

Estudo Exploratório sobre o Uso da Robótica Educacional no Ensino de Introdução a Programação

André Rachman Dargains (Universidade Federal do Rio de Janeiro)¹
Fábio Ferrentini Sampaio (Universidade Atlântica e InovLabs - Portugal)²

Resumo

Este trabalho é um estudo exploratório que se apoia em três pilares na busca por um melhor ensino de programação: a robótica educativa, a abordagem construcionista e a Taxonomia de Bloom revisada, com o intuito de identificar métodos e estratégias para o crescimento da qualidade do ensino de programação introdutória. Para isso, duas aplicações foram realizadas com alunos de ensino médio em escolas públicas do Rio de Janeiro, e seus resultados analisados e comparados com uma disciplina de programação oferecida por uma universidade federal local. Espera-se, com esse estudo, a possibilidade de construir novas propostas e identificar caminhos para o fomento da educação em ciência, tecnologia, engenharia e matemática em escolas públicas brasileiras.

Palavras-chave: Ensino de Programação; Robótica Educacional; Taxonomia de Bloom.

Abstract

This work is an exploratory study about teaching-learning introductory programming by secondary students. It departs from three pillars to identify methods and strategies in order to evaluate the students' progress: educational robotics, the constructionist method and the revised Bloom's Taxonomy. To achieve this, two applications were conducted with high school students in Rio de Janeiro's public schools with no programming background. The results were analyzed and compared with a programming course offered by a local federal university. The results of this study are expected to open ways for new proposals and identify ways for the promotion of science, technology, engineering and mathematics education in Brazilian public schools.

Keywords: Teaching programming, educational robotics, Bloom's taxonomy.

¹ Contato: andre.dargains@gmail.com

² Contato: fabio.oxford@gmail.com

1. Introdução

Tornou-se comum referir-se aos jovens como "nativos digitais", devido a sua aparente fluência com as tecnologias digitais (PRENSKY, 2001). E, de fato, muitos deles sentem-se muito à vontade em enviar mensagens de texto, jogar online e navegar na web. Mas será que isso realmente os faz fluentes nas novas tecnologias? Embora os jovens interajam com mídias digitais a todo tempo, poucos podem criar os seus próprios jogos, animações ou simulações. É como se eles pudessem "ler", mas não "escrever".

Segundo Resnick (2007), a fluência digital não exige apenas a capacidade de conversar, navegar e interagir, mas também a capacidade de projetar, criar e inventar novas mídias. Para isso, é necessário aprender algum tipo de linguagem de programação. Nessa mesma linha de pensamento, autores como Wing (2006), afirmam que essa habilidade apoia o desenvolvimento do "pensamento computacional", ajudando a conhecer importantes estratégias de resolução de problemas e de design (como a modularização e design iterativo) que passam para domínios de não programação. Uma vez que a programação envolve a criação de representações externas de seus processos de resolução de problemas, ela também oferece ao programador oportunidades para refletir sobre o seu próprio pensamento (DiSESSA, 2000).

No entanto, aprender a programar pode ser muito difícil para iniciantes, não só para jovens como também pessoas de qualquer faixa etária (KELLEHER; PAUSCH, 2005). Além dos desafios de aprender a formar soluções estruturadas para os problemas e entender como os programas são executados, programadores novatos também precisam aprender uma sintaxe rígida e comandos que podem ter nomes aparentemente arbitrários ou talvez confusos. Enfrentar esses desafios simultaneamente pode ser opressivo e muitas vezes desanimador para esses iniciantes.

Nos últimos anos, a comunidade internacional interessada no ensino de computação e de TICs na educação tem apresentado diferentes trabalhos focando no ensino de programação, ainda nos primeiros anos do ensino regular (GOH; ARIS, 2007; VAL; PASTOR, 2012; FLANNERY et al., 2013; MARTINEZ et al., 2018), ao mesmo tempo em que procuram alternativas pedagógicas e computacionais para ensiná-la (MAJOR et al., 2011; CHIOU, 2004).

