

# Avaliando o uso do Scratch como abordagem alternativa para o processo de ensino-aprendizagem de programação

Viviane Cristina Oliveira Aureliano<sup>1,2</sup>  
Patrícia Cabral de Azevedo Restelli Tedesco<sup>1</sup>

<sup>1</sup>Centro de Informática – Universidade Federal de Pernambuco (UFPE)  
50740-560 – Recife – PE – Brasil

<sup>2</sup>Instituto Federal de Pernambuco – Campus Belo Jardim (IFPE)  
55150-000 – Belo Jardim – PE – Brasil  
{vcoa,pcart}@cin.ufpe.br

**Abstract.** *Teaching how to program is one of the big challenges in Computing Education. In this context, this paper presents a preliminary study aimed at evaluating the learning environment Scratch as an alternative approach to the traditional teaching programming process. In order to evaluate the efficacy of this alternative approach, we have compared it to traditional one. The results obtained suggest that it is a good practice to adopt these two approaches together. Such a way of teaching should be investigated in further detail, especially with regards to students' grades and participation in the classroom.*

**Resumo.** *Ensinar programação é um dos grandes desafios na Educação em Computação. Neste âmbito, este artigo apresenta um estudo preliminar que objetiva avaliar o ambiente introdutório de ensino Scratch como abordagem alternativa para o processo de ensino-aprendizagem de programação, contrastando esta abordagem a uma abordagem tradicional de ensino. Os resultados obtidos sugerem que uma estratégia que adote estas duas abordagens em conjunto deve ser investigada mais detalhadamente no que diz respeito às notas obtidas e também em outros aspectos, tais como a motivação e participação dos alunos em sala de aula.*

## 1. Introdução

O ensino de programação para iniciantes de cursos da área de tecnologia é uma preocupação que vem crescendo ao longo dos anos em virtude das dificuldades pelas quais grande parte dos alunos passa ao iniciar suas vidas como programadores. Por isso, ensinar programação é um dos sete grandes desafios na educação em Computação [Caspersen e Kölling 2009].

Anos de pesquisas na área mostram que são várias as dificuldades pelas quais passam os alunos. Conceitos que dependem de um entendimento do programa por inteiro, tais como a elaboração de um projeto de um programa, a modularização ou a retirada de erros [Lahtinen, Ala-Mutka e Järvinen 2005], e conceitos abstratos, como recursão e ponteiros [Lahtinen, Ala-Mutka e Järvinen 2005; Milne e Rowe 2002], são algumas das primeiras dificuldades citadas pela literatura. Métodos tradicionais de ensino que não consideram as particularidades de aprendizado de cada um dos alunos e a maneira

inadequada de estudo também são mencionados [Gomes e Mendes 2007]. Contudo, o maior dos problemas está relacionado à combinação e à utilização apropriada dos conceitos básicos de programação para a construção de um programa [Caspersen e Kölling 2009; Lahtinen, Ala-Mutka e Järvinen 2005]. Desta forma, os alunos parecem entender os conceitos e as estruturas que compõem uma linguagem de programação, mas não sabem o momento correto de utilizá-las durante a elaboração dos seus programas.

Pesquisas também apontam vários fatores de sucesso no ensino de programação. Alguns trabalhos afirmam que possuir experiência prévia com programação tem um efeito positivo no sucesso de cursos introdutórios de programação [Byrne e Lyons 2001; Hagan e Markham 2000]. Outros mostram que a habilidade matemática e a exposição a cursos matemáticos são importantes indicativos do bom desempenho obtido em disciplinas introdutórias nos cursos de Ciência da Computação [Bergin e Reilly 2005; Byrne e Lyons 2001; Wilson e Shrock 2001]. Relações entre os diferentes estilos de aprendizagem dos alunos e o aprendizado de programação também foram encontradas. Thomas e co-autores (2002), por exemplo, listam como um dos problemas do ensino em programação que o aprendizado somente pode ser melhorado quando os professores levarem em consideração que cada aluno possui um estilo de aprendizado diferente e, por isso, necessidades e preferências de aprendizado particulares.

