

A Linguagem Scratch no Ensino de Programação: Um Relato de Experiência com Alunos Iniciantes do Curso de Licenciatura em Computação

Klissiomara L. Dias¹, Miquéias de L. Serrão¹

¹Instituto Ciberspacial – Universidade Federal Rural da Amazônia (UFRA)
Av. Presidente Tancredo Neves, Nº 2501 - Cep: 66.077-830 - Belém-Pará-Brasil

klissiomara.dias@ufra.edu.br, miqueiaslimaserrao@hotmail.com

Abstract. *This paper presents a report on the experience of the use of Scratch Language with a group of students of the first year in the Course of Computation Degree. The main objective was to identify the effects on this group as for the learning of the most elementary concepts of computer programming. Results suggest that there was a good educational outcome of the students as for the application of the central concepts of programming, what was measured through the analyses of the tests applied along the experiment and the projects developed by them, as well as the improvement of the interest level of this group to the learning of computer programming, measured by a survey after the experiment*

Resumo. *Este artigo apresenta um relato de experiência do uso da Linguagem Scratch com um grupo de alunos do primeiro ano de um curso de Licenciatura em Computação. O objetivo principal foi identificar os efeitos sobre este grupo em relação ao aprendizado dos conceitos mais elementares de programação de computadores. Os resultados sugerem que houve um bom aproveitamento dos alunos em relação à aplicação dos conceitos centrais de programação, o que foi medido por meio da análise de testes aplicados ao longo do experimento e dos projetos por eles desenvolvidos, bem como o aumento do nível de interesse desse grupo em relação ao aprendizado de programação, medido por meio de enquete realizada após o experimento.*

1. Introdução

Os altos índices de evasão e reprovação nos cursos da área da computação e informática é um problema enfrentado por vários países incluindo o Brasil.

Referente a dados relacionados ao Brasil, Freitas e Piva Jr (2011), com base nos dados do Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (INEP), relatam que dos 40.435 estudantes ingressantes em cursos de computação no ano de 2008, apenas 35,1% conseguiram concluir seus cursos.

A Universidade Federal Rural da Amazônia (UFRA) possui desde 2009 dois cursos nas áreas de computação e informática: Licenciatura em Computação e Bacharelado em Sistemas de Informação. Os alunos destes cursos são apresentados às disciplinas introdutórias de programação logo no primeiro ano de seu ingresso e, a

exemplo do acima exposto, é também uma realidade na UFRA os altos índices de reprovação e baixo rendimento dos alunos nessas disciplinas.

Visando investigar e testar alternativas para promoção de melhorias no processo de ensino-aprendizagem para iniciantes em programação de computadores, este trabalho apresenta parte dos resultados de um experimento realizado para avaliar os efeitos do uso do Scratch no ensino-aprendizagem de conceitos centrais de programação em um grupo de alunos calouros do curso de Licenciatura em Computação da UFRA.

As demais seções estão estruturadas da seguinte forma: a Seção 2 apresenta um resumo de trabalhos voltados ao tratamento deste tema, tendo o Scratch como linguagem base e, portanto, relacionados ao trabalho aqui proposto. A linguagem Scratch, com suas principais características, é apresentada na Seção 3, por ser a linguagem escolhida para o experimento aqui reportado. A Seção 4 apresenta o planejamento realizado para a execução do experimento proposto. Os resultados da execução deste experimento são apresentados na Seção 5 e na Seção 6 são realizadas as conclusões e propostas de trabalhos futuros.

2. O Scratch como Alternativa no Ensino-Aprendizagem de Programação

A disciplina introdutória de programação é de fundamental importância para os cursos das áreas de computação e informática, pois serve como base, direta ou indireta, para todas as demais disciplinas relacionadas.

