

Resgatando a Linguagem de Programação Logo: Uma Experiência com Calouros no Ensino Superior

Alberto A. C. Raiol¹, João Sarges¹, Aline Souza, Sivaldo Silva, Fábio Bezerra¹

¹ Instituto Ciberespacial – Universidade Federal Rural da Amazônia (UFRA)
Av. Presidente Tancredo Neves, 2501 – Belém – PA

{alan.raiol, joao.sarges, fabio.bezerra}@ufra.edu.br

Abstract. *Computer programming is a complex activity, specially for students who start their studies in the area of informatics. This reality has just raising the repetition and dropout rates in higher education. This paper presents a teaching proposal developed with freshmen of information systems course, using the Logo programming language as a teaching tool. In both experiments reported in the study we observed that the group of students who participated in the workshops have performed better in introductory programming course, and that this improvement was most notable in the first experiment, where participants demonstrated more motivated.*

Resumo. *A programação de computadores é uma atividade complexa e por isso incomoda muitos alunos que iniciam seus estudos em informática. Essa realidade acaba elevando os índices de repetência e evasão nos cursos superiores da área de computação. O presente trabalho apresenta uma proposta bem sucedida de ensino de programação de computadores desenvolvida com os calouros do curso de sistemas de informação, utilizando a linguagem de programação Logo como ferramenta de ensino. Nas duas experiências relatadas no trabalho, observamos que a participação dos alunos nas oficinas contribuiu para melhorar o desempenho em disciplina introdutória de programação, sendo que, essa melhora foi mais notável na primeira intervenção, onde os participantes se demonstraram mais motivados.*

1. Introdução

Por volta da década de 50 os computadores começaram a ser comercializados, e a partir disto surgiram computadores que possibilitaram melhorar a programação e a capacidade de armazenamento de suas informações. Foi neste período que surgiram também as experiências iniciais do uso do computador na educação, mais precisamente em cursos de pós-graduação e em centros de pesquisa da IBM nos EUA [Valente et al. 1999]. Nos dias atuais o ensino e estudo da computação encontra-se modificado, por exemplo, com a crescente oferta de vídeo-aulas e outros recursos pedagógicos. No entanto, o cenário profissional encontra-se carente de profissionais qualificados, bem como tem atraído menos estudantes para os cursos de computação e áreas afins.

Acreditamos que nos cursos superiores de graduação em computação, as disciplinas que utilizam ou exigem programação de computadores são as mais importantes ou fundamentais, pois são essenciais para construir ou entender os diferentes recursos de software disponíveis. Entretanto, essas disciplinas são vistas com imensa preocupação

por vários estudiosos e educadores, como também pelas instituições de ensino, já que elas possuem o maior índice de reprovação entre os alunos [Valaski e Paraiso 2012]. Em [Sirotheau et al. 2011] os autores destacam o problema do elevado índice de repetência e evasão presentes nos cursos de computação.

Aprender a programar é um processo realmente complicado e dificultoso para um grande percentual de alunos que iniciam seus estudos em computação [Mota et al. 2008]. Uma vez que o primeiro contato com a programação ocorre geralmente nas disciplinas introdutórias de curso superiores de computação, a assimilação de seus conceitos iniciais torna-se uma tarefa muito difícil para maioria dos estudantes, o que é um fator muito desestimulador e que se agrava se a prática desses conceitos estiver aliada a uma linguagem de programação cuja sintaxe seja bem específica e complexa [Valaski e Paraiso 2012]. Conforme os trabalhos de [Pimentel et al. 2003] e [de Jesus Gomes 2000], muitos acadêmicos iniciantes apresentam enormes dificuldades em absorver e aplicar certos conceitos abstratos de programação. Em [Mota et al. 2008] os autores associam essa dificuldade à complexidade da linguagem de programação empregada e ao processo complexo de aprender e desenvolver lógica de programação. Em [de Jesus Gomes 2000] os autores condicionam essa dificuldade à falta de compreensão dos alunos sobre os conceitos básicos de programação, assim como a dificuldade em aplicar tais fundamentos. Nesse sentido, em [Neto e Schuvartz 2007] os autores esclarecem que existe um conjunto de habilidades vinculadas a programação que os calouros precisam agregar para poderem programar. Essas habilidades compreende-se como capacidade de resolver problemas, raciocinar logicamente, abstrair informações, habilidade matemática, dentre outras.