Segundo autores como Lepuschitz e outros (2017) e Grover e Pea (2013), ainda hoje, não existe consenso sobre um conjunto universal de estratégias indicadas para a apresentação, estudo e prática desses conteúdos, obrigando desta forma cada instituição a percorrer seus próprios caminhos na busca de soluções. Com o intuito de amenizar esta situação, vários professores e pesquisadores vêm trabalhando diretamente na realização de

experimentos práticos, assim como na construção de ferramentas e metodologias para facilitar o processo de ensino-aprendizagem desta área.

Um problema constante no ensino da programação é a própria avaliação e comparação de desempenho de diferentes práticas de ensino. Como colocado por de Jesus e Raabe:

Entre os muitos desafios enfrentados por um professor de programação introdutória está a avaliação. Como um professor poderia mensurar objetivamente a aprendizagem de seus alunos? Supondo que os alunos obtenham ótimo desempenho, como saber, por exemplo, se o desempenho não foi causado por testes demasiadamente fáceis? O professor poderia comparar o desempenho de sua turma com uma turma de referência. Mas, nesse caso, como saber se a turma de referência é realmente referência? (de JESUS; RAABE, 2009, p. 1).

Segundo Barker e Ansoarje (2007), inovadores no uso da robótica na sala de aula têm relatado muitos elogios. No entanto, há uma clara falta de pesquisa quantitativa sobre a forma como a robótica pode facilitar o aprendizado de disciplinas STEM (do Inglês *Science, Technology, Engineering and Mathematics*) para estudantes. A maioria das pesquisas envolvendo robótica na sala de aula foi conduzida com alunos do ensino médio e de faculdade, com resultados dependentes da percepção de professores ou estudantes, em vez de projetos de pesquisa rigorosos baseados em dados de desempenho do aluno. A utilização de uma taxonomia, como propõe este trabalho, tem como objetivo neutralizar essa falha.

O levantamento da literatura apropriada para o embasamento teórico deste trabalho encontrou diferentes pesquisas com o objetivo de atacar os problemas relacionados ao ensino de programação. As buscas focaram, principalmente, três aspectos do problema: dificuldades de motivação dos alunos; ferramentas para construção do material pedagógico, mensuração e comparação de resultados; estratégias apropriadas para o ensino de programação.

Este trabalho, portanto, tem como objetivo unir a robótica educacional, utilizando estratégias baseadas nas ideias do Construcionismo (PAPERT, 1991) para nortear o planejamento de um curso de programação introdutória, e a Taxonomia de Bloom revisada (ANDERSON et al., 2001) para desenhar o material didático e provas, avaliar os ganhos cognitivos dos participantes e comparar os resultados com outras pesquisas.

Uma resposta satisfatória dos alunos às intervenções feitas pode significar um passo importante na direção da aplicação de uma proposta unificada para o ensino de programação, adaptável ao currículo do nível médio de escolas brasileiras.

Para tal, foi realizado um estudo exploratório dividido em duas etapas: a primeira, chamada Oficina, tratou de uma aplicação piloto de ensino de programação introdutória com o uso da robótica educacional. Seu objetivo foi coletar experiências e averiguar a possibilidade de sucesso de uma segunda execução, muito mais ampla. Nesta primeira etapa já foram utilizados os insumos do trabalho, como a Taxonomia de Bloom revisada, o conteúdo didático aplicado no curso de Computação de uma Universidade e os métodos de coleta comentados mais à frente.

A segunda etapa do trabalho utilizou todos os dados coletados durante a execução da Oficina para construir a segunda aplicação, o Curso. Embora os aspectos cruciais tenham sido mantidos da primeira para a segunda etapa, alguns foram alterados conforme a experiência vivida no estudo. A principal diferença entre as etapas é a duração da aplicação realizada em cada uma, que passou de oito horas em cinco aulas na primeira etapa para trinta e três horas em onze aulas durante o Curso.