Aliada às pesquisas anteriormente citadas, outra linha que tem ganhado destaque dentro da área de Educação em linguagens de programação inclui a construção de ferramentas que apóiam o processo de ensino e aprendizagem dos estudantes. Entre elas, algumas das mais conhecidas são: Alice [Cooper, Dann e Pausch 2000], Greenfoot [Henriksen e Kölling 2004] e Scratch [Resnick *et al* 2009]. Classificadas como “Ambientes Introdutórios de Ensino” - *Initial Learning Environments* [Fincher *et al.* 2010], todas elas apresentam uma linguagem de programação visual e têm por objetivo apoiar os seus usuários no aprendizado de conceitos de programação de uma maneira mais divertida e estimulante. Dentre estas ferramentas, a introdução do Scratch como ferramenta de apoio ao ensino de uma disciplina introdutória de programação na Universidade de Harvard foi encarada como positiva para o aprendizado de alunos iniciantes na disciplina [Malan e Leiner 2007].

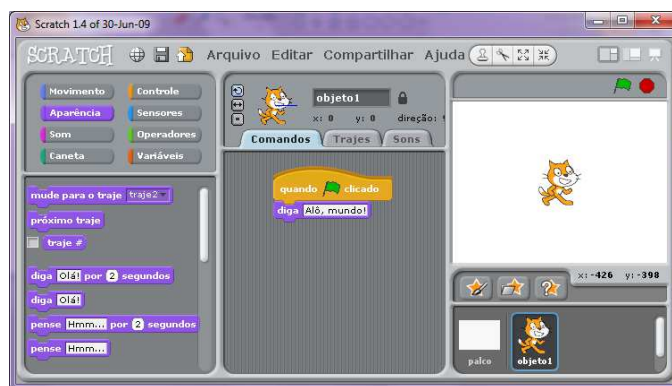
Apesar das vantagens relatando o ambiente Scratch como parte integrante de uma abordagem alternativa para o ensino de programação para iniciantes, não foi encontrado nenhum estudo que avalie os benefícios de sua utilização nesse contexto, comparando-o a uma abordagem tradicional de ensino. O estudo preliminar apresentado neste trabalho, então, tem por objetivo avaliar o processo de ensino-aprendizagem de programação para alunos iniciantes através da utilização do Scratch, a qual denominamos abordagem alternativa. Tal abordagem deve ser contrastada com uma abordagem tradicional que envolve a utilização da linguagem de programação C e o ambiente de desenvolvimento Dev-C++ [Dev-C++ 2012].

O restante do artigo está organizado da seguinte maneira. Na seção 2 é feita a apresentação da ferramenta Scratch. O estudo realizado neste trabalho é descrito na seção 3. Na seção 4 são apresentados os resultados. Por último, na seção 5 são apresentados as conclusões e os trabalhos futuros.

## 2. Scratch

Desenvolvido pelo *Lifelong Kindergarten Group (LLK)*, grupo de pesquisa do *MIT Media Lab*, o ambiente introdutório de ensino Scratch foi criado com o propósito de introduzir a programação de maneira fácil e rápida para aqueles que não possuem nenhum tipo de experiência no assunto [Maloney et al. 2010]. Para tanto, o processo de ensino-aprendizagem, com a utilização do Scratch, ocorre através de uma linguagem de programação visual que permite a manipulação de mídias, tais como imagens e músicas, para a criação de histórias interativas, de jogos ou de animações, por exemplo [Maloney et al. 2010; Maloney et al. 2008].