Esses muitos problemas relacionados ao processo de aprendizagem de programação, além de provocar o baixo rendimento e a reprovação de muitos alunos, logo nas disciplinas introdutórias de programação, ocasionam o desinteresse do aluno pelo curso, levando aumento da evasão e a repulsa pela programação, sendo que, o acadêmico formado nessas condições, não tendo as habilidades almejadas de programador, dificilmente procurará trabalho em tal atividade [Neto e Schuvartz 2007].

Os cenários acima mencionados, abrangem de forma geral a situação do ensino de programação, vivido pelos iniciantes no ensino superior nos cursos de computação da Universidade Federal Rural da Amazônia (UFRA), principalmente aos que cursam a disciplina Técnicas de Programação I. Esses problemas motivaram as experiências relatadas neste trabalho, de modo que pudéssemos intervir e investigar estes mesmos cenários no curso de Bacharelado Sistemas de Informação (BSI) na UFRA, do campus Belém. Então, através de oficinas utilizando a Linguagem de Programação Logo, aplicadas aos calouros do curso, além de buscarmos possibilitar maior envolvimento dos estudantes com o conteúdo de algoritmos, observamos que o contato extra-classe com a programação lúdica de computadores trouxeram alterações no desempenho acadêmico desses alunos. Para isso, ao final da experiência, verificamos e comparamos as médias das turmas em duas disciplinas: Técnicas de Programação I e Lógica Matemática.

O restante deste Trabalho está organizado, como segue: na Seção 2 apresentaremos diversos trabalhos voltados ao ensino de computação; na Seção 3 abordaremos as ferramentas educacionais utilizadas no desenvolvimento deste trabalho, apresentaremos o plano de ensino, faremos uma breve descrição sobre o cenário em que foi desenvolvida

a oficina e também o processo de avaliação que realizamos para coletar os resultados; na Seção 4 apresentaremos as experiências realizadas através das oficinas com as turmas de BSI 2013 e BSI 2014, evidenciando as características e os resultados encontrados em cada aplicação; finalmente, na Seção 5 apresentaremos considerações finais sobre a referida experiência.

2. Trabalhos Correlatos

A dificuldade de aprendizagem de programação de computadores é um problema antigo, relatado por vários autores, como por exemplo [Pimentel et al. 2003, Valaski e Paraiso 2012, Chaves et al. 2013, Sirotheau et al. 2011, Dantas et al. 2013, da Silva et al. 2014]. O baixo rendimento apresenta-se como um problema explorado nos trabalhos [Pimentel et al. 2003, Valaski e Paraiso 2012, Chaves et al. 2013], ou seja, essa dificuldade de aprendizagem impacta no desempenho dos alunos ao longo do curso de graduação em computação. Além do baixo rendimento, outro efeito da dificuldade de aprendizagem de programação de computadores é a evasão ou a repetência, que são consequências claramente mais sérias. Nesse contexto, os trabalhos em [Sirotheau et al. 2011, Chaves et al. 2013, Dantas et al. 2013, da Silva et al. 2014] discutem os problemas da evasão e repetência.

Em [Valaski e Paraiso 2012] encontra-se uma experiência realizada com calouros de sistema de informação, utilizando o software educacional Alice. Apesar dos resultados obtidos concluírem que a aplicação não ajudou de forma significativa no aumento do desempenho acadêmico dos alunos, verificou-se que o uso da plataforma Alice é recomendada para estudantes que sentem dificuldades em programação.

