

# Oficinas de Aprendizagem de Programação em uma Escola Pública através do Ambiente Scratch

Washington P. Batista, Henderson S. Chalegre,  
João Paulo S. Sena, Allen H. M. Santos,  
David M. B. dos Santos, Carlos A. Rodrigues, Roberto A. Bittencourt

<sup>1</sup> UEFS – Universidade Estadual de Feira de Santana  
Av. Transnordestina, s/n, Novo Horizonte  
Feira de Santana – BA, Brasil – 44036-900

{davidmbs, carod, roberto}@uefs.br

**Abstract.** *Raising the interest of high school students on the topic of computational thinking is an important issue in the field of computing education. In this paper, we describe a learning experience of computer programming in a high school by means of a game-based, challenge-based approach using Scratch. We offered a workshop to students of a public high school during a five-week period. Students showed high motivation and satisfaction to learn through playing.*

**Resumo.** *Despertar o interesse de estudantes da educação básica para as possibilidades de expressão que o computador oferece é uma questão relevante na área de educação em computação. Este artigo descreve uma experiência de aprendizagem de programação no ensino médio através de uma abordagem baseada em desafios e jogos com o apoio do ambiente Scratch. Durante um período de cinco semanas, foram oferecidas oficinas para duas turmas do ensino médio de uma escola pública estadual. Os resultados apontam para a motivação elevada e para a satisfação dos participantes em aprender brincando.*

## 1. Introdução

A demanda por profissionais de computação tem crescido continuamente nos últimos anos, tanto no Brasil como no resto do mundo. Entretanto, no Brasil, o número de egressos dos cursos de graduação desta área não vem acompanhando a oferta de vagas no mercado de trabalho [Observatório Softex 2012, Fusco 2011]. Embora estes cursos contem com um grande número de vagas, bem como de alunos matriculados, a elevada evasão faz com que apenas uma pequena fração dos alunos os concluam [Fusco 2011]. Há várias razões para esta evasão, destacando-se a alta reprovação em disciplinas de matemática e as dificuldades de abstração e raciocínio lógico em disciplinas técnicas. Estas habilidades são especialmente relevantes em disciplinas básicas de programação de computadores. Os conceitos introdutórios de programação abordados no primeiro semestre de cursos de computação costumam ser um desafio para os graduandos, fazendo com que muitos abandonem os cursos já nesta etapa inicial [Azzam and Career 2006]. Uma razão comumente relatada por professores e estudantes para esta evasão é que o ensino de programação nestes cursos aborda uma quantidade significativa de conceitos razoavelmente complexos em um curto período de tempo [Azzam and Career 2006].

O pensamento computacional envolve diversas habilidades como, por exemplo, abstração e decomposição, busca e ordenação, representação de dados e pensamento recursivo. É cada vez mais presente em nosso cotidiano, tanto na área de computação propriamente dita como em outras áreas do conhecimento como Biologia, Química, Medicina, Economia e Ciências Sociais, facilitando sobretudo o desenvolvimento de suas pesquisas. Por isso, incentiva-se a ideia de trabalhar o pensamento computacional desde a mais tenra idade [Wing 2006]. Trabalhos prévios sugerem que o contato com programação anterior a disciplinas introdutórias na graduação tem efeitos positivos nestas [Patil and Goje 2009].

A aprendizagem de programação no ensino médio pode não apenas atrair profissionais com aptidão e talento para a Computação, como também ajudar estudantes de outras áreas em sua educação ou em suas futuras profissões. Infelizmente, a maioria das escolas brasileiras não está preparada para este desafio. A maioria das escolas brasileiras de ensino fundamental e médio não oferece nenhuma disciplina ou atividades que procurem ensinar noções básicas de programação de computadores.

Por outro lado, o ensino de programação na educação básica pode se beneficiar largamente dos ambientes e linguagens desenvolvidos especificamente para reduzir as barreiras iniciais na aprendizagem de programação [Kelleher and Pausch 2005]. O uso de uma ferramenta que facilite o processo de aprendizagem, especialmente no primeiro contato, pode ter impactos positivos na atitude de escolares em relação à área de computação [Bittencourt et al. 2013]. Em nossa experiência, ambientes lúdicos de programação como Scratch [Maloney et al. 2010] são bastante úteis neste processo.