É por meio das disciplinas introdutórias de programação que os alunos são apresentados aos principais fundamentos computacionais que norteiam o restante do curso. Por este motivo, o desempenho do aluno nesta disciplina é, na maioria das vezes, decisivo para sua evolução no curso. Não por acaso, esta é uma das disciplinas que os alunos mais apresentam dificuldades quando ingressam em um curso dessa natureza.

Malan e Leitner (2007) ressaltam que mesmo as estruturas de exemplos mais simples apresentadas logo no início de um curso introdutório de programação, como por exemplo, “*hello world*” se tornam um desafio para o aluno uma vez que este se preocupa em entender a parte sintática do programa desviando a atenção da lógica nele embutida.

Em um de seus trabalhos realizado na Universidade de Harvard, os autores citados utilizaram a linguagem Scratch como alternativa para iniciar a disciplina introdutória de programação, antes de partir para a introdução da linguagem Java. Para os autores, o Scratch é uma porta de entrada para os alunos aprenderem outras linguagens de programação como Java, Python, etc., pois através dela é possível trabalhar a lógica da programação sem a preocupação com perda de tempo e desvio de foco para resolução de erros de sintaxe, dificuldade presente na maioria das linguagens.

Segundo os autores, através de enquetes realizadas no final do curso para saber se a experiência com o Scratch facilitou o entendimento da linguagem Java, 76% dos alunos informaram que o Scratch influenciou positivamente. Enquanto 16% disseram que seu aprendizado não sofreu influência positiva ou negativa porque já tinham experiência anterior com programação.

Assim com o trabalho supracitado, a experiência aqui relatada também objetivou introduzir conceitos básicos de programação, por meio do Scratch, para alunos iniciantes em programação. Contudo, esta experiência foi realizada em paralelo ao andamento da

disciplina introdutória de programação regular do curso, como atividade extracurricular e no contraturno de aulas, apenas com um grupo de alunos da turma em questão, diferentemente dos autores citados, que utilizaram o Scratch como alternativa para a execução da primeira parte da disciplina introdutória de programação.

Aureliano e Tedesco (2012) também realizaram um experimento com o Scratch, com o objetivo de avaliar comparativamente o processo de ensino-aprendizagem de programação, especificamente do conceito de *arrays*, nas linguagens Scratch e C. Contudo, relatam que os resultados não permitiram comparar se houve melhoria no aprendizado de *arrays* com uma ou outra linguagem. O trabalho aqui proposto não tem por objetivo fazer comparações do aprendizado usando diferentes linguagens. Contudo, vai além no sentido de explorar as noções básicas de todos os principais conceitos elementares de um curso introdutório de programação, não se restringindo ao conceito de *arrays*.

Bittencourt et al. (2013) propuseram o Scratch como uma das ferramentas lúdicas utilizada em uma oficina de 4h de duração com alunos iniciantes do curso de Engenharia de Computação. Os autores propuseram desafios que visavam incrementar um jogo específico, à medida que os alunos exploravam de forma intuitiva os conceitos necessários para sua resolução. O trabalho aqui apresentado também propõe o uso de desafios como estratégia metodológica, mas não está restrito à sua aplicação no incremento e evolução de um único projeto. No nosso caso, os desafios propostos permitem soluções que podem ser representadas por meio de jogos, artes, animações, entre outros tipos de projetos.

Porque a experiência aqui relatada também foi realizada tendo como base a Linguagem Scratch, a próxima seção apresenta um breve resumo dessa linguagem.

3. A Linguagem Scratch

O Scratch é uma linguagem de programação desenvolvida pelo grupo de pesquisa *Lifelong Kindergarten*¹ do Laboratório de Mídias do Massachusetts Institute of Technology (MIT) e tem como objetivo principal a introdução da programação de computadores para usuários sem experiência em programação, com interesse especial em crianças (a partir de 8 anos) e adolescentes, mas não restrita a estes.

