

A UTILIZAÇÃO DA FERRAMENTA *SCRATCH* COMO AUXÍLIO NA APRENDIZAGEM DE LÓGICA DE PROGRAMAÇÃO

Valter dos Santos Mendonça Neto

Departamento de Computação –Faculdade Pitágoras
Av. Daniel de La Touche, 23 - Cohama São Luís - MA, 65045-250

valternetosnt@gmail.com

Abstract. *Teaching the art of thinking in a structured way, ie, programming is the biggest challenge in teaching computer science courses. In this context, this article has as main focus presents a study that aims to evaluate the use of the Scratch software on the introductory environment of teaching in the Logic programming subject has as main focus presents a study that aims to evaluate the use of the Scratch software on the introductory environment of teaching in the Logic programming subject. The research was carried out with students from the classes of this year of a Computer Technician Course of Technical Education Institution after the identification of learning disabilities in the subject in previous classes in 2012. The results obtained show that the use of the tool in conjunction with an inspiring methodology brought favorable results in student learning.*

Resumo. *Ensinar a arte de pensar de forma estruturada, ou seja, programação é um dos grandes desafios no ensino de cursos da área de informática. Neste contexto, este artigo tem como foco principal apresentar um estudo que objetiva avaliar a utilização do software Scratch no ambiente introdutório de ensino da disciplina de Lógica de Programação. A pesquisa foi realizada com alunos das turmas do presente ano do curso Técnico de Informática de uma Instituição de Ensino Técnico após a identificação da dificuldade de aprendizagem na disciplina em turmas anteriores no ano de 2012. Os resultados obtidos mostram que o uso da ferramenta em conjunto com uma metodologia motivadora, trouxe resultados bastante favoráveis no aprendizado dos alunos.*

1 Introdução

Na grade dos cursos da área da informática, a disciplina de Lógica de Programação, ganha destaque por ser pré-requisito para as demais de desenvolvimento de *software*. Portanto, para a formação do profissional, é importante que, enquanto aluno, consiga compreender o que é ensinado nela.

Mas a disciplina tem se mostrado um grande obstáculo para os alunos da área. Muitos tem tido dificuldades em entender a lógica computacional. Rodrigues (2004) afirma que a disciplina tem um dos maiores índices de reprovação em todas as instituições de ensino brasileira e que esta realidade tem gerando um motivo de preocupação e reflexão por parte dos professores.

Sales e Dantas (apud ROCHA, 2010) mostram ainda que, os cursos de Lógica de Programação iniciam com uma média de 50 alunos e, em poucos meses, constata-se que as taxas de reprovação e/ou desistência chegam a 60%.

Assim, na tentativa de minimizar isso, várias soluções têm sido propostas e dentre elas destaca-se a utilização de ferramentas computacionais por possibilitar que os mesmos desenvolvam a capacidade de solucionar problemas, por meio do aprender fazendo.

O presente artigo visa apresentar a utilização de uma destas ferramentas computacionais, o *Scratch*, nos momentos iniciais da disciplina de Lógica de Programação com o intuito de desenvolver a interpretação e resolução de problemas propostos em programação.

A seguir, destaca-se o *software Scratch*. Apresenta-se o ambiente da ferramenta e as características que possibilitam sua utilização na disciplina de Lógica de Programação e sua aplicação por meio de um estudo de caso em turmas do curso Técnico de Informática em uma Instituição de Ensino.

Neste artigo mostra-se ainda as atividades desenvolvidas nas turmas, além dos dados coletados por meio da aplicação de questionários e da análise de documentos da instituição e os resultados e dificuldades encontradas com o uso da ferramenta. Após, relata-se as conclusões a respeito do estudo proposto neste artigo.

Portanto, por meio deste, espera-se contribuir na busca em fazer com que os alunos dos cursos de informática tenham um real aprendizado, uma vez que entender a lógica de programação será de extrema importância para seu futuro enquanto profissional da área.

2 Scratch

O *Scratch* é uma linguagem gráfica de programação desenvolvida no *Media Lab* do Instituto de Tecnologia de *Massachusetts*, baseada nas linguagens *Logo* e *Squeak*, mas que pretende ser mais simples, fácil de utilizar e mais intuitiva. Divulgada em maio de 2007, a ferramenta é um *software* gratuito, possui uma IDE em que não é preciso digitar funções ou endereços, é uma linguagem de programação visual. Foi criada com o propósito de introduzir a programação e a conceitos matemáticos, ao mesmo tempo em que incentiva o pensamento criativo, o raciocínio sistemático e o trabalho colaborativo (*SCRATCH*, apud PEREIRA et. al, 2012).