Este trabalho relata nossa experiência de oficinas de programação com Scratch em uma escola pública estadual do ensino médio e fundamental do município de Feira de Santana, Bahia. O objetivo destas oficinas foi capacitar estudantes no uso de princípios básicos de algoritmos e programação para a expressão flexível no computador através do desenvolvimento de programas. Este projeto foi iniciado no segundo semestre de 2012, com atividades desenvolvidas em algumas escolas deste município. O presente artigo relata as oficinas realizadas com duas turmas de alunos de ensino fundamental e médio durante o mês de dezembro de 2014 na escola pública estadual Instituto de Educação Gastão Guimarães (IEGG).

## **2. Trabalhos Relacionados**

Ainda há relativamente poucas experiências descritas em profundidade sobre o ensino de conceitos de programação no ensino fundamental e/ou médio no Brasil. Em 2005, uma iniciativa foi desenvolvida em uma escola pública de Cardoso Moreira – RJ, procurando ensinar programação através de uma combinação de linguagem natural, pseudocódigo e Pascal. Entretanto, os próprios autores concluem que o ensino da programação de maneira excessivamente formal pode não estimular os alunos a continuarem aprendendo [Pereira Jr et al. 2005]. Além disso, os programas desenvolvidos, utilizando interfaces de linha de comando, costumam ser pouco atrativos para a realidade dos estudantes, acostumados com o uso de Internet e ambientes gráficos.

Alguns trabalhos que procuram atrair estudantes para a área de computação utilizam ambientes lúdicos com capacidade de incorporar mídias diversas, como é o caso do Scratch, para que os primeiros contatos com a programação sejam mais infor-

mais e interessantes. Para ilustrar, podemos citar algumas iniciativas desenvolvidas na Bahia [Machado et al. 2010], Paraíba [Scaico et al. 2012, Scaico et al. 2013] e Pernambuco [Aureliano and Tedesco 2013]. A semelhança entre elas é que, ao menos em algum momento, utilizaram Scratch para o ensino de programação. Cada uma delas, entretanto, teve suas particularidades, até mesmo devido ao contexto em que estavam inseridas. A primeira proposta utilizou C e Scratch no ensino médio e atividades de computação desplugada no ensino fundamental. A computação desplugada objetiva desenvolver habilidades típicas de computação (e.g., desenvolvimento de algoritmos) sem o uso de computadores [Bell et al. 2009]. As atividades ofertadas na Paraíba envolveram uma gincana de computação desplugada e uma olimpíada interna de Scratch, ambas para o ensino médio. Aureliano e Tedesco (2013), por outro lado, investigaram o uso de Scratch e Portugol no ensino técnico a distância.

Este trabalho apresenta uma proposta de ensino-aprendizagem de programação em escolas de ensino fundamental e médio através do uso de Scratch. Diferente dos outros trabalhos, a nossa metodologia é baseada na descoberta de conceitos através de desafios, com o apoio de tutores estudantes de graduação em computação. De modo similar a outros trabalhos, utilizamos jogos como elemento motivador. Procuramos ainda relatar em detalhes as atividades realizadas pelos alunos, priorizando uma análise comportamental e do raciocínio lógico.

### **3. Scratch: Um Ambiente Lúdico de Aprendizagem de Programação**

O Scratch é um ambiente desenvolvido pelo *Lifelong Kindergarten Research Group* do MIT, cujo objetivo é facilitar a aprendizagem de programação. Neste ambiente, a criação de programas é feita através de manipulação direta e montagem de blocos lógicos e computacionais, sem a necessidade de conhecimento prévio de qualquer linguagem de programação [Maloney et al. 2010]. Baseado na pedagogia socioconstrutivista, Scratch foi projetado para atender três princípios básicos de design: (1) ser mais manipulável: é uma ferramenta baseada nos brinquedos Lego que permite o encaixe de blocos de programação, formando scripts a serem executados; (2) ser mais significativo: investindo em diversidade e personalização, permite que seus usuários criem os mais diferentes projetos, desde jogos até histórias; (3) ser mais social: facilitando o compartilhamento de projetos ou a cooperação por pares através de uma rede social [Resnick et al. 2009].

