

Prática de ensino de Programação de Computadores com Robótica Pedagógica e aplicação de Pensamento Computacional

Humberto Augusto Piovesana Zanetti¹, Cláudio Luís Vieira Oliveira²

¹Escola Técnica Rosa Perrone Scavone – Centro Paula Souza – Itatiba, SP

²Faculdade de Tecnologia de Jundiaí – Centro Paula Souza – Jundiaí, SP

humberto.zanetti@etec.sp.gov.br, prof.claudioluis@fatec.sp.gov.br

***Abstract.** This paper discusses a teaching practice for computer programming through Educational Robotics and Computational Thinking. This paper aims to bring an educational proposal capable of reducing the main difficulties of beginners in programming, assisting in the construction aspects of troubleshooting.*

***Resumo.** Este artigo discute sobre uma prática pedagógica para ensino de programação de computadores por meio de Robótica Pedagógica e Pensamento Computacional. Tem como objetivo trazer uma proposta de ensino que possa amenizar as principais dificuldades de alunos iniciantes em programação, auxiliando nos aspectos relacionados a construção da solução de problemas.*

1. Introdução

Alunos ingressantes em cursos da área da Computação, tanto no nível técnico ou superior, quando em contato pela primeira vez com disciplinas introdutórias de programação de computadores sentem dificuldades, o que muitas vezes promove a desmotivação logo no início do curso. Essas disciplinas têm como objetivo trabalhar o raciocínio lógico, abstração e apresentar uma linguagem de programação com uma sintaxe específica. É importante destacar que começar a ensinar detalhes sintáticos da linguagem antes que os alunos percebam quais são as finalidades e utilidades dos conceitos de programação pode potencializar essa desmotivação. Como aponta o trabalho de Gomes, Henriques e Mendes (2008), essas disciplinas são responsáveis por altos índices de reprovações, trazendo consequências negativas prematuramente em toda a formação do aluno.

O processo de aprendizagem dos conceitos iniciais de programação é complexo e, muitas vezes, necessita de um nível de abstração que não está presente na maioria dos alunos iniciantes, havendo a necessidade de se criar um ambiente mais diversificado e motivador para o aluno. Benitti (2012) discute que entre as soluções e ferramentas possíveis, se destaca a Robótica Pedagógica (RP). Adotar a RP como diferencial de aprendizagem proporciona um vasto conjunto de recursos para o desenvolvimento de práticas pedagógicas mais estimulantes que as tradicionais adotadas no ensino de programação.

Em relação ao desenvolvimento do raciocínio computacional (algorítmico) necessário aos alunos em disciplinas de programação, é relevante adotar um método no

qual seja possível apresentar e traduzir as noções de algoritmo dentro da disciplina. A capacidade de abstração é algo fundamental para o sucesso na aprendizagem de programação, principalmente para compreender problemas e propor soluções. A adoção do Pensamento Computacional pode orientar de forma ampla a atividade mental de abstrair problemas e formular soluções descritas em algoritmos. O termo Pensamento Computacional (em inglês *Computational Thinking*) foi introduzido por Wing (2008), com o objetivo de englobar desde a estruturação do raciocínio lógico, até o comportamento humano para a ação de resolução de problemas.

Assim, este trabalho traz uma proposta metodológica que delinea um contexto de aprendizagem significativa de programação de computadores suportada por uma linguagem de programação visual para o desenvolvimento do Pensamento Computacional e aliada a ferramentas de RP. Também apresenta um estudo realizado através de uma oficina e prática com alunos matriculados em um curso de ensino médio integrado ao técnico de Informática, utilizando uma ferramenta de programação visual S4A (*Scratch 4 Arduino*) e um robô controlado pela plataforma eletrônica Arduino. Além do estudo, o artigo ainda apresenta uma revisão bibliográfica sobre o uso de RP e Pensamento Computacional no contexto do ensino de programação de computadores.