O termo *Scratch* provém da técnica de *scratching* utilizada pelos *Disco-Jockeys* do *Hip-Hop* que para realizar misturas musicais mais atrativas e inesperadas, giram os discos de vinil com as mãos para frente e para trás. Assim, com o *Scratch* é possível misturar mídias, tais como imagens, gráficos, sons e músicas, para a criação de histórias interativas, de jogos ou de animações, por exemplo.

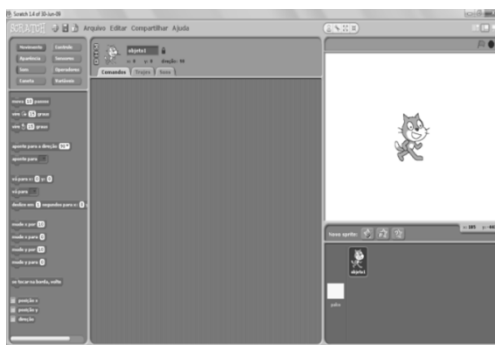


Figura 1. Tela principal da ferramenta Scratch

Mélo (2011) destaca o *Scratch* como uma excelente ferramenta para o ensino de conceitos de Lógica de Programação por possuir uma interface visual amigável e simples levando, de forma intuitiva, às principais estruturas de uma linguagem como: variáveis, operadores, estruturas de decisão e de repetição, e outros.

Por meio da ferramenta é possível aproximar o usuário cada vez mais do ambiente de programação, sem que haja necessariamente a necessidade de aprender uma linguagem de programação específica. Além disso, por não trabalhar com linhas de código (usa-se somente interface), possibilita a criação de programas de maneira mais simples e dinâmica, além de estimular o raciocínio lógico, e de permitir visualizar graficamente a execução do programa criado.

Desta forma, considerando o público-alvo e com o intuito de trabalhar em um ambiente motivador para os alunos neste momento introdutório em programação, por meio do desenvolvimento de programas com uma interface gráfica, adotou-se o *Scratch* como ferramenta de trabalho na disciplina de Lógica de programação.

3 Descrição do estudo

Para melhor entendimento dos resultados sobre a proposta em questão, fez-se necessário um estudo de campo, uma vez que a pesquisa possui a finalidade de testar, analisar e interpretar os dados colhidos para confirmar de forma coerente e crítica o tema abordado.

A pesquisa propõe a utilização da ferramenta *Scratch* nas primeiras aulas de Lógica de Programação como apoio no ensino da disciplina.

Durante o estudo, foram coletados os dados referentes à assimilação do conteúdo, conhecimentos e habilidades adquiridas, trabalhos desenvolvidos e reprovações utilizando o formulário de avaliação de turma realizado pela instituição e a aplicação de questionários.

Desta maneira, a pesquisa de campo mostra-se essencial, pois possibilita a comprovação dos resultados obtidos com coleta e análise dos dados gerados.

3.1 Caracterização da pesquisa de campo

O estudo foi realizado em uma instituição de ensino, situada na cidade de São Luís, com alunos, que tem entre 16 a 17 anos de idade, do curso Técnico em Informática em que 67% das disciplinas trabalham com ênfase prática em programação, análise e desenvolvimento de sistemas.

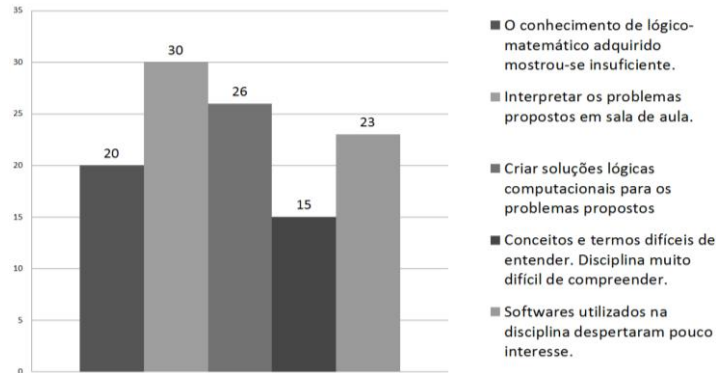
3.2 Observação em sala de aula: turma do curso técnico de 2012

Em 2012, a instituição iniciou quatro turmas, 63 alunos, nos turnos manhã e tarde. Nessas, a disciplina de Lógica de Programação foi ministrada da seguinte forma: 15 aulas com exposição da teoria, resolução de exercícios, com o aumento gradativo da dificuldade, utilizando o papel e/ou o aplicativo Visualg.