Nossa escolha do Scratch como ambiente básico de programação deu-se por sua simplicidade e ludicidade. Trata-se de uma ferramenta que oferece um espaço lúdico que permite aos usuários desenvolverem jogos, animações, simulações e outras atividades interessantes para todas as idades [Maloney et al. 2010, Aureliano and Tedesco 2013]. Por isso, Scratch tem sido amplamente utilizado para facilitar a aprendizagem de programação por crianças [Maloney et al. 2010] e adolescentes [Al-Bow et al. 2008]. Ao participar das oficinas criando animações e jogos que incitem sua imaginação e diversão, os estudantes ganham a prática no uso desta ferramenta, fazendo com que eles se interessem em usá-la novamente em casa e que continuem aprendendo livremente.

A ferramenta organiza os conceitos de programação em seções funcionais. Na seção de Controle, estão as estruturas condicionais e de repetição. Enquanto a seção de Movimento possui os comandos para modificar a posição de objetos no plano, a de Sensores possui verificadores de localização e comandos de leitura de dispositivos de entrada

(mouse e teclado). A seção de Aparência possui comandos para diálogo e mudanças dos trajes dos objetos. Na seção de Som, estão os comandos para iniciar/parar os sons do projeto. Já a seção de Operadores contém blocos para operações lógicas e aritméticas. Por sua vez, a seção Variáveis possui comandos para armazenar e manipular valores. Finalmente, a seção de Caneta permite que os objetos façam desenhos ao se movimentarem.

#### **4. Metodologia**

O objetivo principal das oficinas de programação é capacitar estudantes de ensino médio e fundamental para a expressão flexível no computador através de programas escritos com o suporte do ambiente lúdico de programação Scratch. Os objetivos específicos são: (1) ensinar programação através de jogos de modo a criar e fortalecer o interesse de escolares por cursos técnicos e superiores em Computação; (2) reduzir a complexidade do ato de programar para que os alunos não sintam muita dificuldade nos primeiros contatos com o pensamento computacional e a programação; (3) apresentar uma das atividades desempenhadas por um profissional da área de computação; e (4) descobrir e incentivar o desenvolvimento de talentos na área de computação.

A metodologia de ensino-aprendizagem utilizada é baseada em aprender fazendo e aprender brincando. Os tutores propõem desafios aos participantes e estes procuram responder aos desafios através da descoberta no ambiente Scratch, geralmente construindo scripts por tentativa e erro. Quando as dúvidas surgem ao longo deste processo, são sanadas pelos tutores ou monitores. O processo é centrado no estudante, incentivando que cada um busque, no seu próprio tempo, suas próprias soluções para os problemas que surgem durante o desenvolvimento – os tutores só intervêm quando é realmente necessário, para tirar dúvidas ou evitar bloqueios.

##### **4.1. Participantes**

Os participantes foram convidados através de visitas de nosso grupo de pesquisa e extensão às salas de aula da escola estadual IEGG – Instituto de Educação Gastão Guimarães. Não houve seleção porque a quantidade de interessados não superou a quantidade de vagas previstas. As oficinas tiveram a participação de 23 estudantes do ensino fundamental (oitava série) e médio (primeiro e segundo anos). Os participantes foram distribuídos em duas turmas: (1) 10 na turma T1: 4 homens e 6 mulheres, com média de idade de 15,9 anos (desvio padrão de 1,6) e (2) 13 na turma T2: 6 homens e 7 mulheres, com média de idade de 15,8 anos (desvio padrão de 1,1). Nenhum participante ouvira falar do Scratch ou de outro ambiente lúdico de programação. Cinco participantes relataram já ter experiência prévia com programação.

##### **4.2. A estrutura das oficinas**

Foram programadas duas oficinas, uma para cada turma, de modo que nenhuma turma ficasse sobrecarregada e que cada participante tivesse acesso a seu próprio computador. Com a carga de horária de 20 horas, divididas em cinco aulas de quatro horas, cada turma se reuniu uma vez por semana durante um período de cinco semanas. As oficinas foram realizadas no turno oposto ao das aulas dos participantes. Cada oficina foi mediada por dois tutores e apoiada por dois monitores. Tanto os tutores quanto os monitores eram graduandos de Engenharia de Computação da Universidade Estadual de Feira de Santana (UEFS), cursando entre o segundo e o quinto períodos do curso.