Um projeto no Scratch é formado por um palco e por diferentes objetos. O palco é o plano de fundo estático onde os objetos podem executar as suas ações. Para dar vida a seus projetos e definir o comportamento de cada um dos objetos presentes, os usuários associam a estes objetos sons, imagens e variáveis que podem ser manipulados pelos diferentes tipos de comandos fornecidos pela linguagem de programação do ambiente. Os comandos do Scratch precisam somente ser arrastados e soltados no espaço de comandos, onde, posteriormente, podem ser encaixados uns nos outros. Os comandos no Scratch assemelham-se a peças de quebra-cabeça e indicam com sua forma como podem ser combinados corretamente. Desta maneira, os comandos quando combinados formam programas que estão sempre sintaticamente corretos. Sem a preocupação com erros sintáticos, os usuários focam apenas na criação da lógica de funcionamento de seus projetos [Malan e Leitner 2007].



**Figura 1. Ambiente introdutório de ensino Scratch.**

O Scratch vem sendo aplicado em diferentes projetos e vem contribuindo positivamente para a criatividade, a resolução de problemas e a colaboração de seus usuários [Resnick et al. 2009; Maloney et al. 2008; Resnick 2002]. Através da criação de histórias interativas, jogos e animações, os proponentes do Scratch afirmam que é possível tornar as pessoas que fazem uso deste ambiente, digitalmente fluentes, de maneira tal que elas passam a ser construtoras e não somente usuárias de tecnologia [Resnick et al. 2009]. Além disso, acreditando na facilidade de programação fornecida pela ferramenta, Malan e Leitner (2007) obtiveram resultados positivos na utilização do Scratch como ferramenta de apoio ao ensino da disciplina introdutória de programação na Universidade de Harvard.

A linguagem de programação C, por sua vez, foi desenvolvida em 1973 por Dennis M. Ritchie para a construção do sistema operacional Unix e continua sendo muito popular

até os dias atuais. Também bastante conhecido, o ambiente de programação Dev-C++ é utilizado para a edição e compilação de programas escritos na linguagem C. Ao contrário dos programas feitos no Scratch, para construir os seus programas em C, um aluno precisa conhecer as palavras-chave da linguagem e a maneira como estas palavras formam uma estrutura sintaticamente correta. Conseqüentemente, além de se preocupar com a lógica de funcionamento dos seus programas, os alunos precisam também estar preocupados com as particularidades de uma linguagem de programação convencional. Preocupações como estas envolvem, por exemplo, usar corretamente um laço de repetição *for* ou finalizar uma instrução com um ponto-e-vírgula (;). Para ilustrar a diferença entre as duas linguagens e seus ambientes, apresentamos nas Figuras 1 e 2, o programa mais simples que comumente é ensinado nas primeiras aulas de disciplinas introdutórias de programação em cada um dos ambientes.

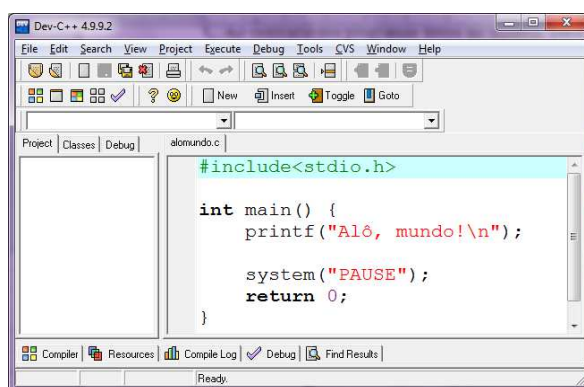


Figura 2. Programa simples escrito em C utilizando o Dev-C++.

Acreditamos que ensinar programação utilizando o Scratch possibilita um melhor desempenho do ponto de vista das notas obtidas por alunos iniciantes em programação.

### 3. Descrição do estudo

O estudo preliminar apresentado neste trabalho tem por objetivo avaliar o processo de ensino-aprendizagem de programação para alunos iniciantes através de uma abordagem alternativa de ensino que utiliza o Scratch como linguagem de programação e ambiente de desenvolvimento. Tal abordagem deve ser contrastada com uma abordagem tradicional que envolve a utilização da linguagem de programação C e o ambiente de desenvolvimento Dev-C++. Sendo assim, a questão que deve ser respondida neste trabalho é a seguinte: *Ensinar programação utilizando o ambiente de programação visual Scratch possibilita um melhor desempenho para os alunos iniciantes em programação?* O desempenho dos alunos será avaliado em função das notas obtidas por eles em cada uma das abordagens.