O artigo apresenta na Seção 2 uma revisão bibliográfica sobre os desafios no ensino de programação e apresenta uma breve descrição sobre RP e suas aplicações; na Seção 3 é discutida a abordagem pedagógica baseada em Pensamento Computacional para resolução de problemas; a Seção 4 apresenta a prática e maiores detalhes da oficina com os alunos; a Seção 5 apresenta a análise dos resultados; e por fim, na Seção 6 são apresentadas as considerações finais.

2. Robótica Pedagógica e os desafios no ensino de programação de computadores

A aprendizagem de programação de computadores é um processo difícil, sendo necessário um alto nível de abstração e compreensão de linguagens com sintaxe complexa. Gomes *et al.* (2008) salienta que falta aos alunos iniciantes competências necessárias para resolução dos problemas. Métodos tradicionais de ensino de programação utilizam fluxogramas e pseudocódigos para demonstração de conceitos abstratos, sem se importar com o processo de construção da solução. Outro desafio no ensino de programação, apontado por Gomes *et al.* (2008), é a pertinência da motivação em um ambiente de aprendizagem. Muito do que é lecionado em sala de aula utiliza métodos de estudos baseados em leituras sucessivas, memorização e mecanização de procedimentos. A mecanização dos processos na aprendizagem não apresenta os porquês da utilização de certo conceito de programação.

O ensino de programação de computadores tem como propósito fazer com que os alunos desenvolvam um conjunto de competências necessárias para conceber sistemas computacionais capazes de resolver problemas reais. Entretanto, em termos gerais, existe uma grande dificuldade em compreender e aplicar certos conceitos abstratos de programação, por parte de uma porcentagem significativa dos alunos que ingressam em disciplinas introdutórias nesta área. Uma das grandes dificuldades reside precisamente na compreensão e, em particular, na aplicação de noções básicas, como as estruturas de controle à criação de algoritmos que resolvam problemas concretos [Gomes *et al.* 2008].

No aspecto teórico da Educação, Papert (1985) aponta que a utilização de robôs como

artefatos educacionais tem um grande potencial para promover um ambiente favorável ao aprendizado dentro da sala de aula. Benitti (2012) reforça que a presença da RP durante as aulas pode trazer experiências educacionais mais completas e um retorno mais rápido de conceitos aprendidos pelos alunos devido ao fator motivador presente. Rusk et. al. (2008) aponta que a RP pode trazer uma interdisciplinaridade interessante, explorando uma ampla gama de possíveis aplicações para envolver os jovens com diferentes interesses.

Jovens geralmente não estão interessados em abordagens tradicionais, tornando-se motivados quando as atividades de RP são introduzidas como uma maneira de conectar outras disciplinas ou áreas de interesse. Com a utilização da RP, além das questões de motivação e aquisição de conhecimento, torna-se possível construir um ambiente de trabalho colaborativo e atividades interpessoais, o que muitas vezes é impossível com instrumentos educacionais convencionais. A RP trabalha com competências, além daquelas propostas pelos currículos escolares, oferecendo aos alunos experiências reais das áreas da informática, eletrônica, mecânica e design, através da resolução de um determinado problema ou desafio [Papert 1985].

Conforme o trabalho de Zanetti e Bonacin (2014), com a adoção da RP é possível observar o impacto de instruções em um objeto concreto e associar os comandos abstratos aos movimentos e ações do mesmo. A utilização da RP contribui para o desenvolvimento de competências necessárias para alunos iniciantes em programação, como: raciocínio lógico, representação e comunicação, resolver problemas por meio de erros e acertos, aplicação das teorias formuladas em atividades práticas e capacidade crítica [Zilli 2004].

3. Pensamento Computacional como modelo de solução de problemas

Disciplinas como a programação de computadores ou Lógica de Programação permite realizar um encadeamento coerente de uma sequência lógica de instruções para resolução de problemas, constituindo um dos saberes elementares para desenvolver o Pensamento Computacional. Como supracitado, usualmente o conteúdo relacionado à disciplina é ensinado de forma isolada, não se preocupando com a contextualização dessas instruções ou a correlação com outras áreas de conhecimento. Nesse sentido, o aluno não compreende a aplicabilidade do conhecimento que é ensinado, tornando o processo de aprendizagem não significativo.