Um resumo de cada aula, com os projetos desenvolvidos, os objetivos e os conteúdos abordados, é apresentado na Tabela 1. A primeira aula da oficina é voltada para uma rápida explanação sobre a ferramenta, além da produção de uma animação. Ao explorar as seções funcionais da ferramenta, os participantes se familiarizam com a grande quantidade de comandos que podem ser usados para montar seus scripts. A animação é uma maneira de incentivar cada participante a usar a imaginação para criar algo de seu próprio interesse. Nesta aula, aprende-se, principalmente, sobre estruturas de controle (condições e repetições).

**Tabela 1. Cronograma das oficinas**

| <b>Aula</b> | <b>Projeto</b>                     | <b>Objetivos</b>  | <b>Conteúdos</b>   |
|-------------|------------------------------------|---|--|
| 1           | <i>Animação</i>                    | Conhecer e entender a ferramenta, e produzir uma animação simples.                          | Estruturas condicionais e de repetição (Seções: controle, aparência e movimento).  |
| 2           | <i>Pong</i>                        | Construir um jogo simples com interação entre objetos e cenário.                            | Estruturas condicionais e de repetição, variáveis, operadores lógicos (Seções: controle, aparência, movimento, sensores, variável e números).  |
| 3           | <i>Space Invaders</i>              | Construir um jogo um pouco mais complexo, com vários objetos interagindo.                   | Estruturas condicionais e de repetição, variáveis, operadores lógicos (Seções: controle, aparência, movimento, sensores, variáveis e números). |
| 4           | <i>Space Invaders e Mario Bros</i> | Concluir o jogo anterior e iniciar a construção de um jogo em estilo <i>side scroller</i> . | Estruturas condicionais e de repetição, variáveis, operadores lógicos (Seções: controle, aparência, movimento, sensores, variáveis e números). |
| 5           | <i>Mario Bros</i>                  | Concluir o jogo anterior.   | Estruturas condicionais e de repetição, variáveis, operadores lógicos (Seções: controle, aparência, movimento, variáveis e sensores).          |

A segunda aula é baseada na produção de um jogo clássico chamado *Pong*, com uma raquete rebatendo uma bola que se movimenta por um cenário com paredes. Durante a criação do jogo, os participantes aprendem a criar uma solução algorítmica para o problema que atenda aos requisitos do jogo. Além de reforçarem o aprendizado de conceitos anteriores, aprendem mais estruturas de controle e também a utilizar variáveis e operadores lógicos.

A terceira aula e parte da quarta aula são dedicadas ao desenvolvimento do jogo clássico *Space Invaders*, no estilo *arcade*. A produção deste jogo estimula os participantes a pensar um pouco mais sobre a lógica do jogo e a explorar mais blocos de comandos presentes na ferramenta e, assim, fixar de maneira mais perene alguns conceitos. Aqui, acrescenta-se a opção de troca de traje, presente na seção de Aparência do Scratch.

A segunda parte da quarta aula e a quinta aula são dedicadas à construção de uma parte do jogo *Mario Bros*, um popular jogo em estilo plataforma com *side scroll*. Eles desenvolvem uma fase onde Mario pode interagir com alguns objetos do cenário. Este jogo, assim como o *Space Invaders*, encoraja os alunos a pensar um pouco mais sobre lógica de programação. Para a produção deste jogo, os participantes utilizam comandos de quase

todas as seções, com exceção das de Som e Caneta (desnecessárias ou não disponíveis, pois não havia saídas de áudio nos computadores utilizados).

### 4.3. Coleta e análise de dados

Cada participante respondeu a um questionário após a oficina, sobre dados demográficos e suas percepções quanto às suas habilidades de programação. Os resultados foram analisados através de estatística descritiva. Além disso, os tutores e monitores realizaram uma autoavaliação após a realização da oficina, buscando identificar aspectos positivos e aspectos a serem melhorados.