#### 3.1. Participantes

O estudo foi realizado em uma instituição de ensino que atua na educação de nível técnico e superior no Estado de Pernambuco. O estudo ocorreu no segundo semestre de 2011 e contou com a participação de 29 alunos matriculados na disciplina de Lógica de programação, disciplina introdutória de programação de um curso técnico profissionalizante em Informática da referida instituição. Neste curso, a disciplina de Lógica de Programação vem seguindo uma abordagem de ensino tradicional. Nesta

abordagem, as aulas são ministradas utilizando-se uma linguagem de programação imperativa, como Pascal ou C, e um ambiente de desenvolvimento tradicional, como o Pascalzim ou o DEV-C++. Neste segundo semestre de 2011 e nesta turma, em especial, a disciplina estava sendo ministrada com a linguagem C e com o ambiente de desenvolvimento, Dev-C++.

Para a execução do estudo, os alunos matriculados na disciplina de Lógica de Programação foram divididos em dois grupos através da realização de um sorteio. Com o primeiro grupo foi adotada a abordagem alternativa e, por este motivo, ele foi nomeado grupo A. Com o segundo grupo foi adotada a abordagem tradicional e, por isso, ele foi nomeado grupo T. O grupo A continha 15 alunos, no entanto, os dados de apenas 13 deles puderam ser aproveitados. Destes 13 alunos, 11 eram homens e 2 eram mulheres. As idades dos alunos desse grupo variavam de 17 a 31 anos, com média de 22,23 anos. Nenhum deles tinha experiência com o Scratch nem tinha conhecimento sobre o conteúdo abordado na aula. O grupo T continha 14 alunos, todos homens. Neste grupo, as idades variavam de 18 a 36 anos, com média de 23,07 anos. Além disso, ainda no grupo T, três alunos haviam participado de disciplinas de programação anteriormente, dois deles porque estavam repetindo a disciplina e o último porque cursava o Curso Superior em Análise e Desenvolvimento de Sistemas em outra instituição de ensino. Destes três, apenas um deles conseguiu mostrar através de suas respostas que estava familiarizado com o assunto. Os outros 11 alunos nunca tinham estudado nada sobre o conteúdo abordado.

### **3.2. Execução**

O estudo foi executado em dois dias distintos. No primeiro dia, como os alunos participantes do grupo A não conheciam o ambiente Scratch, foi realizada uma ambientação neste ambiente. Logo após, o grupo A passou por uma aula da disciplina sobre o conteúdo *arrays* utilizando a abordagem alternativa. O conteúdo *arrays* foi escolhido por ser considerado um dos tópicos mais difíceis de serem ensinados por professores de disciplinas que envolvem programação [Dale, 2006].

Como era necessário que todos os alunos submetidos a este estudo passassem pela experiência com uma linguagem de programação mais tradicional não somente nos conteúdos abordados anteriormente na disciplina, mas também no conteúdo abordado durante o estudo, foi impossível deixar o grupo A de fora do segundo dia do estudo. Assim, neste segundo dia, o grupo T se juntou ao grupo A, e ambos assistiram a uma aula também sobre o conteúdo *arrays* ministrada na abordagem tradicional. Para garantir a consistência do estudo, as aulas ministradas nas duas abordagens seguiram o mesmo roteiro. No início de cada aula foi realizada uma avaliação para medir o nível de conhecimento que os alunos possuíam sobre o conteúdo abordado. Da mesma forma, ao final da execução de cada aula, foi realizada uma avaliação com o objetivo de medir o nível de aprendizado dos alunos com cada uma das abordagens.