O Pensamento Computacional não se limita às ferramentas, como se fosse totalmente dependente das tecnologias [de Paula, Valente e Burn 2014]. Está mais próximo a um pensamento analítico, que compartilha características com o pensamento matemático (habilidades necessárias para soluções de problemas), pensamento sistêmico (projetar e avaliar sistemas complexos que operam sob restrições específicas) e pensamento científico (conhecimento teórico sobre o assunto, análise de resultados e senso crítico). Portanto, se trata de uma maneira específica de se pensar e de se analisar uma situação ou artefato, sendo influenciado por todas as áreas de conhecimento e diferentes tecnologias [Wing 2008; de Paula, Valente e Burn 2014].

Nesse contexto, é válido afirmar que o Pensamento Computacional é altamente requerido no repertório de habilidades dos alunos em disciplinas relacionadas à programação de computadores. O termo “computacional” não pode ser compreendido como sinônimo de “programação”, mas sim como uma maneira de se compreender os sistemas computacionais, como eles funcionam e como são projetados e programados. A

programação não é o fim, mas sim um dos meios para se atingir os objetivos [de Paula, Valente e Burn 2014]. Estudiosos na década de 1980, como Seymour Papert [Papert 1985], já afirmavam que era necessário promover o desenvolvimento do Pensamento Computacional junto aos alunos, argumentando sobre o pensamento analítico (chamado em seu trabalho de “pensamento procedimental”), utilizando a linguagem de programação Logo, que precedeu várias ferramentas pedagógicas de ensino de programação, incluindo ferramentas de RP.

Adotando esse argumento, nota-se que o Pensamento Computacional direciona a aprendizagem para algo mais amplo, o que favorece a apresentação de toda estrutura curricular e a abrangência dos saberes necessários nas disciplinas de programação de computadores. Se alinhado com o ferramental da RP, torna-se possível através desse meio menos abstrato que o tradicional, estimular a construção do conhecimento e o engajamento dos alunos, como uma maneira de se fomentar o raciocínio lógico, a habilidade cognitiva e a prática de solução de problemas. Também é importante ressaltar o argumento Construcionista atrelado à RP, que de acordo com essa teoria, o processo de aprendizagem é mais significativo e profundo, já que os alunos podem trabalhar em projetos e desafios que lhes são pessoalmente mais atrativos e, conseqüentemente, mais significativos [Rusk *et al.* 2008].

4. Aplicação da oficina e experimentação

Para esse trabalho foi realizado uma oficina e, posteriormente, uma prática com um grupo de alunos com o intuito de analisar o uso de RP e, para promover e estimular o Pensamento Computacional, direcionando a resolução de problemas, construção de conhecimento e autoavaliação. Há quatro objetivos bem definidos alinhados às seguintes dificuldades comumente encontradas em alunos iniciantes em programação: (i) avaliar o próprio conhecimento; (ii) compreender e interpretar problemas; (iii) resolver problemas aplicando o Pensamento Computacional e; (iv) depuração de erros e correção dos mesmos.

4.1. Ferramentas e plataformas utilizadas

O ambiente de programação S4A foi originado de um projeto anterior, o Scratch, desenvolvido pelo *Lifelong Kindergarten Group*, do MIT (*Massachusetts Institute of Technology*), que tem como objetivo proporcionar o desenvolvimento de artefatos digitais, como jogos e animações, de maneira fácil e intuitiva. Por se tratar de uma plataforma aberta, muitos projetos se originaram do Scratch, que no caso do S4A, foi adaptado para fazer interface com a plataforma Arduino [CITILAB 2015].

Nesse ambiente, o computador deve ser conectado diretamente à placa Arduino, tendo controle direto aos seus pinos, controlando os componentes ligados à placa, como sensores e motores. Esses controles são feitos através de um conjunto de blocos que são organizados e estruturados como uma sequência de instruções (algoritmo). A Figura 1 apresenta uma visão geral do ambiente (a) e mostra um exemplo de *script* de programação no S4A para piscar um diodo emissor de luz (*led*) conectado ao pino digital 13 da placa Arduino (b).