## 5. Resultados

Apesar de não possuírem conhecimentos prévios de programação, os participantes conseguiram criar animações de maneira intuitiva ao navegar pela ferramenta. As dúvidas geralmente eram sobre estruturas de controle, i.e., “Sempre”, “Sempre se” e “Repita”, e eram rapidamente sanadas pelos tutores ou monitores. No projeto da animação, todos os participantes utilizaram as seções funcionais de Controle, Aparência e Movimento. Todos também usaram, ao menos uma vez, os comandos “Anuncie para todos” e “Quando eu ouvir” (seção de Controle).

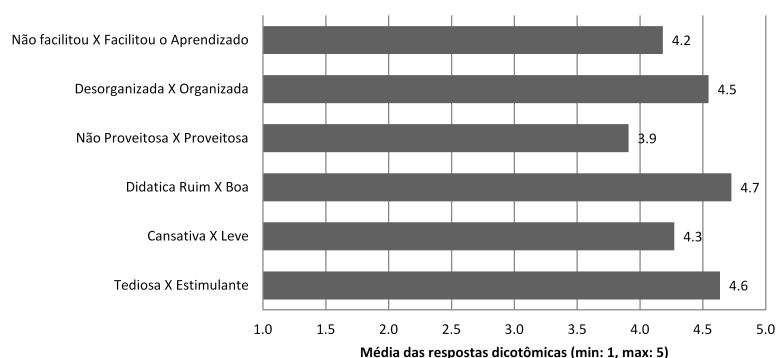
Durante a criação do jogo *Pong*, as dúvidas dos participantes se concentravam em como evitar que a bolinha não passasse pela barra e como fazer a bola voltar após tocar na barra. Os participantes importaram objetos para o projeto disponibilizados por nossa equipe, mas também puderam desenhar seus próprios objetos, caso preferissem. Todos conseguiram fazer o movimento da bola e, após algumas explicações, entenderam que poderiam utilizar a seção de sensores para detectar se a bola tocava na barra.

Uma questão recorrente foi sobre como fazer a bola apontar para um lugar aleatório ao tocar na barra. Por isso, explicou-se o comando “Sorteie número entre valor1 e valor2”, usado para sortear um ângulo diferente para apontar a direção da bola. Alguns participantes perguntaram sobre como contar pontos, para determinar o vencedor dentre vários jogadores, surgindo naturalmente a necessidade de variáveis. Vale ressaltar que os tutores e monitores explicavam como o jogo funcionava, quais seções funcionais poderiam ser utilizadas e esclareciam dúvidas quando necessário.

No *Space Invaders*, introduziu-se o conceito de variável. O Scratch torna este conceito relativamente simples pela sintaxe utilizada. A principal dúvida era como fazer a interação entre os objetos (e.g., como fazer a nave ou o *alien* atirar). Os participantes também tiveram dificuldade em entender o “Mude para o traje” (seção Aparência). Um exemplo do uso deste comando é quando o tiro atinge o *alien*, e ele “muda para o traje” para explodir e então desaparecer do jogo. Para os participantes que terminaram mais cedo, propôs-se outro desafio: criar uma fase adicional onde o jogador enfrenta um *alien* chefe. Os alunos gostaram do desafio e implementaram a nova fase.

Por fim, a criação do jogo *Mario Bros* não trouxe novas dúvidas. O conhecimento adquirido pelos estudantes era satisfatório para usar a ferramenta independentemente, quase que sem perguntar a ninguém. Como a lógica da fase usada do *Mario Bros* era relativamente simples, os participantes conseguiam realizá-la sem maiores dificuldades. Os que terminavam rápido eram desafiados a melhorar a fase através da adição de um inimigo.

Pedimos que os participantes avaliassem a oficina através de seis critérios-binômios que variavam numa escala de cinco pontos de um extremo a outro. Por exemplo, cada respondente avaliou se a oficina era *tediosa–estimulante*: quanto mais à esquerda (à direita) marcasse, mais tediosa (estimulante) era a oficina, em sua opinião. Os outros cinco critérios-binômios foram: *cansativa–leve*, *didática ruim–boa didática*, *não proveitosa–proveitosa*, *desorganizada–organizada*, e *não facilitou o aprendizado–facilitou o aprendizado*. As médias das avaliações para cada critério são apresentadas na Figura 1.



