



UTILIZAÇÃO DO SCRATCH COMO FERRAMENTA DE AUXÍLIO À APRENDIZAGEM DE PROGRAMAÇÃO

Anderson Souza Rocha – andersoncomp091@gmail.com

Jadson Francisco de Jesus Silva – jadsonecomp@gmail.com

Douglas Almeida Carneiro – douglas.ecomp12@gmail.com

Jéssica Magally de Jesus Santos - jmagally.ecomp@gmail.com

Bianca Leite Santana - biancasantana.ls@gmail.com

Roberto Almeida Bittencourt – roberto@uefs.br

Universidade Estadual de Feira de Santana

Av. Transnordestina, s/n – Novo Horizonte

44.036-900 – Feira de Santana – BA – Brasil

Resumo: *Este artigo apresenta um trabalho desenvolvido na Universidade Estadual de Feira de Santana que objetiva facilitar o aprendizado de programação de computadores por parte dos estudantes ingressantes do curso de Engenharia de Computação da instituição, assim como diminuir os índices de retenção das disciplinas vinculadas à área de programação. Para tanto, a ferramenta Scratch foi utilizada como um ambiente lúdico de aprendizagem e os conceitos básicos de programação foram trabalhados através de uma oficina oferecida no início do semestre letivo.*

Palavras-chave: *ambientes lúdicos de aprendizagem, aprendizagem de programação, Scratch, ferramentas.*

1. INTRODUÇÃO

A aprendizagem de programação de computadores pode ser considerada como um dos desafios enfrentados por estudantes de tecnologia da informação, e também de engenharias em geral ou qualquer outro curso que utilize técnicas de programação para a solução de problemas. Dentre os fatores que dificultam a absorção dos conceitos inerentes à prática da programação, podemos citar a quantidade de conteúdos e habilidades que um estudante necessita aprender/desenvolver para iniciar essa atividade e também a dificuldade em dominar a *Integrated Development Environment* (IDE) de programação, a qual muitas vezes apresenta um ambiente pouco intuitivo e visualmente rústico. Tais conceitos vão desde raciocínio lógico e álgebra, até conhecimentos de projeto de sistemas, estruturas de dados, projeto de algoritmos, sintaxe e semântica de linguagens de programação, entradas e saídas de sistemas, manipulação de arquivos, paradigmas de programação, entre outros. Normalmente, a maioria dos conteúdos e habilidades citados anteriormente é ministrada aos discentes em um ou dois semestres, durante o período de graduação.



Para cursos na área de computação o período de aprendizagem da programação é bastante intenso e, devido à quantidade de conteúdos e a profundidade com que eles são abordados para estudantes desta área, esta etapa da graduação se torna desgastante e complexa, ainda que seja primordial para a boa formação profissional deste estudante. Além da quantidade de conteúdos que os alunos têm que aprender, e habilidades a desenvolver em curtos períodos de tempo, Kölling e colegas citam como outra barreira bastante complicada de ser transposta, a mudança de paradigma de programação, de um paradigma imperativo para um orientado à objetos (KÖLLING *et al.*, 1995).

Outra questão bastante importante no aprendizado de programação é a motivação. Há uma grande disparidade entre os programas desenvolvidos por estudantes durante sua graduação e os softwares cada vez mais complexos que eles utilizam (muitas vezes desde sua infância), e tal fato pode levar o estudante à desmotivação por causa da excessiva simplicidade dos programas que os mesmos são capazes de desenvolver com o restrito conjunto de competências que possuem principalmente no início da graduação em computação, frente aos softwares disponíveis no mercado. Um agravante desmotivador é a metodologia aplicada para ensino de programação, geralmente consistindo de aulas expositivas ministradas por um professor, que geralmente adota uma postura de entidade central para transmissão de conhecimentos (RAPKIEWICZ *et al.*, 2006), tornando assim mais difícil despertar o interesse do estudante para a resolução dos problemas propostos, normalmente excessivamente abstratos. Além disso, muitas vezes estes problemas são considerados obsoletos frente à realidade dos estudantes enquanto usuários de sistemas computacionais.

Para grande parte dos estudantes de cursos de computação, outro entrave para o aprendizado de programação está intimamente ligado à falta de conexão entre suas experiências durante a educação básica, e as habilidades e competências necessárias para a prática da programação. O que torna o processo de aprendizagem ainda mais difícil visto que a maioria dos conceitos a serem assimilados são totalmente novos.