Em [Sirotheau et al. 2011, Chaves et al. 2013, Dantas et al. 2013, da Silva et al. 2014] os autores apresentam soluções distintas que podem contribuir para minimizar os índices de repetência e evasão, como também, o baixo rendimento dos alunos em disciplinas que envolvem programação. Para esse contexto, [Sirotheau et al. 2011] propõe como solução, o uso de uma ferramenta chamada JavaTool, que gera a visualização de códigos e contribui para um feedback automático. [Chaves et al. 2013] desenvolveu uma ferramenta chamada "MOJO", que integra o ambiente virtual de aprendizagem MOODLE com os Juizes Online SPOJ Brasil e URI Online Judge. Em [Dantas et al. 2013] foi desenvolvido um jogo eletrônico para o ensino de programação com a linguagem Python. Em [da Silva et al. 2014] é realizada uma análise empírica, verificando pontos positivos e negativos de cinco ferramentas (RoboMind, Scratch, SuperLogo, Kodu Lab e Alice), dedicadas ao ensino de computação no ensino básico.

Diante disso, observamos através desses trabalhos que a nossa proposta também pode ser uma solução alternativa eficiente diante dos problemas mencionados. [de Jesus Gomes 2000] acredita que a linguagem Logo pode ser um bom instrumento para ensinar conceitos básicos de programação, principalmente para aqueles que estão iniciando seus estudos.

Alguns autores relatam a falta ou a dificuldade de encontrar ferramentas educacionais que efetivamente sirvam de auxílio ao educador e aos alunos, favorecendo o ensino e a aprendizagem de programação. Em [Chaves et al. 2013] e [Dantas et al. 2013] encontramos relatos desse problema, que também é um dos motivos nesses estudos para a

criação de novas ferramentas voltadas para o ensino de programação.

Nesse sentido, verificamos que muitos autores estão cada vez mais interessados em criar novas ferramentas educacionais em vez de procurá-las. Entretanto, [Frigo et al. 2013] apresenta um projeto piloto com 30 alunas explorando de forma fácil e prática diversas tecnologias computacionais, tais como: podcasting, o HTML, kits de robótica educacional e a ferramenta Kodu, para criação de jogos digitais. Em [de Oliveira et al. 2014] é relatada uma experiência com alunos do ensino fundamental, onde foi utilizada a computação desplugada e o Software Scratch para o ensino de conceitos básicos de computação. Essas experiências mostram que as ferramentas utilizadas trouxeram resultados eficazes ou favoráveis para o ensino, contribuindo na aprendizagem e no interesse dos alunos pela programação.

Assim, porque há diversas ferramentas disponíveis e gratuitas no mercado, por que não utilizá-las? Por essa razão, adotamos a linguagem Logo em nossas experiências, gratuitamente disponível por meio do software KTurtle no Linux.

3. Materiais e Métodos

Nesta seção apresentaremos as ferramentas, a metodologia e o método de avaliação que utilizamos nesse trabalho. Na Seção 3.1 mostraremos as ferramentas educacionais utilizadas. Na Seção 3.2 faremos uma descrição dos conteúdos abordados e apresentaremos os cenários onde as oficinas foram aplicadas, destacando as peculiaridades de cada intervenção. E, por fim, na Seção 3.3, descrevemos como avaliamos as nossas oficinas, em cada uma das turmas investigada.

3.1. Linguagem Logo e Ambiente KTurtle

Sob influência do avanço que a computação vinha apresentando na década de 60 e a influência das teorias construtivistas de Jean Piaget, Seymour Papert e sua equipe desenvolveram no MIT (Massachusetts Institute of Technology) a Linguagem de Programação Logo. Essa linguagem trouxe algo que difere do que se tinha até então sobre a informática, pois tinha como objetivo construir a programação de forma mais acessível [Prado 1996].