**Figura 1. Avaliação da oficina**

De modo geral, podemos dizer que a oficina foi bem avaliada. É perceptível que os estudantes acharam a oficina estimulante, leve, com boa didática, proveitosa, organizada e propícia ao aprendizado. O item com menor avaliação foi sobre a oficina ser proveitosa ou não, com média 3,9, o que sugere uma certa neutralidade e a possibilidade de alguns participantes não pretenderem utilizar os conhecimentos aprendidos após a oficina.

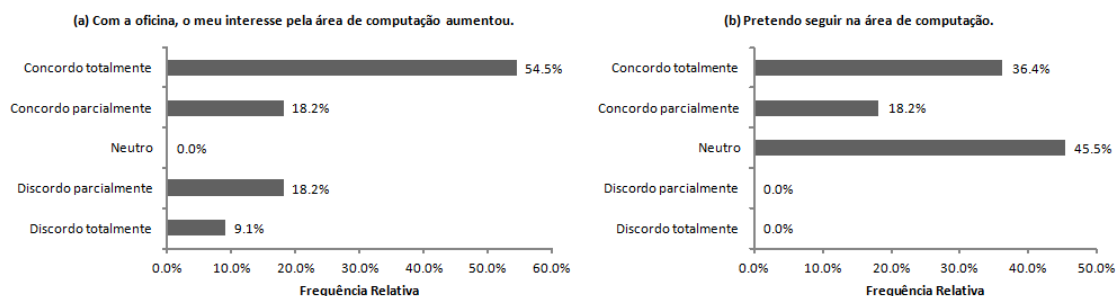
Ainda questionamos os estudantes quanto ao interesse em computação: se o seu interesse aumentou pela área e se pretendiam seguir uma carreira na área de computação. As respostas variaram em graus de concordância e os resultados são descritos na Figura 2. A Figura 2(a) ilustra que 72,7% concordaram, em algum grau, que o interesse pela área aumentou, enquanto que o restante discordou. Já a Figura 2(b) mostra que 54,6% dos participantes demonstraram intenção de seguir uma carreira em computação, enquanto que o restante se manteve neutro. Em suma, percebe-se um interesse significativo pela computação ao final da oficina.

## 6. Lições Aprendidas

As lições aprendidas neste trabalho foram derivadas a partir de discussões sobre as autoavaliações dos tutores e monitores.

*Metodologia de ensino-aprendizagem: aprendizagem ativa e lúdica.* Percebemos que a metodologia é fundamental para manter o interesse dos estudantes e facilitar a aprendizagem. Além disso, o uso de analogias entre Scratch e assuntos que os participantes já conheciam ajudou muito no ensino de algoritmos. Ao observarmos os erros dos participantes, sempre íamos até eles e discutíamos o problema encontrado e possíveis soluções. Isso os fazia repensar a situação e mudar alguns dos seus conceitos. Algumas

vezes, eles já sabiam como resolver o problema, e só precisavam de alguma dica ou incentivo. O uso constante de perguntas sobre o conceito em questão mostrou-se eficaz para verificar se os conceitos estavam sendo realmente aprendidos.



**Figura 2. Interesse pela computação após a oficina**

Os desafios também fizeram com que eles ficassem mais motivados em resolver os problemas propostos. Era comum que fossem além do solicitado, revelando uma curiosidade sobre o que podia ser feito com Scratch. Ainda nesta perspectiva, verificamos que a combinação de desafios com jogos se mostrou muito eficaz em motivar os estudantes a continuarem indo para as oficinas. Alguns deles relataram que até pensavam em como poderiam criar outros jogos ao chegarem em casa. Além disso, já que o jogo *Mario Bros* foi planejado para o último dia da oficina, muitos participantes se mostraram bastante motivados em desenvolver o projeto. A ferramenta Scratch contribuiu para facilitar o exercício da criatividade e a liberdade de realizar diferentes maneiras de desenvolver jogos.