A ocorrência deste e dos outros problemas citados anteriormente se reflete nos altos índices de evasão e abandono destes cursos. Segundo uma pesquisa do Observatório SOFTEX, há uma baixa ocupação das vagas disponíveis e alta evasão nos cursos da área de tecnologia da informação (TI) (SOFTEX, 2012). Dados desta pesquisa informam que, em 2008, havia 198.904 vagas para cursos em TI. Estas vagas foram preenchidas por 91.448 ingressantes, indicando o aproveitamento de apenas 46% das vagas disponíveis. Além disso, neste mesmo estudo é apontada uma evasão de 21% no ano de 2008. Estes fatos contribuem para o surgimento de um déficit de profissionais nesta área, e sendo previsto que até 2012 este déficit tenha atingido o número de 116.362 profissionais. No curso de Engenharia de Computação da Universidade Estadual de Feira de Santana (UEFS), foi observado, de 2009 até 2011, um índice de evasão de 20%, embora alguns dados preliminares apontem para o aumento desses índices. Observa-se que a retenção em disciplinas obrigatórias de introdução à programação gira em torno de 50 a 75%. Em 2012, foi verificado que para 40 ingressantes no curso, por semestre letivo, há cerca de 60 a 70 alunos matriculados em introdução à programação no primeiro semestre, e o componente curricular do segundo semestre, que trata do paradigma de programação orientado à objetos (POO), teve 50 a 60 alunos matriculados por período.

Neste trabalho, é apresentada uma proposta para minorar estes problemas, baseada na aplicação de metodologias ativas de aprendizagem, combinada com o uso de ambientes lúdicos próprios para o ensino de programação para jovens. Busca-se ensinar a programação através do desenvolvimento de jogos e outras atividades divertidas,



visando fortalecer a motivação do estudante. Além disso, a complexidade também pode ser reduzida através da utilização de ambientes voltados especificamente para o ensino de programação para jovens e/ou iniciantes, onde o principal objetivo é entender e praticar a lógica de programação. Para tanto, utilizamos uma ferramenta bastante intuitiva para facilitar os primeiros contatos com a programação e com o desenvolvimento do processo criativo. Por fim, através da utilização de uma metodologia de aprendizagem centrada no aluno, o estudante assume um papel ativo no seu próprio processo de aprendizagem, permitindo que cada estudante possa aprender de acordo com o seu próprio ritmo e interesse.

Em um projeto piloto, foi realizada, na primeira semana do primeiro semestre letivo de 2013, uma oficina de aprendizagem de programação utilizando como ambiente de desenvolvimento a ferramenta Scratch. Foram explorados conceitos básicos de programação para os ingressantes do curso de Engenharia de Computação da UEFS, usando como objeto de aprendizagem o desenvolvimento de um jogo. Os resultados desta iniciativa foram interessantes, principalmente pela motivação demonstrada pelo público, o que culminou explicitamente numa maior facilidade de aprendizado.

2. TRABALHOS RELACIONADOS

O quadro de evasão e retenção dos estudantes nas disciplinas introdutórias de computação, como algoritmos e linguagens de programação, vem provocando o surgimento de várias iniciativas que objetivam tentar facilitar o aprendizado dos alunos ingressantes nesta área. Estas iniciativas geralmente envolvem mudanças de paradigmas na metodologia de ensino e aprendizagem e/ou a introdução de algum ambiente lúdico de aprendizagem de programação, como o próprio Scratch.

Dentre as experiências prévias de utilização do Scratch como ferramenta para reduzir as barreiras para a aprendizagem de programação, está o trabalho desenvolvido pelo Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE). Neste projeto, eles utilizaram a ferramenta Scratch na disciplina de Algoritmos e Lógica de Programação e uma metodologia onde os alunos desenvolvem pequenos programas com o objetivo de estimular o seu raciocínio lógico, conceitos matemáticos e outros vinculados à computação (PEREIRA *et al.*, 2012). Segundo os autores, os alunos que usaram a ferramenta ganharam uma melhor compreensão de conceitos de programação tais como estruturas de repetição e decisão, variáveis e operadores.

Outro trabalho correlato é o desenvolvido pelo Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina (IFSC), utilizando o Scratch no ensino introdutório de programação. A ferramenta foi utilizada neste projeto para o aprendizado dos conceitos de programa, estruturas de decisão e repetição, variáveis, operadores e outros (MÉLO *et al.*, 2011). Segundo os autores, os alunos se sentiram bem mais motivados utilizando o Scratch em seu processo de aprendizagem.