Durante as aulas identificou-se que os alunos, em quantidade bastante expressiva, apresentavam grandes dificuldades na compreensão e aplicação dos conceitos de programação.

Para entender melhor esta problemática, um questionário foi aplicado com somente 53 alunos, pois os demais faltaram no dia da aplicação (3) ou não estão mais na instituição (7). Sendo assim, a análise de dados baseou-se nas respostas dos alunos que tiveram concepções parecidas com algumas exceções.

Gráfico 1. As principais dificuldades na disciplina de Lógica de Programação



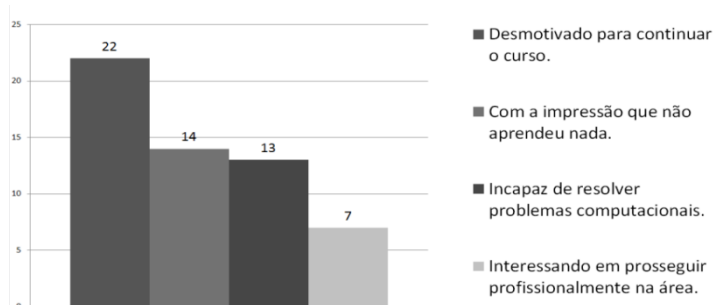
Analisando o gráfico 1, verificou-se que 30 alunos mostraram ter dificuldade de interpretar os problemas propostos em sala. Pode-se associar isto ao fato de 20 considerarem o conhecimento lógico-matemático adquirido insuficiente, uma vez que acostumados a decorar o conteúdo ministrado, não conseguiam desenvolver o raciocínio lógico exigido pela disciplina.

Em decorrência disto, 26 alunos não conseguiam criar soluções lógicas computacionais para os problemas propostos. Em alguns casos, esta dificuldade era agravada pelo fato dos alunos considerarem os conceitos e termos difíceis de entender (15).

Destaca-se ainda, que 23 alunos consideraram que o *software* utilizado na disciplina (Visualg) despertou pouco interesse. Alguns afirmaram que a ausência de uma interface gráfica ou algo que se aproximasse dos aplicativos usados por eles no dia-a-dia, mais interativo, gerava um descontentamento e um desânimo, em outras palavras, não os empolgava o suficiente para um melhor aproveitamento da disciplina.

As consequências destas dificuldades, apontadas pelos alunos, podem ser visualizadas no gráfico 2 abaixo.

Gráfico 2. Consequências após a disciplina



Observa-se que, 22 alunos finalizaram a disciplina desmotivados com o curso. Este sentimento era visível durante as aulas, em que os mesmos mostravam pouca ou nenhuma iniciativa.

Verifica-se ainda que, 14 responderam que ao finalizar a disciplina saiu com a impressão que não aprendeu nada e 13 disseram ser incapaz de resolver problemas computacionais.

Outro ponto a ser verificado é o número de reprovações na disciplina e na subsequente, Técnicas de Programação, uma vez que, para a aprovação dessa, o conhecimento da primeira é fundamental. Dos 63 alunos, somente 1 não conseguiu aprovação. Deste quadro, 50% dos aprovados só conseguiram tal resultado pelo fato da instituição possuir como política a recuperação paralela, ou seja, oferecer todas as possibilidades para que o aluno não reprove.

Os resultados da disciplina seguinte foram mais preocupantes. Depois de esgotadas todas as possibilidades, 22 dos alunos reprovaram, sendo que em uma das turmas o índice de reprovação foi de 50% dos alunos.

Portanto, diante dos dados expostos, como proposta para solucionar ou amenizar as dificuldades enfrentadas pelo aluno na disciplina de lógica de programação e com o propósito de trabalhar um ambiente mais motivador, adotou-se a ferramenta de trabalho *Scratch* nos momentos iniciais da disciplina.