O ambiente KTurtle serviu de base para as experiências desenvolvidas neste trabalho. Este ambiente é organizado em várias áreas, uma das quais é reservada para a codificação do programa de computador. O KTurtle utiliza o turtlescript, que é uma linguagem de programação de computadores baseada na linguagem Logo [Breijs et al. 2014]. O KTurtle encontra-se disponível nas variadas distribuições Linux, e no momento da redação deste trabalho, está na versão 3.5.5, sendo que o KTurtle 4.0 já está em desenvolvimento. Neste ambiente de programação o usuário interage através duas áreas: (i) uma área de edição de códigos e (ii) uma área de desenho. Dessa forma, quando a área de escrita do código é executada, o usuário poderá ver o resultado imediato na área de desenho. Este feedback na área de desenho tem como principal elemento da sua interface uma tartaruga que “responde” aos comandos escritos (codificados) na área de desenho. A implementação desta área se deu em substituição do robô que era usado no passado para interpretar os comandos.

Dentre as características do KTurtle, que fundamentam o uso desta ferramenta na realização deste trabalho: (i) simplicidade, pois os comandos que são utilizados encontram-se resumidos, para facilitar o contato das pessoas com pouca experiência em

programação de computadores; (ii) o idioma da ferramenta possui o pacote em português do Brasil, facilitando seu uso; (iii) está disponível de forma livre, gratuito e para diversas distribuições Linux, além de já estar incluso no pacote do KDEdu, sendo distribuído pelo Linux Educacional, tornando-se uma ferramenta importante para ser utilizada no contexto da sala de aula de escolas públicas.

3.2. Descrição da Oficina

Para a realização da oficina, os dois instrutores juntamente com o orientador, realizaram o convite em sala de aula para os discentes calouros do curso de BSI, com aproximadamente 50 estudantes. O convite vinha acompanhado do relato da importância que a habilidade com a programação de computadores tem para o desempenho do aluno no curso de sistemas de informação, bem como uma descrição sucinta da oficina, como: conteúdo, horário e dias das aulas e frequência mínima.

A oficina foi organizada com carga horária total de 24h/aula, dividida em conteúdos básico e avançado. No módulo básico, organizado para ser executado em 8h/aula em dois dias de aula, foram planejados os seguintes assuntos: conceitos iniciais da computação; algoritmo; raciocínio lógico; descrição do ambiente Kturtle; comandos de movimentação e rotação da tartaruga; e comandos de configurações de cores, traços e fonte. No módulo avançado, organizado para ser executado em 12h/aula e três dias de aula, foram abordados os seguintes assuntos: variáveis; comandos de entrada e saída de dados; operadores (matemáticos, lógicos e de comparação); as estruturas de seleção e repetição; e encerrando com conceitos e criação de funções. O último dia de oficina, com 4h/aula de duração, foi planejado para concentrar as atividades de revisão do conteúdo, aplicação de prova e avaliação da oficina e dos instrutores.

Este trabalho é resultado da execução de duas oficinas, uma para os calouros de BSI ingressantes em 2013 e outra para os calouros de BSI ingressantes em 2014. Para a primeira oficina, realizada em 2013, mais de 20 alunos realizaram a inscrição. Como haviam disponíveis apenas 15 vagas, foi preciso sortear quais alunos seriam contemplados com a vaga na oficina. Dos 15 participantes, apenas 11 obtiveram frequência mínima de 75%, que serão considerados como pertencentes ao grupo de intervenção da turma de 2013. A oficina ocorreu durante os meses de junho e julho de 2013, iniciando no dia 8 de junho e encerrando dia 12 de julho, acontecendo durante seis finais de semana, com quatro horas de aula em cada encontro.

Para a turma de 2014, ampliamos para 20 vagas a oficina. No entanto, diferente do que aconteceu com a turma de 2013, apenas 14 alunos efetuaram sua inscrição. Dos inscritos, apenas nove concluíram a oficina com o percentual de frequência mínimos exigido (75%). No entanto, notou-se que muitos não se inscreveram devido a mesma ser oferecida no horário da tarde, o que é uma dificuldade para muitos alunos da UFRA que optam pelo curso noturno, justamente por possuírem alguma outra atividade no período diurno/vespertino. Diferente da turma de 2013, na turma de 2014 a oficina foi planejada para ser executada em seis dias consecutivos. Porém, somente após um mês e 15 dias, conseguimos realizar a atividade de avaliação (último dia de aula). Além disso, inovamos na primeira aula ao utilizarmos o jogo online "Quebra cabeça com Angry Birdy", desenvolvido pela organização sem fins lucrativos code.org, é um jogo que contribui no aprendizado de conceitos básicos de programação (algoritmo, sequência, estrutura de seleção e repetição), mas a partir de uma abordagem lúdica.