O Scratch também já foi utilizado para o ensino de programação em cursos oferecidos na Educação a Distância (EaD) (AURELIANO e TEDESCO, 2013). Neste trabalho, o Scratch foi utilizado em uma disciplina introdutória de programação, no caso, Lógica para Programação. Os autores utilizaram o Scratch no início da disciplina, de modo a contribuir para um melhor entendimento dos conceitos iniciais de programação, e depois o curso seguiu o fluxo normal da disciplina. A avaliação dos resultados desta iniciativa, segundo os autores, foi obtida através de questionários e fóruns envolvendo os alunos onde os alunos mostraram divididos em relação à eficácia desta iniciativa.



Outras ferramentas, além do Scratch, já foram utilizadas como ambientes lúdicos de ensino e aprendizagem em programação. O ambiente de programação Alice foi utilizado na Escola de Engenharia da Universidade Presbiteriana Mackenzie (EE/UPM) no ensino de programação. Neste trabalho os autores citam o desenvolvimento do Projeto Alice na EE/UPM e demonstram as possibilidades da utilização do Alice, especialmente nas disciplinas de ensino de linguagens de programação orientadas a objetos (BARROS *et al.*, 2012).

Além da introdução de ambientes lúdicos de ensino de programação, estratégias pedagógicas também são utilizadas para diminuir a evasão e retenção nas disciplinas algoritmos e linguagens de programação. O uso de jogos numa fase introdutória do ensino dessas disciplinas, utilizando recursos computacionais e gráficos, é considerado um recurso pedagógico que possibilita minimizar as dificuldades como desmotivação dos alunos. A desmotivação costuma ocorrer num contexto onde se resolvem exercícios, em sala de aula ou fora dela, tornando monótono o desenvolvimento da lógica e dos algoritmos. Jogos promovem a possibilidade de o aluno construir o seu conhecimento nesta área por meio de ações concretas, experimentações e reflexões (RAPKIEWICZ *et al.*, 2006). Rapckiewicz e colegas (2006) procuraram utilizar uma forma de representação da solução de modo que todos os envolvidos no processo entendessem, sem utilizar uma linguagem pré-definida para tal como o pseudocódigo; ao contrário, a forma de representação da solução era resultado de interações entre os participantes do grupo, que estavam aprendendo juntos, numa abordagem colaborativa.

Uma experiência desse tipo é descrita em *Jogo Baralho das Variáveis* (KAHWAGE *et al.*, 2013). Neste trabalho, foi desenvolvido um jogo para ser utilizado no estudo dos conceitos iniciais dos tipos de variáveis e suas operações, nas disciplinas de programação do curso de graduação em Sistema de Informação da Universidade Federal do Pará – UFPA. Este trabalho objetivou minimizar a problemática do ensino-aprendizagem e aumentar o interesse e motivação dos alunos através do apelo que os jogos exercem nos estudantes. Com base nos resultados obtidos, os autores destacam a necessidade de utilizar estes jogos no processo de aprendizagem em consonância com o trabalho do professor em sala de aula, possibilitando a geração de melhores resultados.

3. AMBIENTE LÚDICO DE APRENDIZAGEM

O Scratch é um ambiente gráfico de programação desenvolvido pelo *Lifelong Kindergarten Group*, do MIT, inspirado em linguagens de programação como o Logo, e que permite o desenvolvimento de aplicativos que integram recursos de multimídia de forma intuitiva. Seu objetivo principal é facilitar a introdução de conceitos de matemática e de computação, enquanto também induz o pensamento criativo, o raciocínio sistemático e o trabalho colaborativo (MIT, 2013).

A proposta do projeto é levar o estudante a um processo de aprendizado contínuo, num ciclo que envolve criação, experimentação, compartilhamento e reflexão, resultando na expansão de suas habilidades, conhecimento e competências (RESNICK, 2009). O Scratch é um software gratuito que possui um ambiente de desenvolvimento que não exige que o usuário digite comandos e funções. Em lugar disso, ele usa uma linguagem icônica muito semelhante aos blocos construtivos de brinquedos Lego. A tela inicial do Scratch pode ser vista na Figura 1.