3.3 Aplicação da ferramenta em sala de aula: turma do curso técnico 2013

Em 2013 a instituição iniciou novas turmas, nos turnos manhã e tarde. Nestas, 36 alunos se submeteram a essa nova abordagem. Assim a disciplina foi estruturada da seguinte forma: das 15 aulas, 5 foram destinadas ao ensino da ferramenta, dos conceitos básico (operadores, variáveis, comandos de entrada e saída, etc.) e comandos de seleção e repetição e 10 destinadas ao ensino da disciplina com o Visualg.

É importante destacar que o objetivo da ferramenta foi introduzir os conhecimentos de programação. Além disso, após cada aula os alunos eram motivados a continuarem praticando levando problemas para serem resolvidos em casa.

A seguir, detalha-se o planejamento das aulas ministradas na disciplina de Lógica de programação com o *Scratch*.

3.3.1 Planejamento das aulas

a) Aula 01 - Introdução a Lógica de Programação.

Nessa aula, os alunos são apresentados ao conceito de lógica de programação e o de algoritmos como solução de situações a serem seguidos passo a passo. Em seguida, apresenta-se a ferramenta *Scratch* e o seu ambiente e comandos de maneira geral.

b) Aula 02 – Conceitos iniciais de Programação

Nessa aula são apresentados, de forma mais detalhada, os conceitos iniciais de programação na ferramenta *Scratch*: comandos de entrada e saída de dados, operadores (aritméticos, relacionais e lógicos), variáveis e atribuição de dados.

c) Aula 03 – Criando programas no *Scratch*

Esta aula é dedicada à resolução de problemas utilizando a ferramenta. Mostramos os comandos dos grupos ainda não utilizados como aparência, som e caneta.

d) Aula 04 – Criando programas no *Scratch*: estruturas de decisão

Nesta aula mostra-se as estruturas de seleção presentes na ferramenta (comandos se, se...senão, quando ouvir, e outros).



Figura 2. Aula 04 – Cálculo da média

Como exemplo, destaca-se o tradicional problema do cálculo da média, na figura 2 acima, seguido da situação (aprovado ou reprovado) de um aluno, utilizando o comando se...senao, só que em um contexto diferente. Neste o usuário digita as notas do aluno e o professor (intitulado de Burro Inteligente) realiza o cálculo e fornece a situação do aluno, que muda de estado (imagem) de acordo com o resultado.

e) Aula 05 – Criando programas no *Scratch*: estruturas de repetição

Na última aula mostra-se as estruturas de repetição oferecidas pela ferramenta (sempre, repita, repita...até, e outros).

Ao final destas 05 aulas, desafiamos os alunos a produzirem um jogo utilizando todos os conhecimentos adquiridos. A proposta despertou nos alunos motivação, excitação e o desejo de produzir o melhor programa que pudessem fazer.



Figura 3. Jogo do aluno: Exemplo1

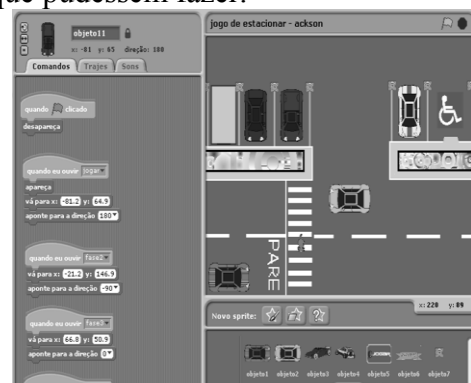


Figura 4. Jogo do aluno: Exemplo2

Na figura 3, visualiza-se um dos jogos produzidos, cujo objetivo é coletar as medalhas suspensas, fugindo do dragão e tomando cuidado para não cair no fogo.

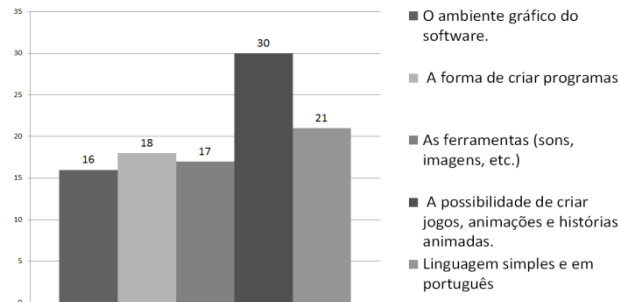
Na figura 4, mostra-se outro exemplo. Neste o objetivo é estacionar o carro, sem bater nas barras, na vaga do estacionamento, tomando cuidado com os outros veículos.