#### **Ambientação com o Scratch com o Grupo A**

Neste primeiro momento, participaram apenas os alunos do grupo A. Como eles não tinham nenhuma experiência prévia com o Scratch, esta fase do estudo consistiu de uma ambientação com o ambiente Scratch. Nesta ambientação, foram abordados os seguintes tópicos: entrada e saída de dados, variáveis, operadores aritméticos e relacionais,

estruturas de repetição e estruturas condicionais. Estes tópicos já haviam sido abordados durante as aulas anteriores, contudo, utilizando-se apenas a abordagem tradicional. Assim, como os alunos não sabiam como eles deveriam realizar para utilizar tais estruturas na nova ferramenta, mostrou-se necessário apresentar como elas eram executadas utilizando o Scratch. Em seguida à aula propriamente dita, os alunos foram guiados por uma pequena lista de exercícios para que pudessem experimentar a ferramenta. A ambientação com o Scratch durou aproximadamente 1 hora e 15 minutos.

### **Aula seguindo abordagem alternativa com o Grupo A**

Logo após a ambientação com o Scratch, ainda no mesmo dia, ocorreu uma aula do conteúdo *arrays* seguindo a abordagem alternativa que utiliza o Scratch como ambiente e linguagem de programação. Esta etapa do estudo contou também apenas com a participação do grupo A que foi submetido às seguintes etapas: avaliação pré-aula, aula propriamente dita e avaliação pós-aula. A avaliação pré-aula teve como objetivo coletar o conhecimento prévio que os alunos possuíam a respeito do conteúdo *arrays*. A avaliação pós-teste, por sua vez, teve o intuito de avaliar o aprendizado dos alunos sobre o conteúdo ministrado através da abordagem alternativa. Para efeito de comparação das abordagens, os resultados obtidos na avaliação pós-aula da abordagem alternativa serão confrontados com os resultados obtidos na avaliação pós-aula da abordagem tradicional. Durante a aula propriamente dita, os seguintes tópicos foram abordados: definição e declaração de *array*, inserção e remoção de elementos, substituição de elementos, referência a um elemento, tamanho do *array* e verificação se um determinado elemento está contido no *array*. Nesta fase, como nenhum dos alunos tinha conhecimento sobre o assunto *arrays*, a avaliação pré-aula durou apenas 15 minutos. A aula foi ministrada em aproximadamente 1 hora e, posteriormente, foi disponibilizado em torno de 1 hora para que os alunos respondessem as avaliações pós-aula.

### **Aula seguindo abordagem tradicional com os Grupos A e T**

No dia seguinte, os grupos A e T participaram da aula, também sobre o conteúdo *arrays*, seguindo a abordagem tradicional que utilizou a linguagem de programação C e o ambiente de programação Dev-C++. Como toda a turma precisava passar pela linguagem de programação C, não foi possível deixar o grupo A de fora desta fase do estudo. Nesta aula, os grupos A e T foram submetidos a um procedimento semelhante ao da etapa anterior, ou seja, ambos os grupos passaram pelas etapas de avaliação pré-aula, aula propriamente dita e avaliação pós-aula. As avaliações realizadas pelo grupo T neste dia eram exatamente as mesmas que foram realizadas pelo grupo A na etapa anterior, ao passo que as avaliações realizadas pelo grupo A eram equivalentes no conteúdo, mas continham diferentes questões. Nesta aula, um dos participantes do grupo T já possuía algum conhecimento sobre o assunto *arrays*, por isso, a avaliação pré-aula teve duração maior do que a da aula anterior, de aproximadamente 35 minutos. A aula foi ministrada em aproximadamente 1 hora e 45 minutos e, posteriormente, também foi disponibilizado em torno de 1 hora para que os alunos respondessem as avaliações pós-aula.

### 3.3. Análise do estudo

Para medir o aprendizado dos alunos em ambas as abordagens, ao finalizar as aulas seguindo as diferentes abordagens, foi aplicada uma avaliação pós-aula que mensurou o conhecimento dos alunos em relação ao conteúdo que foi ministrado. Estas avaliações envolveram questões conceituais e de implementação sobre o assunto *arrays*. Logo após o estudo, as avaliações pós-aula foram corrigidas e, por possuir 12 questões, elas foram pontuadas com uma nota de 0 a 12. Estas notas, atribuídas às avaliações pós-aula, serviram então como dados para a análise estatística realizada neste estudo. As avaliações pré-aula, por sua vez, foram utilizadas apenas para verificarmos efetivamente se os alunos não possuíam conhecimento algum sobre o assunto abordado nas aulas. Apenas um aluno do grupo T conseguiu mostrar através das respostas da sua prova que estava familiarizado com o assunto.