Nossa intenção com a atividade do jogo se concentrava em obter uma compreensão sobre os conhecimentos dos alunos em lógica de programação e também contribuir na promoção do seu exercício. Sobre esta questão os trabalhos em [Scaico et al. 2012a, Scaico et al. 2012b, Dantas et al. 2013] relatam que está sendo uma prática comum utilizar jogos educacionais no ensino de programação, pois eles podem contribuir significativamente no aprendizado de conceitos e no desenvolvimento de competências dessa área.

3.3. Avaliação das Oficinas

Apesar de termos planejado para as oficinas um dia dedicado à revisão do conteúdo e avaliação, todos os dias de aula havia a exposição de exercícios de fixação dos conceitos e comandos ensinados, bem como um ou mais desafios. Os exercícios de fixação eram aplicados logo após a explicação do conteúdo, possuindo menor grau de complexidade, enquanto que os desafios eram aplicados após os exercícios e se caracterizavam por serem mais complexos, necessitando de algoritmos mais elaborados para resolvê-los, o que exigia dos alunos maior tempo para desenvolverem suas respostas.

Em ambas as intervenções, de modo geral, os aspectos considerados para avaliar o desempenho do aluno nas atividades das oficinas em cada módulo foram: (i) realizou (para quem fez todos os exercícios ou desafios); (ii) realizou parcialmente (para quem deixou de realizar alguma das atividades); e (iii) não realizou. Entretanto, na segunda intervenção, essa avaliação foi mais detalhada, pois se especificou também o tipo de acompanhamento que o instrutor teve com o aluno durante as atividades. Assim, também utilizamos os critérios (i) com ajuda (para quem precisou do auxílio do instrutor, com uma dica por exemplo, para resolver a atividade) e (ii) sem ajuda (para quem não precisou de nenhum auxílio).

Como uma forma de avaliar se nossas intervenções foram significativas, optamos por verificar e comparar o desempenho acadêmico das turmas investigadas em duas disciplinas regulares do curso de BSI, ofertadas aos alunos no primeiro semestre do curso, que foram: Técnicas de Programação I e Lógica Matemática. Assim, nessa análise, cada turma foi separada em dois grupos: o grupo de intervenção, compostos por aqueles que participaram da oficina e atenderam a frequência mínima de 75%; e o grupo de controle, formado por aqueles que não participaram da oficina ou não atenderam a frequência mínima de 75%.

4. Resultados e Discussões

Nesta seção apresentamos o desempenho dos alunos na oficina, ou seja, se acompanharam bem o conteúdo proposto na oficina de Logo. Também apresentamos a avaliação dos efeitos que a participação dos alunos na oficina de Logo teve em seu desempenho acadêmico. Para tanto, esta seção está organizada da seguinte forma: na Seção 4.1 apresentamos os resultados com a turma de 2013; enquanto que na Seção 4.2 apresentamos os resultados da turma de 2014; finalmente, na Seção 4.3 discutimos esses resultados.

4.1. Turma BSI 2013

Os alunos que participaram da oficina em 2013 obtiveram bons resultados na oficina. Dos 11 participantes que frequentaram as oficinas com frequência mínima de 75%, sete

realizaram todos os exercícios e desafios do módulo básico, e nove realizaram todos os exercícios e desafios do módulo avançado. Em relação a atividade avaliativa realizada no final da oficina, verificamos que oito alunos acertaram todas as questões, dois acertaram 80% da prova, e apenas um acertou 40% da prova.