As demais aulas foram ministradas utilizando o Visualg como aplicativo para resolução de problemas propostos em sala. Neste segundo momento, sempre que se apresentava um conceito se fazia associações com o que foi visto anteriormente com o *Scratch*, no intuito de facilitar a compreensão e mostrar que posteriormente, quando estivessem aprendendo uma linguagem de programação, eles usariam os mesmos conceitos adaptando somente as formas e particularidades da mesma.

3.3.2 Questionário aplicado aos alunos

Para verificar os resultados obtidos com o uso o *Scratch* aplicou-se, após a disciplina, um questionário com 8 questões, sendo que neste tópico analisa-se somente 6.

Gráfico 3. Características que mais atraíram os alunos

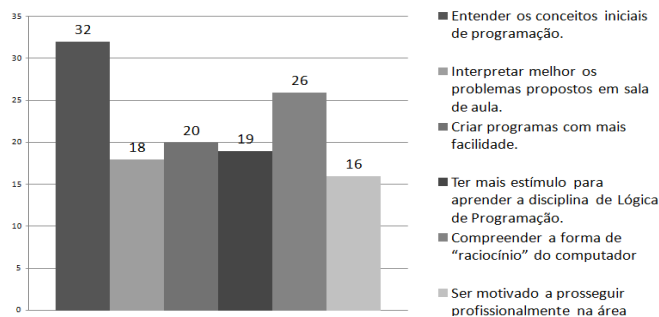


No gráfico 3, o fato do aplicativo possuir tradução para o português facilitou a sua aceitação e utilização por parte dos alunos (21). Em seguida, a forma de criar programas por meio do encaixe de blocos atraiu os alunos (18), com relação a este aspecto, durante a pesquisa, verifica-se duas particularidades:

- Alguns alunos, mesmo com a facilidade no método de programação, preferiam programar usando linha de código por considerarem que desta forma entendiam melhor o conteúdo e que essa maneira de programar se assemelhava mais com o que é, no entendimento deles, programar, além de ser a forma que eles utilizariam nas demais disciplinas do curso.
- Identificamos que alguns estudantes tiveram maior desempenho na disciplina no primeiro momento, programando com o *Scratch*, que no segundo utilizando o Visualg. Questionados a respeito alegaram que com a ferramenta era mais simples e divertido criar programas.

Outro ponto observado pelos alunos é o fato de poder utilizar ferramentas como som e imagem, por exemplo, associado ao ambiente gráfico, com blocos coloridos que são encaixados e a possibilidade de, utilizando esses recursos, criar jogos, animações e histórias animadas. Estes aspectos geraram um clima mais divertido e descontraído durante as aulas e curiosidade e frustração por parte dos alunos de outras turmas, que estão finalizando o curso técnico em informática este ano, que desejaram ter aprendido a programar utilizando esta nova abordagem. Além disso, criou expectativas e antecipação por parte de turmas que ainda teriam a disciplina, tanto que no momento da mesma, os alunos já estavam familiarizados com ferramenta.

Gráfico 4. Habilidades adquiridas pelo uso da ferramenta



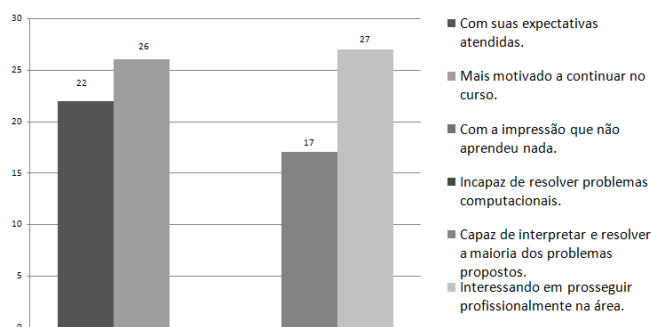
Verifica-se no gráfico 4 que, 32 alunos conseguiram entender os conceitos iniciais de programação, 26 a forma de "raciocínio" do computador e 20 afirmaram que passaram a criar programas com mais facilidade - pontos importantes almejados neste trabalho.