Este trabalho está organizado em cinco seções. A primeira delas diz respeito à introdução e aos aspectos gerais da pesquisa. A segunda aborda de forma resumida a revisão de literatura. A terceira detalha o estudo realizado. Na quarta seção a análise dos resultados é apresentada, seguida pela seção 5 que trata das conclusões e trabalhos futuros.

2. Revisão da Literatura

Esta seção apresenta a revisão de literatura realizada sobre os três pontos mencionados na seção 1, nomeadamente o uso da robótica educativa, experiências da abordagem Construcionista, e o uso da Taxonomia de Bloom, todos no âmbito do ensino de programação.

2.1. Robótica Educacional

Muitos autores têm então buscado alternativas para tornar esse aprendizado mais compreensível e eficiente. A robótica, uma dessas alternativas, já foi classificada como facilitadora de aprendizagem de princípios científicos e matemáticos através da experimentação de materiais concretos, incentivadora de classes baseadas em resolução de problemas (ROGERS; PORSTMORE, 2004; NOURBAKHS et al., 2005) e promotora de aprendizagem cooperativa (BEER; CHIEL; DRUSHEL, 1999).

Concomitantemente, outros estudos sobre o uso da robótica na sala de aula têm apontado para um alto grau de interesse e envolvimento dos alunos, fomentando o interesse em carreiras de matemática e ciências (KAYA; HASAN, 2019; EGUCHI, 2016).

Diferentes iniciativas documentam os ganhos diretos do uso da robótica no ensino de programação de computadores. Uma revisão sistemática de literatura, conduzida por Major e seus colegas (2011) apontou mais de 60 estudos unindo o ensino de programação introdutória e robótica, publicados nos últimos 14 anos em congressos e revistas internacionais. Os resultados dos trabalhos, em sua maioria, apontaram para ganhos efetivos no processo de aprendizagem de programação utilizando a robótica versus o ensino tradicional de programação. A revisão também mostra que em 68% dos trabalhos foram utilizados materiais de robótica físicos, contra 20% de trabalhos que utilizam simuladores e 12% que apresentaram ambos. Esse resultado, portanto, aponta para uma preferência para o uso de circuitos, placas e fios, ou seja, material físico para o uso da robótica em aplicações didáticas.

Um argumento para a preferência por materiais físicos no ensino com robôs é que os alunos veem os robôs como brinquedos e diversão (MAUCH, 2001). De fato, um kit de robótica amplamente usado não só pela comunidade científica, mas também em escolas, é desenvolvido pela LEGO³, um conhecido fabricante de brinquedos de blocos de construção para crianças. Alunos que usam este kit podem construir e programar robôs usando os mesmos materiais que eles têm em casa. Isso faz com que tudo o que aprendam com os kits pareça divertido também (BARKER; ANSORJE, 2007).

Autores como Fagin e Merkle (2003) e Barnes (2002) utilizaram robôs para ajudar a ensinar as linguagens de programação Java e ADA, respectivamente, em salas de aula. A principal ênfase em seus cursos foi de ensinar as sintaxes e suas estruturas básicas enfocando os aspectos de engenharia e mecânica de robôs. Outros cursos que também utilizam a robótica têm centrado na construção e programação dos robôs em si (NOURBAKHS et al., 2005; BEER; CHIEL; DRUSHEL, 1999).

Goh e Aris (2007) basearam-se na plataforma Lego MindStorms para ministrar um curso focado na construção de robôs de competição. Separados em grupos, o objetivo dos alunos foi o de se organizarem em funções distintas (programador, engenheiro, designer) e montar o robô mais performático da turma. Diferentes ganhos foram notados, como trabalho em equipe, pesquisa de informações por conta própria e a ruptura da visão da programação e robótica como disciplinas muito complexas.

Chiou (2004) utilizou a robótica para atacar o crescimento do desinteresse por cursos baseados em matemática, ciências e tecnologia. O autor afirma ainda que a robótica

³ <https://www.lego.com/en-us/mindstorms>

educacional precisa ser implementada com grande cuidado para que não seja causado o efeito contrário, ou seja, inadvertidamente afastar os alunos interessados.