Quanto ao desempenho nas disciplinas de Técnicas de Programação I (TPI) e Lógica Matemática (LM), a Tabela 1 exibe as médias observadas nas duas disciplinas, mas separando as médias da seguinte forma: a turma inteira, a turma de intervenção (alunos que participaram da oficina) e a turma de controle (alunos que não participaram na oficina).

Tabela 1. Médias da Turma BSI 2013 - Turma, Intervenção e Controle

	TP I	LM
Média da Turma	7,02	7,50
Média do Grupo de Intervenção	7.83	7.96
Média do Grupo de Controle	6.70	7.34
Diferença Significativa?	Não (p-value = 0.114)	Não (p-value = 0.095)

A diferença de desempenho da turma de intervenção é evidente. Como pode ser observado, a média do grupo de intervenção é superior à média da turma, e por conseguinte, superior à média do grupo de controle. Vale observar que para a disciplina de Técnicas de Programação I essa diferença é mais evidente.

4.2. Turma BSI 2014

Os alunos que participaram da oficina em 2014 também obtiveram bons resultados na oficina. Dos nove participantes que frequentaram as oficinas com frequência mínima de 75%, sete realizaram todos os exercícios e desafios do módulo básico, e seis realizaram todos os exercícios e desafios do módulo avançado.

Adotando os critérios “Sem Ajuda” e “Com Ajuda”, que refere-se ao apoio prestado aos alunos na realização das atividades, observamos que no módulo básico 77% dos exercícios e desafio apresentados pelos alunos foram resolvidos sem a ajuda do instrutor. No módulo avançado esse número caiu para 71%, o que ainda é bastante expressivo. Esses números nos fazem acreditar que o aproveitamento dos alunos no curso foi satisfatório. Por outro lado, na avaliação final, realizada no último dia da oficina, os alunos obtiveram rendimento abaixo do esperado.

Assim sendo, consideramos que um dos principais fatores que contribuíram para esse resultado adverso, foi o longo período decorrido entre as aulas de Logo e a aplicação da avaliação, que foi em torno de um mês e 15 dias e ocorreu devido à uma serie de paralisações e indicativos de greves dos servidores públicos ocorridos na UFRA, o que dificultou a marcação da avaliação com os alunos, que durante esse tempo, nos acreditamos que não continuaram seus estudos com a linguagem estudada.

Quanto ao desempenho nas disciplinas de Técnicas de Programação I (TPI) e Lógica Matemática (LM), a Tabela 2 exibe as médias observadas nas duas disciplinas, mas separando as médias da seguinte forma: a turma inteira, a turma de intervenção (alunos que participaram da oficina) e a turma de controle (alunos que não participaram na oficina).

Tabela 2. Médias da Turma BSI 2014 - Turma, Intervenção e Controle

	TP I	LM
Média da Turma	7,38	8,14
Média do Grupo de Intervenção	7.38	7.76
Média do Grupo de Controle	7.38	8.29
Diferença Significativa?	Não (p-value = 0.998)	Não (p-value=0.542)

Como relatado na tabela, a média dos alunos que realizaram a oficina ficou abaixo da média do grupo de controle na disciplina de Lógica Matemática. No entanto, na disciplina de Técnicas de Programação I, a média grupo de intervenção foi similar à do grupo de controle e à média da turma.

4.3. Discussões

Ambas as turmas tiveram bom aproveitamento na oficina, mas notamos uma diferença quando comparamos o rendimento das duas turmas na resolução dos exercícios e desafios apresentados durante as aulas. Como não temos dados numéricos (ex: nota na prova), não podemos expressar quantitativamente essa diferença. No entanto, como apresentado nas seções anteriores, qualitativamente essa diferença foi observada pelos instrutores. Por exemplo, no ano de 2013 foi necessário realizarmos sorteio dos alunos inscritos, enquanto que em 2014 sobraram vagas.