Além disso, 19 alunos se sentiram mais estimulados em aprender a disciplina de Lógica de Programação, algo que era notório entre os alunos durante as aulas.

Outro ponto dos dados é que 18 alunos se consideraram capaz de interpretar melhor os problemas propostos em sala de aula, amenizando assim, o quadro das turmas passadas observado anteriormente no gráfico 1.

Percebe-se também que, durante as aulas, aqueles que já possuíam o interesse em prosseguir profissionalmente na área de informática (16) formam mais estimulados, além de ter despertado outros a almejamem o mesmo, principalmente em programação de jogos.

Gráfico 5. Consequências após a disciplina



De acordo com gráfico 5, quando perguntados sobre como se consideravam após a disciplina, 27 alunos responderam que se sentiam mais interessados em seguir carreira na área de informática, 26 ficaram mais motivados em prosseguir com o curso técnico e com mais expectativas sobre ferramentas e metodologias que seriam utilizadas nas demais disciplinas do curso, 22 com as expectativas atendidas (desejavam aprender a criar programas) e 17 capaz de interpretar e resolver a maioria dos problemas propostos (quanto mais desafiador o problema, mais entusiasmados os alunos ficavam).

Outro dado importante a destacar foi que nenhum aluno marcou as opções: com impressão que não aprenderam nada e incapaz de resolver problemas computacionais. Isto pode ser reforçado quando verificamos que todos os alunos foram aprovados na disciplina.

Assim, observou-se que houve uma melhora expressiva, quando se compara com os dados das turmas passadas. Principalmente quanto à motivação dos alunos em prosseguir com o curso.

Quando perguntados se a ferramenta ajudou na compreensão da disciplina de Lógica de Programação, apenas 1 aluno marcou a opção não na pesquisa. Ao ser questionado sobre a negativa nas respostas, o mesmo justificou dizendo que não gostou de desenvolver jogos, por não ter afinidade para tal, e que se não se adaptou a forma de programar do *software*.

4 Desvantagens e dificuldades do uso da ferramenta *Scratch*

Para identificar e analisar as desvantagens e dificuldades do uso da ferramenta, incluí-se 2 perguntas no questionário aplicado aos alunos das turmas de 2013, após a disciplina de Lógica de programação.

Dentre as características apontadas na pesquisa como desvantagens destaca-se:

- *Imagens*: neste aspecto, citado por todos, os alunos acentuaram as opções disponibilizadas pelo *software*. Consideraram a qualidade de algumas imagens baixa, além de limitado quanto a opções e quantidades. Além disso, o editor de imagens

possui poucos recursos para tratamento de imagens e os disponibilizados são restritos e trabalhosos de serem utilizados.

- *Áudio*: os alunos demonstraram certa dificuldade para utilizar este recurso. Alguns arquivos não tocavam na ferramenta, mesmo nos formatos compatíveis. Além disso, o fato de não possuir um editor de áudio foi um ponto negativo evidenciado pelas turmas.
- *Vídeos*: a incompatibilidade com vídeos foi uma particularidade vista por alguns como negativo.
- *Baixo desempenho com muitos recursos*: colocar muitos recursos (imagens, sons, músicas) deixa o programa pesado e dependendo da capacidade da máquina utilizada, os comandos ficam lentos e acabam travando.
- *Utilizar muitos blocos de comando para realizar algo*: no *Scratch* em alguns casos, dependendo do que se deseja fazer, necessita da junção de muitos blocos de comando, tornando a programação trabalhosa para os alunos.

Com relação às dificuldades encontradas pelos alunos, além das identificadas acima, destaca-se as relacionadas com criar programas na ferramenta.

Durante as aulas percebeu-se que alguns alunos, uma parcela pequena, tinham problemas em criar programas no *Scratch*. Para ser mais específico possuíam dificuldade em saber qual comando usar para criar algo ou até mesmo para desenvolver a solução do problema. No questionário aplicado alguns relatam, por exemplo, que se confundiam na escolha do bloco de comando (diga ou pergunte) quando precisavam ler e armazenar uma informação do usuário. Além disso, relataram problemas em utilizar alguns comandos como os associados à estrutura de decisão e repetição e os operadores.

Outro ponto citado foi colocar os objetos em cena. Programar no *Scratch* é encaixar blocos de comandos associados a objetos de forma lógica, ou seja, cada um deles possui sua própria programação e é preciso sincronizar as suas ações, pois estão todos no palco.