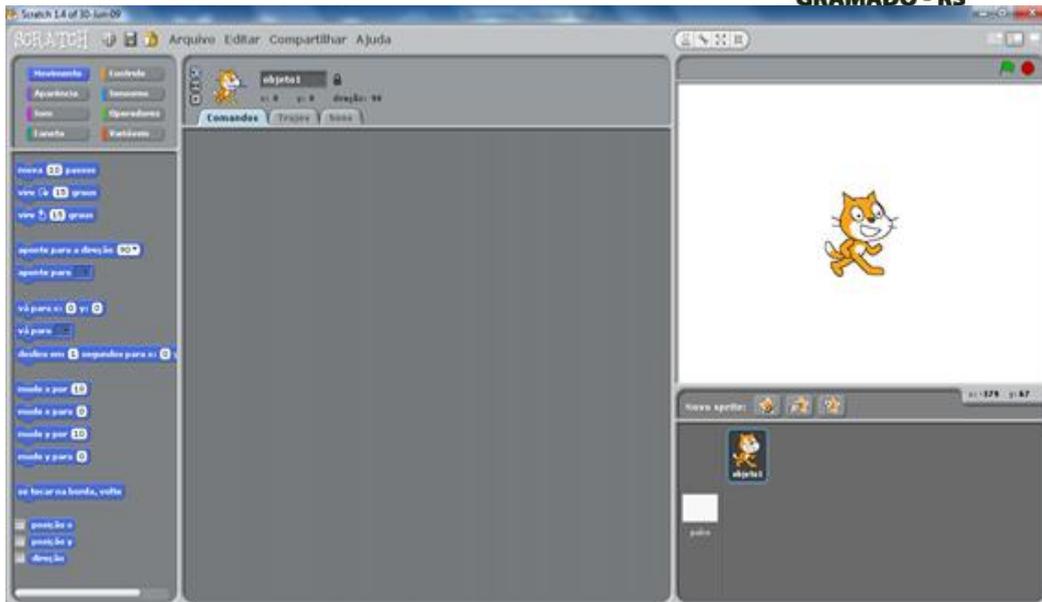


Figura 1 – Tela Inicial do Scratch

Inspirado na filosofia dos brinquedos Lego e em uma pedagogia socioconstrutivista, Scratch foi projetado para atender a três princípios básicos de design: ser mais manipulável, mais significativo e mais social. A gramática do Scratch possui um conjunto de blocos de programas similares a tijolos Lego, onde os blocos de comando se encaixam uns nos outros para fazer o programa. Isto permite aos aprendizes montar scripts para objetos através de manipulação de blocos de controle e de entrada e saída, dentre outros. O encaixe destes blocos só ocorre de uma única maneira, o que proporciona ao usuário a noção da estrutura sintática de um programa de computação.

Com o Scratch, o usuário pode criar os mais diversos projetos, como jogos, histórias, animações, o que evidencia o grande poder desta ferramenta. Na confecção de seus projetos, os usuários acabam manipulando implicitamente importantes conceitos de programação como estruturas de repetição e seleção, entrada e saída, comunicação entre objetos, variáveis e operadores. Os projetos desenvolvidos no Scratch podem ser disponibilizados em uma rede social própria, disponível em: <http://scratch.mit.edu/>, que funciona como uma comunidade de compartilhamento, cooperação e socialização, onde os usuários criam discussões e repassam suas experiências e aprendizados. Atualmente, a comunidade Scratch já conta com mais de três milhões de projetos compartilhados. A rede social do Scratch possui uma seção de criação com um editor que possibilita o desenvolvimento de projetos online. Esse editor possui as mesmas características de interface da ferramenta Scratch para desktop e apresenta algumas funcionalidades diferenciadas como integração com a câmera do computador para criação de *sprites*.

4. A OFICINA DE SCRATCH

4.1. Planejamento

A oficina ministrada para os alunos ingressantes do curso de Engenharia de Computação teve como objetivo principal a apresentação de alguns conceitos básicos de programação através de uma metodologia ativa e lúdica, e, além disso, introduzir novos conceitos para muitos dos participantes, que até então nunca tinham tido experiências



concretas de programação em qualquer linguagem. Outro objetivo também visado por esta iniciativa foi a redução da evasão e retenção dos alunos nas primeiras disciplinas de programação do curso de Engenharia de Computação da UEFS.