Rogers e Portsmore (2004) trabalharam com jovens estudantes utilizando robôs e projetaram um currículo utilizando robôs LEGO e temas de engenharia para alunos da quinta série.

Stager (2009) explora o uso do Construcionismo em uma classe, afirmando que há pelo menos cinco caminhos para o uso adequado da robótica educacional em sala de aula:

- Robótica como disciplina - Robótica é ensinada como uma disciplina própria. Competições de robótica populares, como a First LEGO League, são exemplos dessa abordagem.
- Ensinar conceitos STEM específicos – A robótica pode ser usada para ensinar conceitos científicos físicos, tais como: máquinas simples, força, torque, potência, fricção, vantagens mecânicas; conceitos de ciência da computação como programação, depuração e feedback; conceitos matemáticos como frações, variável, operações aritméticas, etc.
- Unidades temáticas - Estudantes constroem e programam robôs para modelar máquinas e sistemas, como aeroportos, fábricas, parques de diversão ou uma cidade. A expectativa é que as disciplinas escolares tradicionais e seus conceitos subjacentes sejam abordados ou incorporados a estes temas.
- Temas curriculares – A robótica é usada como um meio para resolver problemas específicos ligados a um assunto do currículo formal. Um exemplo poderia ser: "Identificar um problema na África Subsaariana e construir um robô para resolver esse problema." O realismo da solução pode ser subordinado a pensar sobre a natureza do problema.
- Estilo livre – A robótica e programação de computadores são usadas como material de construção como parte do laboratório intelectual do aluno e do seu veículo de auto expressão. O aluno pode usar os materiais para fazer o que quiser. Poderosas ideias emergem nas atividades desse tipo de contexto.

Embora diferentes análises apresentando as variadas conquistas da robótica educacional tenham mostrado que a mesma tem sido eficaz no ensino da programação, foi notado que os diferentes trabalhos utilizam variadas ferramentas para orientação da construção, avaliação e comparação de resultados. O efeito é uma vasta gama de estratégias diferentes empregadas sobre o assunto, incapacitando a comunidade internacional de poder compará-las e eleger os melhores caminhos.

Surge então a necessidade de alinhamento de uma estratégia de construção e avaliação para que os trabalhos possam ser comparados entre si e os projetos mais atraentes, replicados com facilidade.

2.2. Uso de Taxonomias em Cursos de Programação: Busca por uma ferramenta de organização didática e avaliação confiável

Durante o desenvolvimento do trabalho viu-se necessário a adoção de uma métrica para avaliação e comparação dos ganhos cognitivos dos alunos durante os cursos, e uma diretriz a ser adotada para a construção da apresentação dos conceitos e exercícios aos alunos. Para entender melhor como isso se dá, foram levantados estudos propondo cursos ou oficinas de programação que utilizavam algum tipo de taxonomia como norteadora para sua construção.

A partir desses estudos, percebeu-se que a Taxonomia de Bloom (KRATHWOHL et al. 1964) e a Taxonomia de Bloom Revisada (ANDERSON et al., 2001) serviram, para a maioria deles, como base para a elaboração de instrumentos de avaliação e montagem do material didático (de JESUS; RAABE, 2009; WHALLEY et al., 2006; McRACKEN et al., 2001). Assim, neste trabalho, resolveu-se utilizar o domínio cognitivo da Taxonomia de Bloom Revisada para orientar a construção pedagógica da Oficina e do Curso aplicados.