Ao término da disciplina, com os exercícios criados e o desenvolvimento dos jogos como projeto final, as dificuldades foram amenizadas e até mesmo sanadas.

Outro fato a considerar sobre a ferramenta é o de não poder ser utilizada em todos os assuntos, ou seja, trabalhar somente ela durante a disciplina. Um dos fatores é a sua forma de programar conectando blocos, que não se assemelha a forma mais comumente utilizada por outros *softwares* e o outro está associado à ausência de algumas estruturas como vetor unidimensional e multidimensional (matriz) e registro, por exemplo. Mesmo possuindo o bloco de comando lista (não incluído na pesquisa), conhecer estas estruturas é de extrema importância para desenvolver programas mais completos.

5 Conclusão

Existem diversas propostas na busca em melhorar o processo de ensino e aprendizagem dos conceitos iniciais da programação de computadores. A utilização de ferramentas computacionais (o *Scratch*, por exemplo), aliada a uma metodologia que gere uma motivação maior nos alunos em praticar a resolução de problemas, tem se mostrado eficiente.

Tornou-se possível verificar por meio deste trabalho que utilizando a ferramenta *Scratch* como uma nova abordagem, auxiliando no processo de aprendizagem dos alunos nos conceitos iniciais da disciplina de Lógica de Programação trouxe resultados bastante favoráveis.

Confirmou-se que o ambiente interativo e descontraído de programação proporcionado pelo *Scratch* torna a programação atraente, fácil e mais intuitiva, uma vez que os alunos

podem inserir em seus programas elementos como sons e imagens por eles mesmos produzidos ou não.

Percebeu-se também que, o *software* não pode ser usado de forma isolada, sem uma estratégia de ensino, ou seja, sem um planejamento de aulas interessantes e motivadoras que associado à ferramenta despertaram um interesse maior no aluno e tampouco sozinho como única ferramenta de aprendizagem devido a ausência de algumas estruturas importantes na programação.

Assim, o uso do aplicativo no início da disciplina de Lógica de Programação, além de ter motivado mais os alunos, apesar das desvantagens e dificuldades encontradas por eles na interação com a ferramenta, pode facilitar o entendimento das estruturas e comandos (alguns, considerados difíceis pelos alunos) que foram aprofundados com a utilização do *software* Visualg.

Portanto, a utilização do *Scratch* na disciplina com o objetivo de apresentar, de maneira diferenciada, o mundo da programação para alunos iniciantes nos cursos da área de informática pode amenizar o problema pertinente à compreensão da lógica computacional e contribuir de forma significativa para a formação do profissional da área.

Referências

- MÉLO, F. E. N., et. al (2011). Do *Scratch* ao Arduino: Uma proposta para o ensino introdutório de programação para cursos superiores de tecnologia. XXXIX Congresso Brasileiro de Educação em Engenharia, Blumenau, SC. Disponível em <www.abenge.org.br/CobengeAnteriores/2011/sextoestec/art1886.pdf> Acesso em 15 de fevereiro de 2013.
- PEREIRA, P. S.; et. al(2012). Análise do *Scratch* como ferramenta de Auxílio ao Ensino de Programação de Computadores. Instituto Federal de Educação, Ciência e Tecnologia do Ceará, Fortaleza – CE. Disponível em <www.abenge.org.br/CobengeAnteriores/2012/artigos/104281.pdf>. Acesso em 19 janeiro2013.
- ROCHA, P. S. et. al (2010). Ensino e aprendizagem de programação: Análise da aplicação de proposta metodológica baseada no Sistema Personalizado de Ensino CINTED-UFRGS Novas Tecnologias na Educação, Rio Grande do Sul. Disponível em:<<http://seer.ufrgs.br/renote/article/view/18061/10649>> Acesso em 21 abril 2013.
- RODRIGUES, M. (2004). Experiências positivas para o ensino de algoritmos. IV Escola Regional de Computação Bahia-Sergipe. Feira de Santana. Disponível em: <<http://www.uefs.br/erbase2004/documentos/weibase/Weibase2004Artigo001.pdf>> Acesso em 19 janeiro 2013.
- SCRATCH. ABOUT Scratch (Scratch Documentation Site).Disponível em:<http://info.scratch.mit.edu/About_Scratch>. Acesso em: 14 janeiro 2013.