Para a análise dos dados coletados, foi escolhida a distribuição  $t$  de Student, uma vez que o conceito de aleatoriedade foi adotado para a organização dos alunos nos dois grupos participantes do estudo. A distribuição  $t$  de Student é um teste estatístico usado para comparar os valores das médias entre os dois conjuntos de dados e analisar se a diferença entre eles é significativa. O nível de confiança adotado para a análise dos dados foi de 95%.

## 4. Resultados

Os dados apresentados na Tabela 1 referem-se às notas obtidas nas avaliações pós-aula pelo grupo A na aula que seguiu a abordagem alternativa e pelo grupo T na aula que seguiu a abordagem tradicional. Nota-se que a média das notas obtidas pelos participantes do grupo A foi maior do que a média das notas obtidas pelos participantes do grupo T. No entanto, os participantes do grupo A apresentaram um rendimento mais irregular do que os alunos do grupo T. Quando estes dados são analisados em conjunto, não se pode afirmar que a diferença entre as notas do grupo A e do grupo T é significativa. Consequentemente, nada se pode afirmar entre as abordagens alternativa e tradicional nesta situação.

**Tabela 1. Dados entre os grupos A e T nas abordagens alternativa e tradicional.**

	Média	Variância	$ t_0 $
Grupo A na abordagem alternativa	4.38	8.60	
Grupo T na abordagem tradicional	2.93	1.29	
			1.72

Os dados apresentados na Tabela 2 referem-se às notas obtidas nas avaliações pós-aula pelo grupo A nas aulas que seguiram as abordagens alternativa e tradicional. Nota-se que a média das notas obtidas pelos participantes do grupo A na aula que seguiu a abordagem alternativa foi ligeiramente maior do que a média das notas obtidas pelos mesmos participantes na aula que seguiu a abordagem tradicional. No entanto, na abordagem alternativa estes participantes apresentaram um rendimento mais irregular do que na abordagem tradicional. Semelhantemente ao primeiro caso, quando estes dados são analisados em conjunto, não se pode afirmar que a diferença entre as notas do grupo A nas duas abordagens é significativa para que se possa inferir uma diferença no grupo nas diferentes situações.

**Tabela 2. Dados do grupo A nas abordagens alternativa e tradicional.**

	Média	Variância	$ t_0 $
Grupo A na abordagem alternativa	4.38	8.60	
Grupo A na abordagem tradicional	4.03	4.95	
			0.33

Por último, na Tabela 3 são apresentados os dados referentes às notas obtidas pelos grupos A e T na aula que seguiu a abordagem tradicional. Observa-se que a média das notas obtidas pelos participantes do grupo A foi maior do que a média das notas obtidas pelos participantes do grupo T. Além disso, neste caso, as notas dos participantes dos grupos A e T também variaram de maneira pouco regular. Da mesma forma que para os casos anteriores, quando os dados obtidos foram analisados em conjunto, notou-se que não houve uma diferença significativa entre as notas destes grupos na abordagem tradicional.

**Tabela 3. Dados comparativos entre os grupos A e T na abordagem tradicional.**

	Média	Variância	$ t_0 $
Grupo A na abordagem tradicional	4.03	4.95	
Grupo T na abordagem tradicional	2.93	1.29	
			1.63