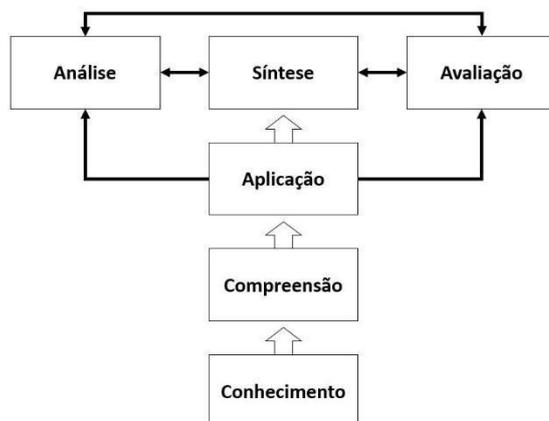
A Taxonomia de Bloom refere-se a uma classificação dos diferentes objetivos que os educadores estabelecem para que seus estudantes os alcancem. Ela divide os objetivos educacionais em três domínios distintos: cognitivo, afetivo e psicomotor. O domínio cognitivo engloba o conhecimento e pensamento, o domínio afetivo o sentimento e as relações interpessoais, e o domínio psicomotor a manipulação física de objetos. O desenvolvimento em cada domínio é medido através de níveis ou categorias organizadas em hierarquias lineares, de forma que para alcançar uma categoria, o aluno deve dominar as categorias anteriores. Portanto, dentro dos domínios, a aprendizagem nos níveis superiores é dependente da obtenção do conhecimento prévio e habilidades em níveis inferiores.

A seguir detalhamos o Domínio Cognitivo da referida Taxonomia, uma vez que neste trabalho, por questões de espaço, vamos focar apenas os resultados relativos a este domínio.

Habilidades no domínio cognitivo tratam do conhecimento, compreensão e pensamento crítico sobre um tema específico. Segundo Scott (2003), a educação tradicional tende a enfatizar as habilidades neste domínio, principalmente as primeiras categorias.

O domínio cognitivo apresenta seis níveis (Figura 1), aumentando em complexidade ao mover-se através das categorias de ordem mais baixas às mais altas. O desenvolvimento de uma categoria não implica no deslocamento imediato para um nível superior. No entanto, segundo Bloom, um estudante precisa dominar as categorias-requisito de uma categoria para poder então alcançá-la.

Figura 1 - Relacionamento entre as categorias do Domínio Cognitivo na Taxonomia de Bloom de acordo com Johnson e Fuller (2006).



Fonte: Johnson e Fuller (2006).

Uma recente reavaliação da Taxonomia de Bloom por Anderson e seus colegas (2001) sugere que os dois ou três primeiros níveis da hierarquia podem ser planos. Os autores também propuseram que a taxonomia deve ser bidimensional, com as (ligeiramente reconfiguradas) categorias originais de *Lembrar*, *Entender*, *Aplicar*, *Analisar*, *Avaliar* e *Criar* formando a dimensão do processo cognitivo e factual, conceitual, procedimental e metacognitivo, como indicado no Quadro 1.

Quadro 1 - Categorias da Taxonomia de Bloom revisada e seus verbos associados

CATEGORIA	PROCESSO COGNITIVO
Lembrar	Reconhecer, recordar
Entender	Interpretar, exemplificar, classificar, sumarizar, inferir, comparar, explicar
Aplicar	Executar, implementar
Analisar	Diferenciar, organizar, atribuir
Avaliar	Checar, criticar
Criar	Generalizar, planejar, produzir

Fonte: Elaborado pelos autores

Segundo Johnson e Fuller (2006), essas taxonomias não definem uma sequência de instruções, mas sim níveis de desempenho que podem ser desejados para qualquer tipo de conteúdo. É esperado que um aluno atuante em um nível cognitivo superior seja capaz de operar nos níveis mais baixos na hierarquia. Isso poderia ser interpretado como o

reconhecimento de um processo de aprendizagem sequencial. No entanto, a Taxonomia não descarta o uso de uma abordagem iterativa para aprender o conteúdo.

Whalley e outros (2006) diz que mesmo para professores de computação experientes, muitas das descrições dos níveis da taxonomia em certas categorias são difíceis de ser interpretados no contexto dos exercícios de programação. Esse problema também é apontado por Pedroza (2011), embora afirme que a Taxonomia revisada solucionou diversas questões da versão original, tornando-a mais indicada para utilização como ferramenta norteadora em trabalhos científicos.