Esta oficina teve uma carga horária total de 4 horas, contando com 26 participantes, e foi facilitada por 6 estudantes, contingente necessário para atender às demandas dos alunos, sendo 2 tutores e 4 monitores, contando ainda com um professor para observação e avaliação da metodologia aplicada, bem como a resposta dos estudantes à oficina.

Em resumo, a metodologia utilizada consistiu em tentar fazer os alunos aprenderem brincando. Os tutores elaboraram desafios e os propuseram aos participantes da oficina. Para solucioná-los, eles utilizaram os recursos disponíveis no Scratch, recebendo sempre auxílio dos monitores quando do surgimento de dúvidas ou de dificuldades de compreensão do problema proposto.

Foi preparada uma apresentação do ambiente Scratch e sua comunidade a fim de despertar o interesse dos alunos para continuarem explorando a ferramenta após a oficina. Como o tempo de duração previsto para a oficina não era suficiente para garantir a passagem por todos os conceitos básicos de programação de maneira convencional, como em uma aula expositiva, a oficina foi planejada de maneira que os conteúdos básicos fossem abordados de maneira indireta. Pensando nisso, parte da oficina foi baseada no desenvolvimento de um jogo que ficou famoso por estar presente no Windows 95, o jogo *Arrow and Bow*. A versão original do jogo baseia-se num arqueiro que tem como missão estourar balões ou qualquer outro objeto flutuante, tendo um limite determinado de flechas para cumprir esse objetivo.

O andamento da oficina se deu de forma muito similar como uma atividade tutorial, onde os discentes tinham um objetivo, mas para alcançá-lo deveriam aplicar/utilizar exclusivamente aquilo que lhe fosse conveniente e não soluções prontas concebidas pelos tutores/monitores. Assim, procurou-se não interferir em excesso no processo de aprendizagem de cada aluno, estimulando sua criatividade, e dando-lhe liberdade para questionar e até mesmo conceber outras possíveis soluções.

Para avaliar a oficina, bem como a resposta dos estudantes, o observador utilizou um protocolo de pesquisa qualitativa, anotando os eventos da oficina e de seus participantes, complementando-as com reflexões sobre as atividades decorridas. Os tutores e monitores também realizaram uma avaliação em forma de reflexão acerca da oficina e da metodologia aplicada. Os participantes também foram ouvidos, através do convite para responder a um questionário após a oficina, indicando pontos positivos e negativos, assim como possíveis sugestões para próximas oficinas.

4.2. Resultados

A oficina foi iniciada com a apresentação da ferramenta Scratch, e também do seu ambiente colaborativo na internet (rede social do Scratch), após a apresentação, foi proposto aos estudantes o desenvolvimento do jogo *Arrow and Bow*.

Apesar de, inicialmente, a maioria dos alunos não possuírem conceitos de programação, intuitivamente eles já se mostraram capazes de criar cenários e objetos para o jogo proposto. As dúvidas dos participantes começaram a aparecer, especialmente em relação a como construir os scripts para cada objeto do jogo, aplicar as estruturas de seleção, controle e repetição. Para interferir somente o necessário, e não retirar a liberdade do estudante, assim como seu papel ativo em seu processo de aprendizagem (uma das propostas da oficina), os tutores deram curtas e pontuais



explicações, cabendo assim aos monitores realizarem o acompanhamento individual dos alunos, bem como sanar suas dúvidas, principalmente as relativas à manipulação da ferramenta.

Foi possível observar que, apesar de o jogo proposto ser um desafio mais geral, cada estudante esteve livre para avançar mais rápido ou caminhar mais lentamente no desenvolvimento do jogo. Alguns conseguiram rapidamente introduzir cenários diferentes, movimentação nos objetos e até a utilização de conceitos de programação para solução de um problema (e.g., um estudante conseguiu fazer um balão furado pela flecha disparada do arco ser reconstituído ao estado inicial após um dado tempo, aplicando estruturas de seleção e laços).

Os comandos disponíveis no Scratch mais utilizados pelos estudantes foram: controle, aparência, sensores, operadores. Alguns não utilizaram as variáveis, embora uma grande parte dos participantes tenha feito tal uso para controle de quantidade de flechas disponíveis para disparo e indicação do fim do jogo. Acredita-se que o motivo para isso tenha sido devido ao fato de quem nem todos tiveram o mesmo ritmo, nem os mesmos interesses ao longo do desenvolvimento. As maiores contribuições dos monitores para sanar as dúvidas dos participantes foram em explanações acerca de sensores e operadores e, apesar da aparente maior complexidade, as estruturas de controle foram compreendidas de forma mais fácil pelos alunos.