Portanto, acreditamos que há, no mínimo, uma diferença motivacional entre os alunos que frequentaram as oficinas nos dois anos. Porque acreditamos que há essa diferença motivacional, poderíamos nos perguntar se, para a turma de 2013, o desempenho desse grupo de intervenção seria maior que o desempenho do grupo de controle mesmo sem executarmos a oficina de Logo. Uma vez que a participação na oficina era voluntária, não podemos apresentar uma resposta mais objetiva ou contundente para essa hipótese. Por outro lado, porque conhecemos os alunos da turma de 2013, acreditamos que o grupo de intervenção relatado aqui é uma amostra representativa da turma, fato que descartaria a referida hipótese. Outra diferença que observamos nas duas turmas é o desempenho nas disciplinas investigadas. Para nossa surpresa, o grupo de intervenção da turma de 2014 obteve menor desempenho na disciplina de Lógica Matemática. Porém, o desempenho na disciplina de Técnicas de Programação I foi igual ao do grupo de controle.

Entrevistando a professora que ministra a disciplina, a mesma nos relatou que os participantes da oficina de Logo tiveram um acentuado progresso em sala de aula. Por isso acreditamos que esse contato adicional com a programação de computadores representa, no mínimo, uma oportunidade para os participantes da oficina de reforçar os conceitos apresentados em sala de aula. No caso, acreditamos que esse reforço tem um efeito ainda maior porque a linguagem trabalhada na oficina é simples e lúdica, facilitando o processo de aprendizagem na disciplina de Técnicas de Programação I, que adota a linguagem C. Por essa razão confiamos que a oficina foi essencial na evolução do desempenho dos alunos, colaborando na aprendizagem e aumentando o interesse pela prática de programar.

5. Considerações Finais

A programação de computadores é uma atividade complexa, mas essencial para o bom desempenho dos alunos nos cursos da área de computação. Portanto, a ausência de ações

que fortaleçam o aprendizado dessa atividade pode ter como efeito a elevada taxa de repetência ou mesmo evasão. Este trabalho relatou duas experiências com calouros do curso de BSI da UFRA, realizadas em 2013 e 2014. Nessas experiências observamos que atividades extra-curriculares com a programação de computadores ajudam os alunos ingressantes de um curso superior em computação a desenvolverem ou ampliarem suas habilidades com a programação de computadores e o interesse pela prática de programar.

Além disso, vemos os resultados como um resgate da linguagem Logo, pois apesar de ser uma linguagem comumente utilizada como ferramenta de ensino de crianças e jovens, as experiências relatadas aqui são evidências suficientes para estimular experiências semelhantes em cursos de computação de outras universidades, auxiliando os professores no ensino de conceitos básicos de programação, uma vez que ela facilita essa aprendizagem, aumentando o interesse dos alunos pela área. Nos cursos de licenciatura em computação, a institucionalização do ensino de programação de computadores com a linguagem Logo pode ser, também, uma oportunidade de desenvolver nos alunos uma habilidade com uma ferramenta de trabalho, uma vez que a linguagem Logo é comumente adotada no ambiente escolar, que esses futuros profissionais experimentarão.

Como trabalho futuro, planejamos repetir a oficina com as próximas turmas ingressantes – por exemplo, já temos programada uma oficina para os calouros de 2015. Além disso, pensamos em avaliar o desempenho dos alunos participantes da oficina em outras disciplinas que envolvem programação, como por exemplo, Programação Orientada a Objetos, Linguagens de Programação e Estrutura de Dados. Assim, poderemos avaliar a hipótese de que o bom desempenho nas disciplinas iniciais de programação de computadores poderá ajudar o aluno a ter bom aproveitamento no curso de computação como um todo.

Agradecimentos

Os autores agradecem o financiamento do programa PIBID/CAPES/MEC.