Os dados obtidos neste estudo não confirmaram a hipótese inicial apresentada neste trabalho. Contudo, é interessante comentar alguns dos resultados encontrados. Sabemos que os resultados apresentados por Hagam e Markham (2000) indicam que ter alguma experiência prévia em programação favorece o aprendizado da disciplina. Quando o grupo A passa duas vezes pela experiência de aula com o mesmo assunto, ele acaba possuindo uma experiência prévia com o assunto abordado. Analisando-se apenas a variância das notas do grupo A nas abordagens alternativa e tradicional, nota-se que houve um decréscimo da abordagem alternativa para a abordagem tradicional. Conseqüentemente, pode-se afirmar que houve uma maior regularidade nas notas do grupo A na abordagem tradicional do que na abordagem alternativa. Apesar de não haver uma diferença significativa entre as notas do grupo A nas duas abordagens, tal aumento na regularidade das notas sugere que uma investigação maior da estratégia que foi adotada com o grupo A deve ser realizada. Podemos assim descobrir que outras conseqüências ocorrem ao se iniciar uma disciplina introdutória de programação com uma abordagem alternativa seguida por uma abordagem tradicional não somente do ponto de vista das notas obtidas. Por exemplo, podemos analisar de que maneira o raciocínio lógico-matemático pode ser reforçado com uma abordagem alternativa antes da abordagem tradicional ou como o aprendizado de alguns assuntos considerados difíceis pelos alunos pode ser facilitado com este tipo de estratégia.

Além das notas obtidas, também podemos fazer algumas considerações sobre a maneira como as aulas foram ministradas. A aula seguindo a abordagem alternativa teve a duração de aproximadamente 1 hora, enquanto a aula seguindo a abordagem tradicional durou aproximadamente 1 hora e 45 minutos. Assim, pode-se perceber que a aula seguindo a abordagem alternativa fluiu mais rapidamente, por conta de características próprias da ferramenta Scratch, como por exemplo, comandos que podem ser facilmente combinados, programas que não geram erros sintáticos e linguagem de programação



escrita em português. Além disso, a interface gráfica do Scratch permite que o programa seja visualizado enquanto é executado. Consequentemente, a sensação que se tem é que a interface gráfica facilita a apresentação do conteúdo por parte do professor e também a assimilação desse conteúdo por parte do aluno, uma vez que se consegue exibir o funcionamento do programa na memória e na interação com o usuário. Por outro lado, a aula mais demorada na abordagem tradicional se deve principalmente à dificuldade de se repassar o conteúdo através de um código escrito e explicar o funcionamento deste código associando-o a uma representação gráfica sem o auxílio de uma ferramenta computacional. Além disso, erros de sintaxe e a linguagem de programação escrita em inglês, acabam tornando o aprendizado dos alunos mais penoso. Desta maneira, acreditamos que há uma influência positiva do uso do Scratch como ambiente de programação para iniciar um curso introdutório em programação, já que o processo de ensino-aprendizado parece ser facilitado em alguns aspectos. Posteriormente, o curso pode seguir seu fluxo normal com uma abordagem mais tradicional que utilize uma linguagem de programação mais comercial, como C ou Java, por exemplo. Como o estudo neste trabalho apresenta dados preliminares, posteriormente, pretendemos realizar novos estudos que analisem outros aspectos, tais como a utilização do Scratch do ponto de vista da motivação e participação dos alunos em sala de aula, por exemplo.

## **5. Conclusões e Trabalhos futuros**

Neste trabalho, apresentamos um estudo preliminar no qual avaliamos o ambiente introdutório de ensino Scratch como abordagem alternativa para o processo de ensino-aprendizagem de programação. Supomos que passando por uma abordagem alternativa como a proposta aqui, os alunos teriam um desempenho mais satisfatório do ponto de vista das notas obtidas do que passando por uma abordagem tradicional usando a linguagem de programação C e o ambiente de desenvolvimento Dev-C++. Entretanto, essa hipótese não pode ser confirmada com os dados obtidos neste estudo.

Ao invés disso, os resultados encontrados sugerem que seja realizada uma investigação mais detalhada das consequências que ocorrem ao se iniciar uma disciplina introdutória de programação com uma abordagem alternativa seguida por uma abordagem tradicional. Essa investigação poderá ser realizada não somente do ponto de vista das notas obtidas. Ela poderá analisar também como o raciocínio lógico-matemático dos alunos pode ser melhorado ou como o aprendizado de assuntos considerados difíceis pode ser facilitado com esta estratégia.