*Interesse na Computação.* Foi gratificante despertar o interesse dos estudantes em programação através de uma ferramenta lúdica. Percebeu-se que a ideia de que programação é difícil foi superada através, sobretudo, da realização de algo interessante através de brincadeiras e diversão. O envolvimento com a ferramenta foi notável, muitos mencionaram que estavam praticando em casa, fazendo jogos personalizados. Por isso, criou-se um grupo em uma rede social para manter o diálogo com os participantes. Também foi gratificante a relação estabelecida entre os estudantes e os tutores e monitores. Durante o intervalo, os primeiros frequentemente perguntavam como era o curso de Engenharia de Computação e o que se fazia lá.

*Estrutura da oficina.* A quantidade de monitores deve ser adequada para não prejudicar o andamento das atividades. Em alguns momentos, como no primeiro encontro, a demanda de dúvidas é alta e é comum que tutores e monitores se sobrecarreguem. Porém, no decorrer da oficina e com o aumento de interesse dos estudantes, notamos que o recurso aos monitores era ainda maior que no início, não por causa de dúvidas sobre conceitos de programação, mas para tentar inovar e implementar novas ideias nos jogos.

*Infraestrutura física.* Uma dificuldade enfrentada foi a infraestrutura da escola parceira, com a falta de computadores em bom estado. Para evitar maiores transtornos, os tutores e monitores verificaram previamente as máquinas e fizeram ajustes naquelas passíveis de conserto. Ainda assim, ocorreram falhas nos computadores durante as oficinas, ocasionando, inclusive, perda de parte dos projetos feitos pelos alunos. Por isso, deve haver um planejamento adicional a fim de definir estratégias para que problemas



com computadores não interferiram muito no andamento das oficinas.

*Frequência nas oficinas.* Alguns alunos não participaram da aula introdutória de Scratch e só vieram no segundo dia de oficina, não tendo acesso aos conceitos iniciais, o que exigiu uma atenção maior por parte dos monitores e tutores. Nestas situações, é preciso refletir sobre maneiras de garantir a participação em todos os dias das oficinas. Uma das estratégias que poderia ser adotada é planejar a oficina com assuntos que estão sendo abordados em sala de aula. Contudo, para que isto seja feito, é preciso garantir maior diálogo com os professores da escola, além de planejar alguns jogos da oficina na forma de jogos educacionais.

## 7. Conclusões

Esse trabalho relatou uma experiência de oficinas introdutórias de programação para estudantes do ensino médio e fundamental, apoiadas pelo ambiente lúdico Scratch. A aprendizagem foi incentivada através de uma abordagem que mesclava jogos, desafios e o ambiente lúdico.

Avaliações dos estudantes através de questionários, além de autoavaliações dos tutores e monitores das oficinas, demonstraram que a oficina foi propícia para o aprendizado, estimulante, organizada, leve e dinâmica, além de ter favorecido o aumento do interesse na área de computação.

Pretendemos continuar este trabalho através de novas oficinas em escolas parceiras. Para tanto, pretendemos desenvolver estratégias para reduzir os problemas de infraestrutura de laboratório e melhorar a frequência dos estudantes. Pretendemos ainda realizar avaliações mais detalhadas através da triangulação de dados de questionários, observações, entrevistas e artefatos desenvolvidos pelos participantes.

## 8. Agradecimentos

Os autores agradecem ao Instituto de Educação Gastão Guimarães (IEGG), em especial, à direção da escola, por ceder o laboratório de informática para que fossem oferecidas as oficinas. Agradecem ainda aos estudantes que participaram da oficina e a suas famílias, que permitiram a utilização de dados coletados nas oficinas. Este projeto é apoiado pela Fundação de Amparo à Pesquisa do Estado da Bahia (FAPESB) e pela Universidade Estadual de Feira de Santana (UEFS), na forma de bolsas de iniciação científica e extensão.