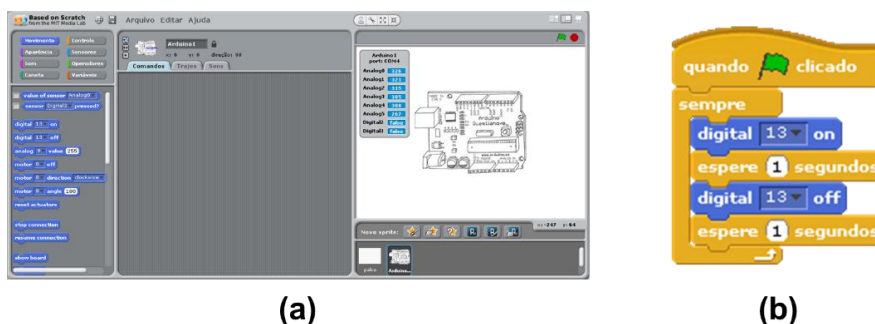


Figura 1. Visão geral do ambiente (a) e exemplo de script (b)

O projeto Arduino teve como objetivo criar uma plataforma de prototipagem e manipulação de componentes eletrônicos que fosse fácil de usar por pessoas que não tivessem conhecimentos técnicos em eletrônica, como, por exemplo, programadores e artistas. A plataforma Arduino pode ser compreendida como um pequeno computador no qual é possível programar rotinas de processo em saídas e entradas de dispositivos e componentes externos ligados a ele, podendo ser utilizado em projetos de Automação ou Robótica, por exemplo. A placa oferece uma solução de baixo custo, de simples aprendizagem e com arquitetura de *hardware* e *software* de código aberto para desenvolvimento de projetos interativos [McRoberts 2011].

Para o experimento deste trabalho foi utilizado um robô controlado por Arduino e constituído por dois motores (A – lado direito e B – lado esquerdo), responsáveis pela locomoção do robô, e por um sensor PIR (*Passive InfraRed*), localizado na parte da frente do robô, responsável pela detecção de obstáculos.

4.2. Metodologia adotada para a prática

A oficina e prática foram realizadas com seis alunos (duas alunas e quatro alunos) do curso Técnico em Informática integrado ao Ensino Médio, da Escola Técnica Rosa Perrone Scavone, localizada na cidade de Itatiba, São Paulo. Todos os alunos têm idade entre quatorze e quinze anos, todos atualmente no 1º ano do curso, cursando a disciplina Lógica de Programação e sem contato prévio com qualquer tipo de ferramenta de RP. O grupo é formado por alunos em processo de aprendizagem de programação e ainda são afetados pelas dificuldades relacionadas à disciplina. A prática foi realizada em umas das salas de aula da escola e teve duração de aproximadamente quatro horas.

A oficina apresentou os conceitos básicos e fundamentais das ferramentas utilizadas, como os componentes e estrutura física do robô. Houve uma breve explanação sobre o ambiente de programação do S4A, assim como quais eram os blocos que controlavam os motores e o que verificaria o valor do sensor PIR. Após a oficina, os alunos foram organizados em dois grupos e começaram a prática de desenvolvimento da solução dos desafios propostos. Em nenhum momento da oficina houve a demonstração parcial ou integral de qualquer solução, pois isso deveria ser feito pelos alunos posteriormente. A Figura 2 mostra a lousa com a descrição dos blocos (a), os exemplos dos blocos utilizados no S4A (b) e os alunos no momento do desenvolvimento da prática (c).

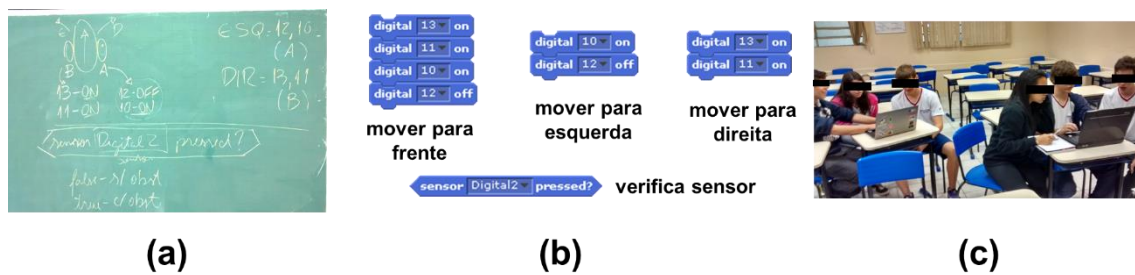


Figura 2. Lousa utilizada na oficina (a), blocos de exemplo (b) e alunos participantes (c)