Como trabalhos futuros, realizaremos uma investigação mais detalhada da estratégia de ministrar uma disciplina introdutória de programação utilizando a abordagem alternativa seguida pela abordagem tradicional. A ideia é que o experimento seja executado em uma quantidade maior de aulas. Consequentemente, as abordagens poderão ser avaliadas através de um conteúdo programático mais extenso. Ademais, pretendemos investigar a utilização de outros ambientes de desenvolvimento para programação na linguagem C para compor a abordagem tradicional. Por fim, também pretendemos avaliar a utilização do Scratch do ponto de vista da motivação e da participação dos alunos em sala de aula.

## Referências

- Bennedsen, J.; Caspersen, M.E. (2009) Recalling Programming Competence, Proceedings do 9th Koli Calling International Conference on Computing Education Research, Koli National Park, Finland, p. 86-95.
- Byrne, P.; Lyons, G. (2001) The effect of student attributes on success in programming. ACM SIGCSE Bulletin, vol. 33, n. 3, p. 49-52.
- Cooper, S.; Dann, W.; Pausch, R. (2000) Alice: A 3-D tool for introductory programming concepts. Journal of Computing Sciences in Small Colleges, vol.15, n. 5, p.107-116.
- Dale, N. B. (2006) Most difficult topics in CS1: results of an online survey of educators, ACM SIGCSE Bulletin, vol. 38, n. 2, p. 49-53.
- Dev-C++ (2012) Disponível em <<http://bit.ly/14aDr0>>. Acesso em: 05 mai. 2012.
- Fincher, S.; Cooper, S.; Kölling, M.; Maloney, J. (2010) Comparing alice, greenfoot & scratch. Proceedings do 41st SIGCSE'10, Milwaukee, Wisconsin, USA. p. 192-193.
- Gomes, A.; Mendes, A. J. (2007) Learning to Program - Difficulties and Solutions. Proceedings do ICEE 2007, Coimbra, Portugal. Disponível em: <<http://bit.ly/Lj3HLu>>. Acesso em: 05 mai. 2012.
- Henriksen, P.; Kölling, M. (2004) Greenfoot: Combining object visualisation with interaction. Proceedings do 19th OOPSLA, Vancouver, Canadá. p. 73-82.
- Lahtinen, E.; Ala-Mutka, K.; Järvinen, H.-M. (2005) A Study of the Difficulties of Novice Programmers. Proceedings do 10th ITiCSE, Monte de Caparica, Portugal, p. 14-18.
- Malan, D. J.; Leitner, H. H. (2007) Scratch for budding computer scientists. Proceedings do 38th SIGCSE'07, Kentucky, USA, p. 223-227.
- Milne, I.; Rowe, G. (2002) Difficulties in Learning and teaching Programming - Views of Students and Tutors. Education and Information Technologies, vol. 7, n. 1, p. 55-66.
- Maloney, J.; Resnick, M.; Rusk, N.; Silverman, B.; Eastmond, E. (2010) The scratch programming language and environment. ACM Transactions on Computing Education, vol. 10, n. 4, article 16, 15 pages.
- Resnick, M.; Maloney, J.; Monroy-Hernández, A.; Rusk, N.; Eastmond, E.; Brennan, K.; Millner, A.; Rosenbaum, E.; Silver, J.; Silverman, B.; Kafai, Y. (2009) Scratch: programming for all. Communications of the ACM, vol. 52, n. 11, p. 60-67.
- Thomas, L.; Woodbury, J.; Jarman, E. (2002) Learning styles and performance in the introductory programming sequence. Proceedings do 33th SIGCSE'2002, Covington, USA. p. 33-37.
- Wilson, B. C.; Shrock, S. (2001) Contributing to success in an introductory computer science course: a study of twelve factors. ACM SIGCSE Bulletin, vol. 33, n. 1, p. 184-188.