A grande maioria dos alunos conseguiu concluir todas as metas da oficina. Alguns foram além e introduziram funcionalidades ao jogo que não foram propostas inicialmente, o que foi considerado como um resultado positivo e reflexo direto da utilização da metodologia ativa de aprendizagem. Percebeu-se, dentre os alunos que tiveram o primeiro contato com a prática de programação na oficina, uma maior dificuldade em visualizar a ordem correta para a introdução de comandos, bem como a tradução da lógica para código. Mas, com o auxílio dos monitores, esta barreira foi total ou parcialmente transposta.

Através do desenvolvimento deste jogo, os alunos puderam trabalhar estruturas condicionais, de repetição, leitura de sensores, saída de áudio e aparência, além de movimentos, operadores lógicos e relacionais, manipulação de variáveis, e o desenvolvimento das habilidades de compreensão de requisitos e de expressão de possíveis soluções em uma forma algorítmica. Através dos recursos presentes no Scratch e do jogo proposto como desafio, a oficina buscou trabalhar com os participantes, mesmo que de forma intuitiva, conceitos de dois paradigmas de programação: procedimental e orientada a objetos. Da programação procedimental, os estudantes utilizaram a execução sequencial dos comandos, a organização das estruturas de seleção e repetição em blocos, e a decomposição do programa em funções. Da programação orientada a objetos, os alunos utilizaram conceitos de objetos (e.g., encapsulamento) e de comunicação entre objetos, já que os objetos participantes do programa possuem seus próprios comandos e rotinas, e enviam mensagens para outros objetos. Assim, os alunos puderam entrar em contato com os dois paradigmas de programação, compreendendo sua complementaridade. Este resultado contrasta com a dificuldade típica que normalmente surge quando da transição de um paradigma para o outro ao longo dos estudos em um curso de graduação.

Outro aspecto interessante da metodologia utilizada na oficina foi a liberdade para criação. Mesmo tendo um jogo como base, os tutores e monitores da oficina provocaram os alunos para que eles criassem o jogo através de sua própria forma de compreender, analisar e capturar os requisitos de cada funcionalidade, e explorassem extensivamente a ferramenta Scratch para a implementação destes requisitos, tanto nos



aspectos de interface com o usuário (*sprites*, cenários, músicas) como na implementação propriamente dita das funcionalidades de forma personalizada por cada estudante.

Visando obter *feedback* dos participantes da oficina, foi elaborado um curto questionário com perguntas gerais acerca da oficina, seu andamento, aproveitamento para as disciplinas que iriam cursar, carga horária da oficina, dentre outras. Quando questionados se gostaram da oficina, a resposta foi praticamente unânime, como na resposta de um dos participantes: “Curti muito, achei simples, e dá uma boa noção de lógica de programação”. Também foram levantadas questões sobre pontos negativos e positivos da oficina. Em geral, os estudantes responderam que o que mais gostaram da oficina foram às noções de lógica de programação que eles obtiveram. Sobre o que menos gostaram, eles mencionaram o tempo curto da oficina e o exíguo conteúdo. Muitos demonstraram muito interesse em participar de um curso mais extenso sobre o Scratch e se aprofundar mais na ferramenta.

As reflexões e observações listadas pelos tutores e monitores em geral convergiram para um consenso de que a oficina acabou superando as expectativas da equipe executora, superando inclusive um temor de que ocorresse um desinteresse dos participantes em desenvolver programas em uma plataforma desenvolvida para crianças e adolescentes e que apresenta, em sua interface gráfica, um aspecto infantil. Além disso, para os tutores e monitores, os objetivos foram alcançados, pois o desafio proposto permitiu aos alunos desenvolverem as competências esperadas e os conceitos de programação pretendidos. Embora de uma forma mais superficial, os laços, estruturas condicionais, e variáveis foram aprendidos.

Outro efeito também observado pelos tutores após realização da oficina foi a redução nas taxas de reprovação dos estudantes ingressantes no curso de Engenharia de Computação na disciplina de Algoritmos e Programação I, conforme podemos observar na Figura 2.