A prática realizada teve como objetivo trabalhar três conceitos de programação: (1) entradas e saídas; (2) estrutura de seleção (referente à estrutura *if-else*); e estrutura de repetição (referente à estrutura *while* ou *do-while*). Para a execução da prática foi utilizada uma arena com um piso branco e marcações com fita preta, delimitando a área que era esperada para a locomoção do robô. A prática foi composta por dois desafios: (1) fazer com que o robô se movesse para frente até encontrar um obstáculo e; (2) que o robô se movesse para frente e quando encontrar o obstáculo, desviar. A Figura 3 abaixo mostra o desenho do primeiro desafio (a), o desenho do segundo desafio (b) e a arena física utilizada (c).

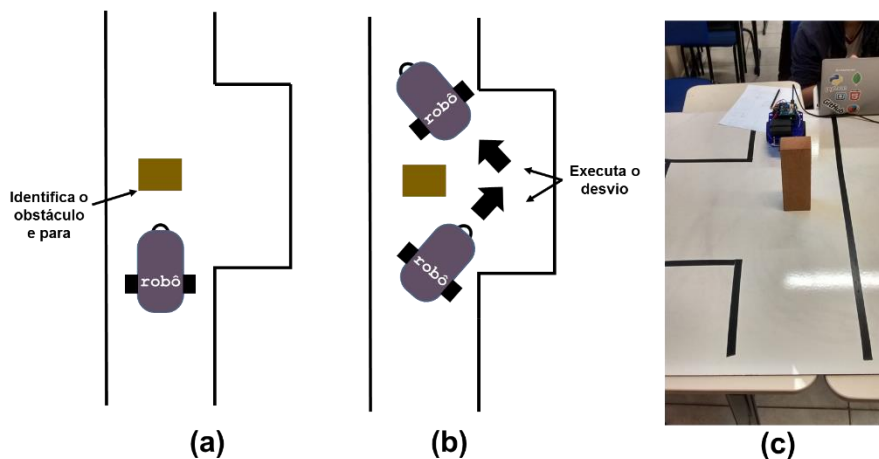


Figura 3. Desenho do desafio 1 (a), do desafio 2 (b) e visão da arena física com o robô (c)

Na prática adotada nesse trabalho, o aluno deve resolver o problema proposto refletindo sobre quais são as ações corretas, quais estruturas de programação são necessárias. E quando surgir problemas, o aluno pode testar e depurar um novo procedimento em busca de um resultado melhor.

O termo “depurar” ou “depuração” relacionado ao problema se baseia em Valente (2005), que afirma que a depuração implica uma nova descrição da solução e, assim, sucessivamente, repetindo o ciclo “descrição-execução-reflexão-depuração-descrição”, como vemos na Figura 4. O uso desse “ciclo de depuração” alinha-se com as diretrizes fundamentais do Pensamento Computacional, e é facilitada pela presença do ambiente de programação e do robô. Tais instrumentos facilitam a análise do programa de modo que o aluno possa achar seus erros. Este processo de achar e corrigir o erro constitui em uma oportunidade única para o aluno aprender sobre um determinado conceito envolvido na solução do problema ou sobre estratégias de resolução de problemas, além de poder, através da programação, relacionar seu pensamento em um nível metacognitivo [Valente, 2005].