Em virtude do seu público alvo principal, a linguagem oferece um ambiente visual para programação (Figura 1-1), visando torná-la mais acessível que outras linguagens, disponibilizando uma interface que permite que programas sejam montados como blocos virtuais (Figura 1-2), arrastando-os para a área de trabalho (Figura 1-3) e montando-os (Figura 1-4), lembrando as peças de encaixar do brinquedo LEGO. A Figura 1 ilustra a interface do Scratch 2 (versão online), utilizado neste trabalho.

A linguagem Scratch disponibiliza comandos que permitem ao aprendiz trabalhar com conceitos computacionais importantes para iniciantes em programação de computadores, tais como entrada e saída, tipos de dados, variáveis, estruturas de controle, operadores e *arrays*. Além disso, também permite trabalhar com comandos que conferem a natureza multimídia inerente a esta linguagem.

¹ Acesso ao site grupo de pesquisa em <http://ilk.media.mit.edu/>.



Figura 1. Interface do Scratch 2 – versão *online*.

4. Planejamento do Experimento

O principal motivo que levou a escolha do Scratch como linguagem para este experimento foi o fato desta ter sido trabalhada pelos autores no Programa Institucional de Bolsas de Iniciação a Docência (PIBID)², programa existente desde 2012 na UFRA, em uma das ações que teve como objetivo introduzir a programação de computadores no ensino fundamental. Esta experiência anterior dos autores motivou o interesse em aplicar a mesma linguagem no ensino superior, reestruturando a proposta para este novo universo.

4.1. Seleção da Turma Participante do Experimento e Divulgação da Oficina

Porque era pré-requisito para este experimento envolver alunos sem experiência em programação, a turma selecionada correspondeu ao primeiro ano do curso de Licenciatura em Computação, alunos ingressos em 2013. A oficina foi planejada para acontecer no segundo período letivo daquele ano, momento em que esta turma estaria cursando a disciplina Algoritmos e Programação, disciplina introdutória de programação deste curso.

A divulgação da oficina se deu por meio da distribuição de folder explicativo, bem como convite feito em sala de aula e divulgação na lista de discussão da turma. As inscrições foram realizadas por meio de formulário *online* disponibilizado duas semanas antes do início da oficina.

A oficina foi planejada para ser realizada no contraturno de aulas, com a oferta de 20 vagas, correspondente à metade dos alunos matriculados naquela turma. Desta forma, a oficina foi ofertada aos aprendizes como uma atividade extracurricular e voluntária, com direito ao recebimento de certificados, condicionado à frequência mínima de 90% da carga horária.

² Programa financiado pelo Ministério da Educação (MEC), por meio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), que visa fomentar a formação inicial de professores para a educação básica.

4.2. Delimitação do Conteúdo da Oficina e Planejamento das aulas

Os autores optaram por trabalhar os principais conceitos apresentados pela ementa da disciplina introdutória de programação deste curso. Para tanto, oficina foi planejada com carga horária de 20 horas, distribuídas em seis aulas, ao longo de duas semanas consecutivas. As cinco primeiras aulas foram destinadas ao ensino dos conceitos centrais de programação e a última para a construção do projeto final.

A disciplina introdutória de programação deste curso é dividida em dois momentos. No primeiro, os alunos são apresentados aos conceitos básicos de algoritmos usando pseudocódigo (portugol ou português estruturado) tais como, tipos de dados, variáveis, estruturas de controle (decisão e repetição), *arrays* e subrotinas. Na segunda parte da disciplina os alunos são apresentados à linguagem C.

O experimento aqui relatado foi planejado para acontecer ainda no período em que a turma em questão estaria no estudo da primeira parte da disciplina regular do curso. Considerando o planejamento informado pela professora da disciplina, à época de sua execução, os alunos já teriam sido apresentados aos conceitos de tipos de dados, variáveis, operadores (lógicos, aritméticos e relacionais) e estruturas de decisão, usando pseudocódigo.

A organização das aulas e sua distribuição na carga horária são ilustradas na Tabela 1.