2.3. O Construcionismo

A abordagem construcionista exige uma multiplicidade de perspectivas, de modo que os alunos tenham uma gama completa de opções para construir seu próprio conhecimento.

Na educação científica, essa abordagem pode proporcionar aos alunos oportunidades para redescobrir as teorias atualmente aceitas de uma determinada ciência, bem como teorias rivais que podem, eventualmente, substituir as atuais posições (PAPERT, 1991). Nesse caso, professores podem fornecer orientação ou motivação para ajudar os alunos nas suas descobertas, mas não dirigem excessivamente o processo de aprendizagem.

Segundo Papert:

O construcionismo compartilha com o construtivismo a visão de que aprender é “construir estruturas de conhecimento” através da internalização progressiva de ações. [...] Em seguida, adicione a ideia de que isso acontece de forma especialmente feliz em um contexto onde o aluno está conscientemente empenhado na construção de uma entidade, seja ela um castelo de areia na praia ou uma teoria do universo. (PAPERT, 1991, p.24)

Segundo Ackerman (2001), devido ao seu maior foco na aprendizagem através da prática, a abordagem de Papert ajuda a entender como as ideias se formam e se transformam quando expressas através de diferentes meios de comunicação, quando atualizadas em contextos particulares e quando trabalhadas por mentes individuais.

A mesma autora afirma ainda que, na visão de Papert, mergulhar em situações desconhecidas, ao custo de experimentar uma sensação momentânea de perda, também é uma parte crucial do aprendizado. Somente quando um aluno realmente “viajou através de um mundo”, através da adoção de diferentes perspectivas, ou colocando diferentes “óculos”, pode começar um diálogo entre as experiências locais e inicialmente incompatíveis.

O foco do Construcionismo na prática em sala de aula sugere a utilização de instrumentos concretos no processo de ensino-aprendizagem. O encorajamento de estudantes na construção de circuitos eletrônicos a partir de componentes básicos leva-os a buscar soluções concretas a problemas práticos, gerando significados por meio de

experiências e ações (VALENTE, 1993; PAPERT, 1994). Para chegarem a uma solução final satisfatória, empregam e desenvolvem habilidades de formulação e teste de hipóteses, raciocínio lógico, resolução de problemas por meio de erros e acertos, entre outras (WING, 2011; ZILLI, 2004).

Papert (1980) constatou que a robótica educacional é uma excelente maneira de colocar a teoria Construtivista em prática. Em seus estudos, ele mostra que crianças que aprendem com a robótica são capazes de imaginar-se no lugar do robô e entender como a programação de um computador funciona: os participantes do estudo foram capazes de transferir a sua compreensão do mundo real para a compreensão da lógica e princípios matemáticos. Papert acreditava que o que faz com que muitos conceitos sejam difíceis para as crianças entenderem é a falta de materiais do mundo real que demonstram tais conceitos. Ele aceitava que os robôs programáveis são flexíveis e poderosos o suficiente para serem capazes de demonstrar ideias que antes não apresentavam analogias simples do mundo real.

Ainda outras pesquisas também indicam que a educação experiencial melhora o desenvolvimento social e acadêmico de crianças, incentivando a interação social e a aprendizagem cooperativa (SLAVIN, 2000).

3. O Estudo Realizado: métodos, ferramentas e questões de pesquisa

3.1. Métodos

O objetivo da presente pesquisa foi o de investigar sobre os possíveis ganhos do uso da robótica educativa no ensino de programação introdutória para alunos do ensino médio sem conhecimento prévio de programação. Para tanto foi construída uma proposta de Oficina + Curso a partir de três eixos norteadores: currículo acadêmico de ensino de programação utilizado pela Universidade Federal do Rio de Janeiro (UFRJ); formatação do programa da Oficina e do Curso a partir da Taxonomia de Bloom Revisada, e uma abordagem construcionista na preparação dos exercícios, desafios e tarefas propostas.