## Referências

- Al-Bow, M., Austin, D., Edgington, J., Fajardo, R., Fishburn, J., Lara, C., Meyer, S., and History, A. (2008). Using Greenfoot and Games to Teach Rising 9th and 10th Grade Novice Programmers. In *Proceedings of the 2008 ACM SIGGRAPH symposium on Video games*, volume August, page 6.
- Aureliano, V. C. O. and Tedesco, P. C. D. A. R. (2013). Utilizando o Scratch para Apoiar o Processo de Ensino-aprendizagem para iniciantes da EaD. In *Anais do XXXIII Congresso da SBC. XXI Workshop sobre Educação em Computação*, volume 21, pages 631–640, Maceió.
- Azzam, A. M. and Career, Y. (2006). Why students drop out CS1 course? In *Proceedings of the Second International Workshop on Computing Education Research*, volume 64, pages 97–108.

- Bell, T., Alexander, J., Freeman, I., and Grimley, M. (2009). Computer Science Unplugged: School Students doing Real Computing without Computers. *New Zealand Journal of Applied Computing and Information Technology*, pages 1–9.
- Bittencourt, R. A., Rocha, A. S., Santana, B. L., Santana, C. S., Carneiro, D. A., Borges, G. A., Chalegre, H. S., Silva, J. F. J., Santos, J. M. J., Silva, L. A., and Andrade, P. H. (2013). Aprendizagem de Programação Através de Ambientes Lúdicos em um Curso de Engenharia de Computação: Uma Primeira Incursão. In *Anais do XXXIII Congresso da SBC. XXI Workshop sobre Educação em Computação*, pages 749 – 758.
- Fusco, C. (2011). Tecnologia sofre com evasão universitária.
- Kelleher, C. and Pausch, R. (2005). Lowering the Barriers to Programming: A Taxonomy of Programming Environments and Languages for Novice Programmers. *ACM Computing Surveys*, 37(2):83–137.
- Machado, E. Z. A., Vasconcelos, I. R., Amorim, K. M., Andrade, A. M. S., Barreto, L. P., and Santos, D. A. (2010). Uma Experiência em Escolas de Ensino Médio e Fundamental para a Descoberta de Jovens Talentos em Computação. *Anais do XXXI Congresso da SBC. XVIII Workshop sobre Educação em Computação*, 1:798–807.
- Maloney, J., Resnick, M., Rusk, N., Silverman, B., and Eastmond, E. (2010). The Scratch Programming Language and Environment. *ACM Transactions on Computing Education*, 10(4):1–15.
- Observatório Softex (2012). *Software e Serviços de TI: A indústria Brasileira em Perspectiva*, volume 2.
- Patil, S. P. and Goje, A. C. (2009). The Effect of Developments in Student Attributes on Success in Programming of Management Students. *2009 International Conference on Education Technology and Computer, ICETC 2009*, pages 191–193.
- Pereira Jr, J. C. R., Rapkiewicz, C. E., Delgado, C., and Xexeo, J. A. M. (2005). Ensino de Algoritmos e Programação: Uma Experiência no Nível Médio. In *Anais do XXV Congresso da SBC. XIII Workshop sobre Educação em Computação*, pages 2351 – 2362.
- Resnick, M., Maloney, J., Hernández, A. M., Rusk, N., Eastmond, E., Brennan, K., Miller, A., Rosenbaum, E., Silver, J., Silverman, B., and Kafai, Y. (2009). Scratch: Programming for All. *Communications of the ACM*, 52(11):60–67.
- Scaico, P. D., de Lima, A. A., Barbosa Belo da Silva, J., Azevedo, L. F. P. S., Raposo, E. H. S., Alencar, Y., and Mendes, J. a. P. (2012). Programação no Ensino Médio: Uma Abordagem de Ensino Orientado ao Design com Scratch. In *Anais do Congresso Brasileiro de Informática na Educação. XVIII Workshop de Informática na Escola*, number Sudol 2011, pages 1–10.
- Scaico, P. D., Lima, A. A. d., Silva, J. B. B. d., Azevedo, S., Paiva, L. F., Raposo, E. H., Alencar, Y., Mendes, J. a. P., and Scaico, A. (2013). Ensino de Programação no Ensino Médio: Uma Abordagem Orientada ao Design com a Linguagem Scratch. *Revista Brasileira de Informática na Educação*, 21(2):92–103.
- Wing, J. M. (2006). Computational Thinking. *Communications of the ACM*, 49(3):33–35.