Tabela 1. Distribuição dos tópicos e carga horária da oficina Scratch

Aulas	Principais Comandos e Conceitos explorados	Carga horária
Aula 01	Introdução à Linguagem e Ambiente Scratch, Comandos de Movimento e Aparência, Tipos de Dados	2h
Aula 02	Comandos de Som, Caneta, Eventos e Sequência de Instruções	4h
Aula 03	Estruturas de Controle (decisão e repetição), Operadores e Comandos de Sensores	4h
Aula 04	Operadores (aritméticos, lógicos e relacionais) e Subrotinas	2h
Aula 05	Variáveis e Arrays Unidimensionais	4h
Aula 06	Desenvolvimento do projeto final	4h
Carga horária total		20h

Relativo ao plano de execução das aulas, o planejamento levou em consideração o método definido pelo *Scratch Curriculum Guide* (Brennan; Chung; Hawson, 2011), que apresenta uma abordagem de aprendizado baseada em projetos. No caso, tal como sugerido pelo guia, o passo a passo necessário para a execução das aulas também foi elaborado pelos autores e descrito em um plano de curso, de modo a torná-lo um padrão independente de quem fosse o facilitador em um dado momento.

Para cada uma das cinco primeiras aulas foram planejados e propostos em média 02 desafios. Os desafios, apesar de independentes entre si, foram elaborados de modo a forçar o aluno a aplicar conceitos vistos em aulas anteriores e não apenas os conceitos explorados na aula do desafio em questão. Cada desafio foi definido por meio de um enunciado explicando em linguagem natural o problema que o aluno deveria resolver.

A sexta e última aula foi planejada para que os alunos desenvolvessem o projeto final da oficina. Nesta aula, eles foram convidados a planejar e desenvolver projetos com

temática de interesse pessoal, buscando aplicar os conceitos aprendidos durante a oficina. Desta forma, diferentes naturezas de projetos poderiam ser desenvolvidas, tais como jogos, histórias interativas, animações, música, arte e simulação. De acordo com Papert (1980) ao explorar temas e naturezas de projetos de seu interesse, o aluno é capaz de expressar melhor os resultados de seu aprendizado.

Desta forma, os desafios e projeto final, foram definidos no experimento como mecanismos para acompanhamento e avaliação da aprendizagem dos alunos.

4.3. Instrumentos Adicionais para Avaliação do Experimento

Os autores definiram um questionário que foi aplicado uma semana antes do início da oficina, com o objetivo de identificar, entre outras coisas, se o aluno já tinha cursado alguma disciplina, oficina ou curso relacionado a Algoritmos e Programação, antes de seu ingresso na instituição, bem como sua motivação em relação ao tema.

Também foram definidos testes para avaliar a capacidade de resolução de problemas dos alunos (antes e depois da oficina) por meio de algoritmos usando pseudocódigo, até então, única forma conhecida por eles para expressar a resolução de problemas computacionais, conforme descrito na Seção 4.2. Os dois testes foram elaborados considerando enunciados diferentes, mas com equivalência em relação ao número de questões, conceitos explorados por cada questão e nível de complexidade, tendo sido aplicados imediatamente antes (pré-teste) e depois (pós-teste) da execução da oficina. A Tabela 2 ilustra a distribuição dos conceitos nas questões.

Tabela 2. Distribuição dos conceitos explorados nos testes (pré e pós testes)

Conceitos Abordados por Questão							
Questões	Operadores*	Entrada/Saída	Tipos de Dados e Variáveis	Estruturas de Decisão	Estruturas de Repetição	Vetores (Unidimensionais)	Subrotinas
Questão 1	X						
Questão 2	X	X	X				
Questão 3	X	X	X				
Questão 4	X	X	X				
Questão 5	X	X	X				
Questão 6	X	X	X	X			
Questão 7	X	X	X	X			
Questão 8	X	X	X	X			
Questão 9	X	X	X	X			
Questão 10	X	X	X	X			
Questão 11	X	X	X		X		
Questão 12	X	X	X	X	X	X	
Questão 13	X	X	X	X			X