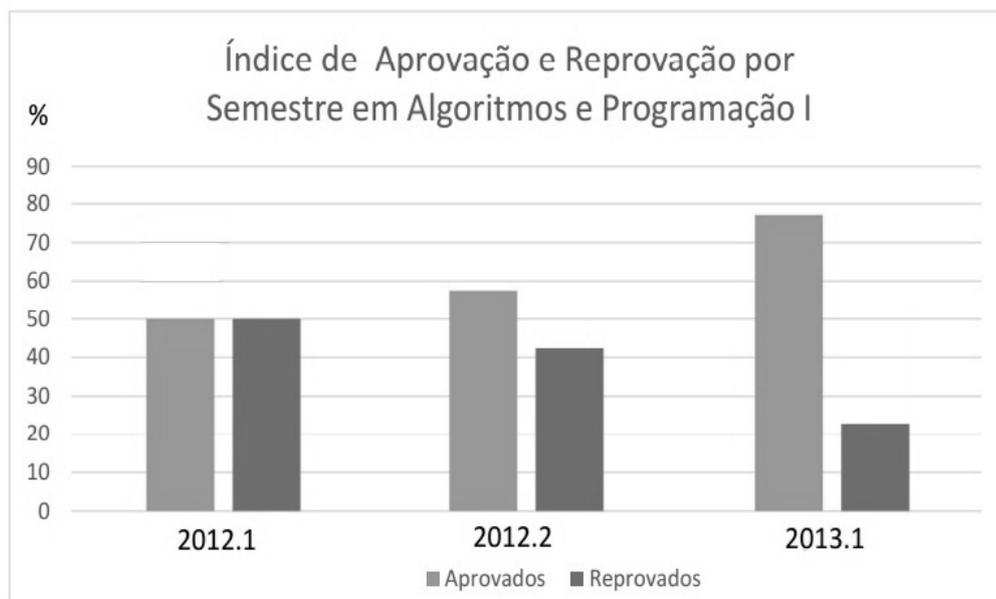


Figura 2 – Gráfico comparativo de aprovação e reprovação de estudantes ingressantes no curso de engenharia de computação.

Nos períodos de 2012.1 e 2012.2 (em que a oficina ainda não tinha sido realizada), o curso de Engenharia de Computação da UEFS sofria com índices de reprovação em



Algoritmos e Programação I de respectivamente de 50% e 42%, que podem ser considerados muito altos, já que esta disciplina é considerada básica no curso, e é o primeiro contato do aluno com a prática da programação. Após a execução da oficina, no período 2013.1, observa-se na Figura 2 uma redução significativa na reprovação e evasão dos alunos, em que este índice caiu de 42% no semestre anterior (2012.2) para 33% em 2013.1. Tal redução sugere que a oficina realizada pode ter contribuído no aprendizado da programação para os alunos, reduzindo a dificuldade que os mesmos enfrentam ao entrar em contato com os conceitos de programação, o que tem impacto na melhora de seus desempenhos. É importante salientar que a redução da evasão e reprovação dos alunos não é consequência somente da realização da oficina. Fatores como mudança de professores ministrantes da disciplina, mudança de metodologia aplicada, modificações nos trabalhos propostos para os estudantes, e até mesmo as características inerentes aos alunos ingressantes, podem influenciar estes índices. Entretanto, o resultado positivo obtido dessa experiência encoraja novas realizações desta oficina como uma ferramenta auxiliar para o aprendizado dos alunos.

5. CONSIDERAÇÕES FINAIS

Este trabalho apresentou uma primeira incursão da utilização da ferramenta Scratch para os estudantes ingressantes no curso de Engenharia de Computação da UEFS. Esta iniciativa objetivou o aprendizado de conceitos básicos de programação por estes estudantes, no intuito de tentar diminuir a evasão de estudantes no curso.

Pretendemos dar continuidade ao projeto nos semestres posteriores. Além disso, de já estarmos dando foi realizado um curso mais prolongado de Scratch aos alunos participantes da oficina. Acreditamos que, com isso, é possível proporcionar o amadurecimento do aluno nos conceitos básicos de programação e assim facilitar uma melhor transição e aproveitamento nas disciplinas de programação em sua vida acadêmica.