Referências

- Breijs, C., Mahfouf, A.-M., e Piacentini, M. (2014). *Manual do Kturtle*.
- Chaves, J. O. M., Castro, A. F., Lima, R. W., Lima, M. V. A., e Ferreira, K. H. (2013). Integrando moodle e juizes online no apoio a atividades de programação. In *Anais do Simpósio Brasileiro de Informática na Educação*.
- da Silva, E. G., Barbosa, A. F., Neto, S. R. S., Lopes, R. H. O., e Rodrigues, A. N. (2014). Análise de ferramentas para o ensino de computação na educação básica. In *XXXIV Congresso da Sociedade Brasileira de Computação*.
- Dantas, V. F., de Macedo, E. R., Andrade, J. R. B., Coutinho, D. R. A., Cavalcante, A. F., Vasconcelos, T. G., e Pereira, M. E. d. S. (2013). Combinando desafios e aventura em um jogo para apoiar a aprendizagem de programação em vários níveis cognitivos. In *Anais do Simpósio Brasileiro de Informática na Educação*.
- de Jesus Gomes, A. (2000). *Ambiente de suporte à aprendizagem de conceitos básicos de programação*. PhD thesis, Universidade de Coimbra.

- de Oliveira, M. L. S., de Souza, A. A., Barbosa, A. F., e Barreiros, E. F. S. (2014). Ensino de lógica de programação no ensino fundamental utilizando o scratch: um relato de experiência. In *XXII Workshop sobre Educação em Computação*, Brasília, DF. SBC.
- Frigo, L. B., Cardoso, P., Cardoso, J. P., Fontana, C., Irizaga, A., Victory, N., Pozzebon, E., e Yevseyeva, O. (2013). Tecnologias computacionais como práticas motivacionais no ensino médio. In *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*. SBC.
- Mota, M. P., Pereira, L. W. K., e Favero, E. L. (2008). Javatool: Uma ferramenta para o ensino de programação. In *Congresso da Sociedade Brasileira de Computação. Belém. XXVIII Congresso da Sociedade Brasileira de Computação*, pages 127–136.
- Neto, W. C. B. e Schuvartz, A. A. (2007). Ferramenta computacional de apoio ao processo de ensino-aprendizagem dos fundamentos de programação de computadores. In *XVIII Simpósio Brasileiro de Informática na Educação*, pages 520 – 528, São Paulo, SP. SBC.
- Pimentel, E. P., França, V. F. d., Noronha, R. V., e Omar, N. (2003). Avaliação contínua da aprendizagem, das competências e habilidades em programação de computadores. In *Anais do Workshop de Informática na Escola*, pages 533–544.
- Prado, M. E. B. B. (1996). Uso do computador na formação de professor: Um enfoque reflexivo da prática pedagógica. Master's thesis, Universidade Estadual de Campinas, Campinas, SP.
- Scaico, P., Lopes, D., Melo, L., e Scaico, A. (2012a). Um jogo para o ensino de programação em python baseado na taxonomia de bloom. In *XXX Cong. da Sociedade Brasileira de Computação/XX WEI-Workshop de Educação em Informática*, Curitiba, PR. SBC.
- Scaico, P., Lopes, D., Silva, M. d. A., Silva, J. d., Neto, S., e Falcão, E. d. S. (2012b). Implementação de um jogo sério para o ensino de programação para alunos do ensino médio baseado em m-learning. In *Anais do XX Workshop sobre Educação em Computação*, Curitiba, PR. SBC.
- Sirotheau, S., de Brito, S. R., da Silva, A. d. S., Eliasquevici, M. K., Favero, E. L., e Tavares, O. d. L. (2011). Aprendizagem de iniciantes em algoritmos e programação: foco nas competências de autoavaliação. In *Anais do Simpósio Brasileiro de Informática na Educação*. SBC.
- Valaski, J. e Paraiso, E. C. (2012). Limitações da utilização do alice no ensino de programação para alunos de graduação. In *Anais do Simpósio Brasileiro de Informática na Educação*.
- Valente, J. A. et al. (1999). Informática na educação no brasil: análise e contextualização histórica. _____ et al. *O computador na sociedade do conhecimento*. Brasília: MEC, -:11–28.