*Lógicos, Relacionais e Aritméticos

As dez primeiras questões abordaram conceitos que todos os alunos, em tese, estariam habilitados a resolver antes do início da oficina. As três últimas questões exploraram conceitos ainda não vistos por eles na disciplina, à época de aplicação do pré-teste, e tiveram como objetivo identificar se algum aluno teria esse conhecimento prévio, independente da disciplina que estava cursando e que poderia ser confrontado com as respostas dadas no questionário de identificação do perfil dos alunos.

5. Resultados

5.1. Quanto à adesão dos alunos e background do grupo de experimento

Conforme exposto na Seção 4.1 foram ofertadas 20 vagas para a participação da turma selecionada, que consistia em 42 alunos matriculados na disciplina introdutória de programação do curso. Contudo, apenas 13 alunos se inscreveram, tendo 12 iniciado e concluído a oficina.

Dos 12 participantes, um deles já havia cursado disciplina introdutória de programação em outra instituição de ensino em curso de Licenciatura em Física, 07 teriam contato com uma linguagem de programação pela primeira vez durante este experimento e 04 afirmaram ter participado de oficinas de introdução à programação ofertadas no primeiro semestre daquele ano, durante a semana de calouros da instituição, que tiveram duração de 8h. Uma das oficinas foi ofertada pelo mesmo grupo dos autores e teve como objetivo apresentar uma visão geral da linguagem Scratch. A outra oficina apresentou a linguagem Logo. Dos 04 alunos citados, apenas 01 havia participado da oficina Scratch.

Desta forma, 05 dentre os 12 alunos participantes do grupo de experimento já tinham noções básicas de programação. O aluno que já havia cursando disciplina de programação em outra instituição havia creditado a disciplina correspondente no curso atual e os demais estavam todos matriculados e cursando a disciplina regular introdutória de programação.

A oficina contou com a participação de dois alunos do último ano do mesmo curso, que se revezavam na facilitação da oficina, atendimento e observação dos aprendizes e uma professora que esteve presente como observadora em todas as aulas.

5.2. Quanto ao Desempenho nos Testes de Algoritmos

A Tabela 3 ilustra as notas (de 1 a 13 pontos) obtidas pelos alunos do grupo de experimento nos testes aplicados antes e depois da execução da oficina:

Tabela 3. Notas do grupo de experimento antes e depois da oficina

Alunos participantes	Notas	
	Pré-Teste	Pós-Teste
aluno 1	4,5	7,4
aluno 2	5,9	6,7
aluno 3	10,7	12,0
aluno 4	7,9	7,9
aluno 5	8,7	12,2
aluno 6	6,6	12,2
aluno 7	5,1	3,4
aluno 8	0,8	12,2
aluno 9	9,6	10,6
aluno 10	6,6	8,1
aluno 11	8,0	11,1
aluno 12	3,6	3,4

Como pode ser observado, é possível identificar uma melhoria em relação ao desempenho da maioria dos alunos no pós-teste se comparado ao pré-teste. Para confirmar a significância destes resultados, os autores aplicaram o Teste T de *Student*

para amostras pareadas considerando uma significância de 5%. A Tabela 4 ilustra os resultados desta análise.

Tabela 4. Teste-t: duas amostras em par para médias

	<i>Antes</i>	<i>Depois</i>
Média (Xi)	6,5	8,933333333
Variância(Si ²)	7,594545455	10,83878788
Tamanho da amostra(n)		12
Média da Diferença (d)	-2,433333333	
Desvio Padrão da Diferença (S _d)	3,425129947	
Erro padrão da Diferença (Epd)	0,988749848	
t	2,461020184	
Hipótese da diferença de média (H ₀)		0
Graus de Liberdade (gl = n-1)		11
Nível de Significância (α)		0,05
p-value para t		0,031621453

A aplicação do teste de hipóteses permitiu verificar que é possível considerar que existe diferença estatisticamente significativa entre as médias do grupo de experimento antes e depois de terem participado da oficina Scratch. No caso, a média deste grupo após a intervenção se apresentou melhor que sua média antes da intervenção.