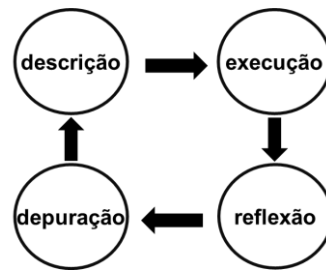


Figura 4. Ciclo proposto por Valente (2005)

Sob essa ótica do ciclo, cada versão da solução do aluno pode ser tida como uma explicitação do seu raciocínio e representação do conhecimento, através de uma linguagem formal. A construção da solução por parte dos alunos iniciou-se com o desenvolvimento de fluxogramas, como mostra a Figura 5(a), para mostrar o raciocínio empregado. Em seguida, houve início da solução no ambiente S4A, com sucessivas tentativas e testes, até a solução final, como mostra a sequência de blocos na Figura 5(b).

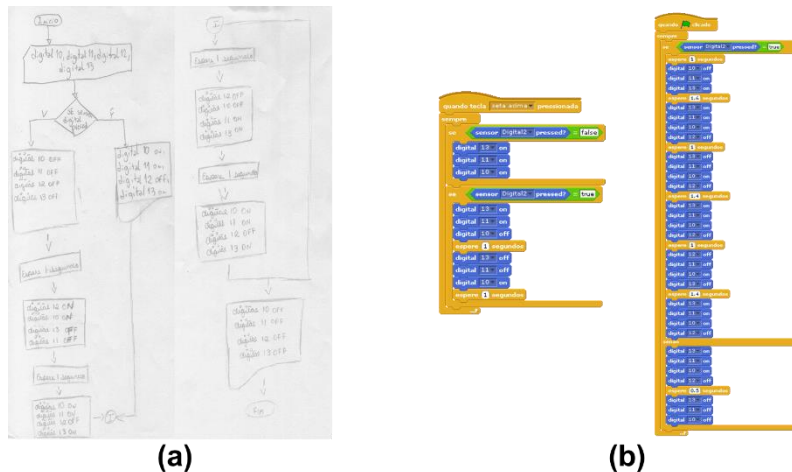


Figura 5. Exemplos de fluxogramas (a) e soluções programadas pelos alunos (b)

Após o término da prática, os alunos foram submetidos a um questionário para avaliar eficiência dessa prática. Os resultados são apresentados na próxima seção.

5. Análise dos resultados e discussão

Para averiguar os resultados da prática foi elaborado um questionário com uma questão relacionada ao nível de dificuldade que os alunos possuem na disciplina de Lógica de Programação; cinco questões de caráter quantitativo, relacionadas à realização da prática e percepção por parte dos alunos utilizando a escala Likert de respostas para especificar o nível de concordância com uma afirmação (questão) [Wainer, 2007] e; por fim, uma questão dissertativa apontando os pontos fortes e fracos, com caráter qualitativo.

Em relação à pergunta “Você possui dificuldades em Lógica de Programação¹?”, todos os alunos mostraram algum nível de dificuldade, ou seja, o grupo apresenta uma homogeneidade na questão de apresentar algum tipo de incerteza ou dificuldade

¹ “Lógica de Programação” foi o termo usado ao invés de “programação de computadores” por ser mais significativo aos alunos do experimento.

específica. Para o experimento a presença de dificuldades é positiva, pois alguma melhoria com a aplicação da prática pode ser notada mais facilmente.

As cinco questões de caráter quantitativo elaboradas foram: (Questão 1) “Em comparação com o método tradicional, este aplicado é mais motivador?”; (Questão 2) “Comparando com o método tradicional, os conceitos de programação são mais claros neste método aplicado?”; (Questão 3) “Você conseguiu construir a solução sem problemas?”; (Questão 4) “Você encontrou os erros facilmente e os corrigiu?” e; (Questão 5) “Você concorda que esse método de ensino ajuda na disciplina?”. A escala adotada foi: “1” sendo “discordo plenamente”, “2” sendo “discordo parcialmente”, “3” sendo “indiferente”, “4” sendo “concordo parcialmente” e “5”, sendo “concordo totalmente”.

Na Questão 1 houve uma maioria de respostas positivas, com três “concordo totalmente” (50%) e duas respostas “concordo parcialmente” (33%), demonstrando que o método proposto no trabalho é relevante e motivador para o processo de ensino e aprendizagem em relação ao tradicional. É possível notar que não há nenhum tipo de rejeição sobre a prática aplicada tendo apenas uma resposta “indiferente” (17%).