Como esta atividade foi executada pela primeira vez na UEFS no primeiro semestre do presente ano, ainda não há como avaliar quantitativamente se houve ou não redução da evasão e retenção. Estes dados somente poderão ser levantados à medida que os alunos avancem no curso. Entretanto, por conta da resposta dos participantes da oficina, acreditamos que contribuição fornecida os auxiliie no curso de sua graduação, e que tais reduções possam ser verificadas nos próximos semestres. De modo geral, pode se dizer que os objetivos que motivaram a realização desta atividade foram alcançados, pois mesmo não podendo garantir que a oficina teve impacto refletindo na diminuição da reprovação, a realização desta serviu como uma importante ferramenta de análise. Assim, podemos perceber qual a forma mais eficiente de abordar os conteúdos durante atividades propostas com a ferramenta Scratch dando liberdade e ao mesmo tempo garantindo que o aluno tenha curva de aprendizagem interessante.

O grupo executor desta oficina já realizou um levantamento, verificando problemas no aprendizado de programação em outros cursos da UEFS além de engenharia de computação, tais como engenharia civil e engenharia de alimentos. Ambos os cursos tem sofrido bastante retenção em suas disciplinas de iniciação à programação. Está em fase de planejamento uma oficina para estes cursos e espera-se, após a sua realização, que os alunos das engenharias tenham maior facilidade para aprender programação e



que esses conceitos possam ser efetivamente utilizados no exercício profissional futuro destes engenheiros.

6. REFERÊNCIAS / CITAÇÕES

AURELIANO, V. C. O.; TEDESCO, P. C. A. R. Utilizando o Scratch para Apoiar o Processo de Ensino-aprendizagem de Programação para Iniciantes na EaD. Anais: XXXIII – Congresso da Sociedade Brasileira de Computação. Maceió: Centro de Convenções de Maceió, 2013.

BARROS, E. A. R. *et al.* Alice: Uso do Software no Processo Educacional Junto aos Cursos de Engenharia. Anais: XL – Congresso Brasileiro de Educação em Engenharia. Belém: UFPA, 2012.

KAHWAGE, C. *et al.* Jogo Baralho das Variáveis. Anais: XXXIII – Congresso da Sociedade Brasileira de Computação. Maceió: Centro de Convenções de Maceió, 2013.

KOLLING, M.; KOCH, B.; e REMBERG, J. (1995). Requirements for a first year object-oriented teaching language, volume 27. ACM.

MÉLO, F. E. *et al.* Do Scratch ao Arduino: Uma Proposta de Ensino Introdutório de Programação para Cursos Superiores de Tecnologia. Anais: XXXIX - Congresso Brasileiro de Educação em Engenharia. Blumenau: Hotel Himenblau, 2011.

MIT – Massachusetts Institute of Technology. **ABOUT Scratch (Scratch Documentation Site)**. Disponível em: <http://info.scratch.mit.edu/About_Scratch>. Acesso em: 04 junho 2013.

PEREIRA, P. S.; MEDEIROS, M.; MENEZES, J. W. M. Análise do Scratch como Ferramenta de Auxílio ao Ensino de Programação de Computadores. Anais: XL – Congresso Brasileiro de Educação em Engenharia. Belém: UFPA, 2012.

RAPKIEWICZ, C.; FALKEMBACH, G.; SEIXAS, L.; DOS SANTOS, N.; ROSA, V.; KLEMMANN, M. (2006). Estratégias pedagógicas no ensino de algoritmos e programação associadas ao uso de jogos educacionais. *Novas Tecnologias na Educação*, 4(2), 2006.

RESNICK, M.; MALONEY, J.; MONROY-HERNÁNDEZ, A.; RUSK, N.; EASTMOND, E.; BRENNAN, K.; MILLNER, A.; ROSENBAUM, E.; SILVER, J., SILVERMAN, B., *et al.* (2009). Scratch: programming for all. *Communications of the ACM*, 52(11):60–67.

SOFTEX (2012). *Software e Serviços de TI: A Indústria Brasileira em Perspectiva*, vol. 2. SOFTEX.



USING SCRATCH AS A TOOL TO SUPPORT LEARNING COMPUTER PROGRAMMING

Abstract: *This paper presents a work developed at the State University of Feira de Santana that aims to facilitate learning of computer programming by the freshmen of the Computer Engineering Program at this university, as well as to decrease the dropout and retention rates in the courses related to the area of programming. To do so, the Scratch tool was used as a playful learning environment, where basic programming concepts were learned through a workshop offered in the beginning of the academic term.*

Key-words: *playful learning environments, learning computer programming, Scratch, tools.*