Analisando o desempenho individual por aluno, os que já tinham conhecimento prévio de programação não obtiveram nota significativamente superior, comparado às notas daqueles que estavam aprendendo uma linguagem de programação pela primeira vez. Pode-se inferir a partir disto, especificamente para o universo dos alunos deste grupo, que o Scratch permitiu aos alunos sem nenhuma base de programação anterior conseguir atingir níveis de desempenho similares ou superiores àqueles que já tinham.

5.3. Quanto aos desafios propostos

De um modo geral, a maioria dos alunos se mostrou curiosa, dedicada e participativa em sala de aula. Este resultado já era esperado considerando que os alunos desse grupo se voluntariaram para participar do experimento, o que demonstra interesse e motivação naturais.

Baseado na observação dos autores durante o processo criativo dos alunos, bem como posterior análise individual dos projetos por eles gerados, foi possível identificar que todos os conceitos computacionais planejados para serem explorados nos desafios propostos foram bem compreendidos e aplicados pelos alunos.

Apesar dos conceitos relativos a estruturas de repetição, *arrays* e subrotinas não terem sido trabalhados pelos alunos na disciplina introdutória de programação, à época da execução da oficina, eles demonstraram ter facilidade em compreender tais conceitos, uma vez que conseguiram aplicá-los nos desafios, a contento.

No que toca as estratégias utilizadas pelos alunos para a resolução dos problemas propostos, foi possível categorizá-los sob duas perspectivas: os mais objetivos e reativos e os que demonstraram mais proatividade e criatividade, estes últimos representaram 58,34% do total. Os alunos mais proativos se propunham desafios intermediários que consistiam em resolver o mesmo problema de diferentes formas e acabavam, com isso,

explorando na maioria das vezes, conceitos que ainda seriam vistos nas aulas seguintes. Os alunos caracterizados aqui como mais objetivos e reativos, focavam na resolução do problema, restringindo-se ao que foi solicitando.

5.4. Quanto ao Projeto Final

A maioria dos alunos (66,67%) escolheu desenvolver um jogo como projeto final, seguido de histórias interativas (16,67%), animações (8,33%) e simulações (8,33%).

Dentre os conceitos mais utilizados pelos alunos estão as estruturas de controle (repetição e decisão). Estruturas de decisão foram utilizadas em 91,67% do total de projetos, enquanto as estruturas de repetição foram exploradas em 83,33% deles. Em relação ao uso de variáveis, 50% dos projetos explorou este conceito.

Os conceitos explorados em menor proporção pelos alunos foram subrotinas e *arrays*. Apenas 16,67% dos alunos aplicaram estes conceitos em seu projeto final.

Relativo à complexidade dos projetos, os alunos que aplicaram o maior número de conceitos estudados durante a oficina, incluindo os conceitos menos explorados pelos alunos como um todo, no caso subrotinas e *arrays*, foram os alunos que tiveram seu primeiro contato com uma linguagem de programação durante este experimento.

Importante destacar que a abrangência dos conceitos explorados impactou diretamente no grau de complexidade dos projetos, visto que estes alunos superaram aqueles que já haviam tido o primeiro contato com uma linguagem de programação. Contudo, estes últimos também se destacaram, apesar terem desenvolvido projetos mais simples, demonstraram ter compreendido os conceitos abordados durante as aulas.

Em relação a aspectos negativos desta etapa da oficina, é importante destacar o tempo que foi destinado para o desenvolvimento do projeto final, de apenas 4h. Foi um tempo observado pelos aprendizes e pelos autores como sendo exíguo. Isto pode, em parte, ter contribuído para que alguns alunos deixassem de aplicar mais conceitos em seus projetos.