Avaliando em relação à clareza na identificação dos elementos² de programação (Questão 2) e como utilizá-los na solução, 50% dos alunos concordaram totalmente e 33% concordaram parcialmente que o método proposto tornava mais claro os componentes de programação. Houve apenas uma resposta como sendo indiferente (17%). Com esses resultados é possível afirmar que a presença da RP auxilia na abstração de conceitos de programação. Essa aceitação positiva pode ser resultado também da significância maior no processo de construção na solução, com alunos motivados a solucionar o problema através do processo de depuração, com tentativa e erros.

As respostas dadas à Questão 3 identifica um retorno predominantemente positivo em relação a facilidade encontrada em resolver os problemas, mas com possibilidade de melhorias. Com 67% de respostas “concordo parcialmente” e 33% de respostas “indiferente” é possível notar que ainda há obstáculos que prejudicam a solução do problema, possivelmente relacionados à introdução de uma ferramenta nova. Mas em uma visão geral, a solução do problema não sofre maiores contratempos em sua construção.

Em relação a Questão 4, temos 83% das respostas “concordo plenamente” e 17% “concordo parcialmente”, mostrando grande potencialidade na adoção da prática na ação de identificação e correção de erros. A forte presença da possibilidade de depuração facilitada surge como um bom indicador da eficácia da utilização da RP para criar um ambiente menos abstrato, auxiliando a construção da solução e verificação da mesma. Isso possibilita o aluno analisar seu programa em termos de efetividade das ideias, estratégias e estilo de resolução de problema, fazendo com que ele comece a pensar sobre suas próprias ideias, ou que faça sua “abstração reflexiva”, como propõe Valente (2005).

Por fim, a Questão 5 avalia a RP como sendo um instrumento facilitador, com 83% das respostas sendo “concordo plenamente” e 17% sendo “concordo parcialmente”. O resultado confirma a eficácia da RP como elemento motivador no processo de ensino-aprendizagem, capaz de motivar alunos iniciantes.

² “Elementos de programação” são usados como termo para englobar os conceitos de entrada e saída, estruturas condicionais, laços de repetição e qualquer outro elemento formal e definido necessário para a construção do programa.

A análise qualitativa baseia-se na questão dissertativa disposta no formulário, na qual aos alunos poderiam apontar pontos fracos e fortes da prática em questão. Entre as respostas de caráter positivo, destacam-se:

“O fato de colocar os conceitos de Lógica na prática facilitam o aluno a ver o que está acontecendo e também a ideia de cumprir um objetivo[...].”

“[...]talvez o principal, é a dinamicidade. O fato do "objeto" a ser programado ser um robô, é um grande incentivo pois é algo mais interessante de se mexer.[...]Os conceitos parecem ser mais claros já que é mais fácil imaginar como ele funciona.”

“Os pontos fortes foram que ficou mais claro para entender como é a programação é executada, e é um método mais simplificado de botar as ideias em prática.”

A partir das respostas supracitadas, é possível notar dois pontos fundamentais sobre o aspecto pedagógico: (1) a possibilidade de explorar uma solução e testá-la em um ambiente menos abstrato (utilizando o robô) e potencialmente mais dinâmico, com fácil identificação das ações e correção de erros, é importante para motivar o aluno à resolver problemas e; (2) o cenário de uso da RP e a exploração de um desafio incentivam os alunos a colocar suas ideias em prática, diminuindo a inibição ao propor alguma solução.

Entre as respostas de caráter negativo, destacam-se:

“Pouca precisão de sensibilidade do robô.”

“[...]eu diria que a eficiência do sensor não foi das melhores.”

“[...] na hora que as linhas de comandos ficam muito grandes aí fica muito confuso de achar onde está o erro.”

Através dessas respostas é possível notar a insatisfação momentânea das ações exercidas pelo robô, mesmo comprovada a ação motivadora que ele exerce. Problemas e inconsistências são esperadas por parte do robô, por se tratar de um instrumento pedagógico, estando longe a eficácia e robustez de equipamentos industriais. O primeiro contato com o ambiente S4A e a inexperiência em um ambiente de programação visual pode ser a causa das dificuldades no momento da leitura e correção da solução, pois todos os alunos participantes usualmente utilizam fluxogramas e códigos-fonte para programar.

