e trabalhos práticos pode-se dizer que a ferramenta facilita e possibilita o aprendizado de lógica de programação.

REFERÊNCIAS

MARJI, Majed. **Aprender a Programar com Scratch** - Uma introdução visual à programação com jogos, arte, ciência e matemática. São Paulo: Novatec, 2014. Disponível em: <<https://s3.novatec.com.br/capitulos/capitulo-9788575223123.pdf>>. Acesso em: 22 de agosto de 2017.

JÚNIOR, Jucimar de Souza Lima; LOPES, Vitor de Carvalho Melo; MARTINELLI, Suéllen Rodolfo (reorganização e adaptação). **Tutorial de Lógica Básica no Scratch**. s/d. Disponível em: <[http://www.scratchbrasil.net.br/images/download-materiais/Lógica%20Básica%20no%20Scratch%20-%20Scratch%20Brasil%20\(tutorial%205\).pdf](http://www.scratchbrasil.net.br/images/download-materiais/Lógica%20Básica%20no%20Scratch%20-%20Scratch%20Brasil%20(tutorial%205).pdf)>. Acesso em 21 de agosto de 2017.

MEDEIROS, Marcos; MENEZES, José Wally Mendonça; PEREIRA, Pryscilla de Sousa. **Análise do Scratch como ferramenta de auxílio ao ensino de programação de computadores**. 2012. Disponível em: <<http://198.136.59.239/~abengeorg/CobengeAnteriores/2012/artigos/104281.pdf>>. Acesso em 21 de agosto de 2017.

MENDONÇA NETO, Valter dos Santos. **A utilização da ferramenta Scratch como auxílio na aprendizagem de lógica de programação**. 2013. Disponível em: <<http://www.br-ie.org/pub/index.php/wcbie/article/view/2675/2329>>. Acesso em 29 de outubro de 2017.

SCRATCH BRASIL – Sobre o Scratch – **Você conhece o Scratch?**. 2014. Disponível em: <<http://www.scratchbrasil.net.br/index.php/sobre-o-scratch/73-conhece-scratch.html>>. Acesso em: 21 de agosto de 2017.