6. Considerações Finais e Trabalhos Futuros

Este artigo teve como objetivo avaliar os efeitos do uso da linguagem Scratch no ensino dos conceitos centrais de programação, em um grupo de alunos iniciantes de um curso de Licenciatura em Computação.

Os resultados deste trabalho sugerem que o Scratch pode ser uma alternativa para que os alunos que nunca tiveram contato com programação de computadores tenham uma primeira experiência menos traumática com uma linguagem de programação.

Em uma enquete realizada com o grupo de experimento ao término daquele período letivo, este aspecto também foi percebido pelos alunos, dando destaque em particular para um deles que relatou ter, depois de participado do experimento, desmistificado seu pensamento original de que programar é uma tarefa difícil. Este aluno teve contato pela primeira vez com uma linguagem de programação a partir do experimento apresentado aqui. Além disso, todos os alunos que responderam a enquete informaram ter aumentado o seu nível de interesse em programação de computadores

após sua participação no experimento e 83,34% deles afirmaram que o primeiro contato com a linguagem Scratch contribuiu para facilitar seu aprendizado em relação à linguagem de programação estudada por eles na disciplina introdutória de programação do curso, uma vez que a oficina Scratch foi concluída antes deles terem iniciado à segunda parte da disciplina em questão, onde seriam apresentados à linguagem C.

Contudo, estes resultados devem ser recebidos com ressalva, visto que o grupo de experimento participou de forma voluntária da oficina proposta, o que provavelmente pode ter contribuído para este cenário favorável uma vez que confere a estes uma motivação natural pelo tema da oficina. Desta forma, os autores não podem afirmar que se o grupo de experimento fosse aleatório e não voluntário esses resultados se repetiriam.

Ainda relativo à enquete citada, em relação à proposta metodológica utilizada, todos os alunos informaram ter sido muito interessante, em particular porque os desafios aguçavam sua curiosidade e motivação para resolver o problema e, com base nisto, viabilizavam a construção do seu conhecimento. Além do fato de estimular uma competição saudável entre eles, ainda que involuntária.

Como trabalho futuro os autores propõem que este experimento seja repetido, porém em grupos não voluntários e aleatórios, para que os resultados dessa experiência possam ser confrontados com sua execução em outros contextos de grupo. Além disso, serviria também ao propósito de testar mais e melhorar a estratégia de ensino proposta. Também é sugerido estabelecer grupos de controle para fins de análises estatísticas de desempenho comparado aos grupos de experimento.

Referências

- Freitas, R. L.; Piva Jr, D.; Estratégias para melhorar os processos de Abstração na disciplina de Algoritmos. In: XIX Workshop sobre Educação em Computação, Natal, RN, 2011.
- Malan, D. J., Leitner, H. H. Scratch for Budding Computer Scientists. In: Special Interest Group on Computer Science Education, Kentucky, EUA. Mar. 2007.
- Aureliano, V.; Tedesco P. Avaliando o uso do Scratch como abordagem alternativa para o processo de ensino-aprendizagem de programação. In: XX Workshop sobre Educação em Computação, Curitiba, Paraná, 2012.
- Bittencourt, R. A., Rocha, A. S., Santana, B. L., et al. Aprendizagem de Programação Através de Ambientes Lúdicos em um Curso de Engenharia de Computação: Uma Primeira Incursão WEI 2013. In: XXI Workshop sobre Educação em Computação, Maceió, AL, 2013.
- Brennan, K.; Chung, M, Hawson, J. Scratch Curriculum Guide Draft. Nov. 2011. Disponível em: <<http://scratched.media.mit.edu/resources/scratch-curriculum-guide-draft>>. Acessado em: 28 de dezembro de 2013.
- Papert, S. Mindstorms: Children, Computers and Powerful Ideas. Cambridge, Massachusetts. Abril, 1980.