6. Considerações finais

Este trabalho teve como objetivo mostrar a potencialidade do uso de ferramentas da RP aliadas ao Pensamento Computacional para resolução de problemas e aprendizagem de conceitos de programação de computadores. A prática proposta apresenta uma possibilidade de obter um meio menos abstrato e mais motivador do que o modelo tradicional comumente apresentado em cursos de Computação. Com esse novo cenário de ensino é possível engajar o aluno como elemento ativo na construção do conhecimento e no fomento do raciocínio lógico.

Os resultados quantitativos e qualitativos obtidos demonstram de maneira positiva que o método aplicado auxilia a composição da solução desenvolvida pelo aluno. A presença da RP apoia a abstração empírica, permitindo o aluno extrair informações mais concretas do objeto ou das ações sobre o objeto. Em relação às dificuldades existentes durante o processo de produção da solução, direcionar o aluno a um “ciclo de depuração”, intensifica e motiva sua participação nesse processo, desde a transposição de sua ideia para artefatos mais formais (como o fluxograma e a programação em blocos) até a

habilidade de depurar, testando e corrigindo defeitos na solução.

Entretanto, a prática apresentada não esgota o problema relacionado ao ensino de programação de computadores em absoluto, mas contribui na exploração do ferramental disponível na RP e a construção cognitiva orientada ao Pensamento Computacional. Como trabalho futuro é esperado a adição de novos elementos que possam contribuir com a abstração e a solução de problemas, assim como meios eficazes de avaliar o desempenho dos alunos na prática pedagógica. Para tanto, será realizado experimentos com um grupo maior e mais heterogêneo de alunos.

7. Referências bibliográficas

- Benitti, F. B. V. (2012). Exploring the educational potential of robotics in schools: A systematic review. *Computers & Education*, 58(3), 978-988.
- CITILAB (2015). Project Scratch: Scratch for Arduíno (S4A). Disponível em <<http://s4a.cat/>>. Acesso em 15 mai. 2015.
- de Paula, B. H.; Valente, J. A.; e Burn, A. (2014). *O Uso de Jogos Digitais para o Desenvolvimento do Currículo para a Educação Computacional na Inglaterra*. Currículo sem Fronteiras, v. 14, n. 3, p. 46-71.
- Gomes, A., Areias, C., Henriques, J., e Mendes, A. J. (2008). *Aprendizagem de programação de computadores: dificuldades e ferramentas de suporte*. Revista Portuguesa de Pedagogia, v. 42, n. 2.
- McRoberts, M. (2011). *Arduino Básico*. São Paulo: Novatec Editora. 453 p.
- Papert, S. (1985). *Logo: Computadores e Educação*. São Paulo: Brasiliense.
- Rusk, N., Resnick, M., Berg, R., e Pezalla-Granlund, M. (2008). *New pathways into robotics: strategies for broadening participation*. Journal of Science Education and Technology, 17(1), 59–69.
- Valente, J. A. (2005). *Pesquisa, comunicação e aprendizagem com o computador*. O papel, 2005.
- Wainer, J. (2007). *Métodos de pesquisa quantitativa e qualitativa para a Ciência da Computação*. Atualização em informática, p. 221-262.
- Wing, J. M. (2008). *Computational thinking and thinking about computing*. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, v. 366, n.1881, p. 3717-3725.
- Zanetti, H.A.P.; Bonacin, R. (2014). Uma Metodologia Baseada em Semiótica para Elaboração e Análise de Práticas de Ensino de Programação com Robótica Pedagógica. In: Anais do Simpósio Brasileiro de Informática na Educação. p.1233-1242.
- Zilli, S. R. (2004). A robótica educacional no ensino fundamental: Perspectivas e prática. Dissertação de Mestrado em Engenharia, Universidade Federal de Santa Catarina, Centro Tecnológico, Florianópolis-SC.