No aspecto metodológico este trabalho pode ser qualificado como um estudo exploratório de base qualitativa e quantitativa no qual foram realizadas duas intervenções, em separado, para alunos do ensino médio, sem conhecimento prévio de programação, com uso intenso da robótica educativa como facilitadora do ensino de programação.

As duas intervenções ocorreram ao longo do ano letivo de 2014 com diferentes alunos do 1o. ano do Ensino Médio pertencentes a duas Escolas Públicas - uma escola estadual e outra federal - situadas na cidade do Rio de Janeiro. No total participaram, de forma voluntária, 22 alunos de ambos os sexos.

As intervenções foram realizadas como atividades extracurriculares, oferecidas após os horários de aula.

A primeira intervenção (chamada aqui de Oficina) apresentou um conteúdo condensado (cinco encontros de 90 minutos cada, 10 participantes), servindo de piloto para avaliar o método de curso, material, tempo de aula e capacidade do professor, entre outros fatores.

Após o término da Oficina piloto, algumas mudanças estruturais foram introduzidas na proposta, levando em conta o feedback coletado durante a mesma, de forma a proporcionar para a segunda intervenção uma melhor experiência aos alunos e professor e um resultado final mais rico e assertivo.

A segunda intervenção ocorreu no segundo semestre de 2014 contendo uma carga horária total de 33 horas divididas em 11 encontros semanais onde participaram 12 alunos no total.

Durante as atividades realizadas no laboratório das Escolas os alunos trabalharam em duplas, e a cada uma delas foram alocados um computador e um kit Arduino. A cada novo tópico, havia uma breve explicação apresentada pelo professor e o lançamento de um desafio. Cada dupla deveria trabalhar para resolvê-lo da forma que achasse melhor, com acompanhamento e orientação do professor. Dessa forma, cada aluno exerceu o comando de seu próprio aprendizado, construindo as suas soluções a partir do conhecimento já adquirido e/ou utilizando estratégias como tentativa e erro (PAPERT, 1991).

As principais estruturas computacionais tratadas nas duas intervenções foram: Entrada e saída de dados; Constantes e variáveis; Operadores lógicos; Operadores aritméticos; Operadores relacionais; Atribuição de valores; Condicionais (*if / else*); Funções; Estruturas de repetição; Laços de repetição (*for / while*).

Os dados coletados utilizados durante o processo de análise foram, principalmente, o diário de bordo escrito pelo professor-pesquisador; captura de tela via software rodando em background; as respostas aos exercícios propostos durante os encontros; as provas aplicadas no término da Oficina e do Curso; e o questionário online de opinião dos alunos apresentado ao final da Oficina e do Curso⁴.

⁴ As provas foram construídas contendo questões relevantes ao tema - programação e robótica, ao mesmo tempo em que manteve a estrutura padrão dos exames do curso de Computação I da UFRJ. Mais detalhes sobre as mesmas estão disponíveis Apêndices B e C da dissertação de Mestrado do primeiro autor (DARGAINS, 2015). Por sua vez, os questionários de opinião da Oficina e do Curso estão disponíveis no Apêndice J da referida dissertação.

3.2. Ferramentas

Nas pesquisas apresentadas na literatura foram encontradas diferentes ferramentas e formas de utilizá-las ao aplicar um curso com a presença da robótica educativa. Entre as opções mais populares de hardware, existem o LEGO MindStorms, o Arduino⁵ e o Raspberry Pi⁶.

Este trabalho optou pelo uso de kits com a placa Arduino. Tal escolha se deu principalmente pelo fato de serem de baixo custo, podendo ser facilmente adquiridos futuramente por instituições de ensino públicas que desejam aplicar cursos de robótica educacional ou incorporar a robótica em outros cursos já oferecidos.

A escolha do ambiente de programação recaiu sobre o ambiente visual de programação DuinoBlocks, proposto por Alves em seu trabalho de Mestrado (2013) como facilitador no uso do Arduino (Figura 2).

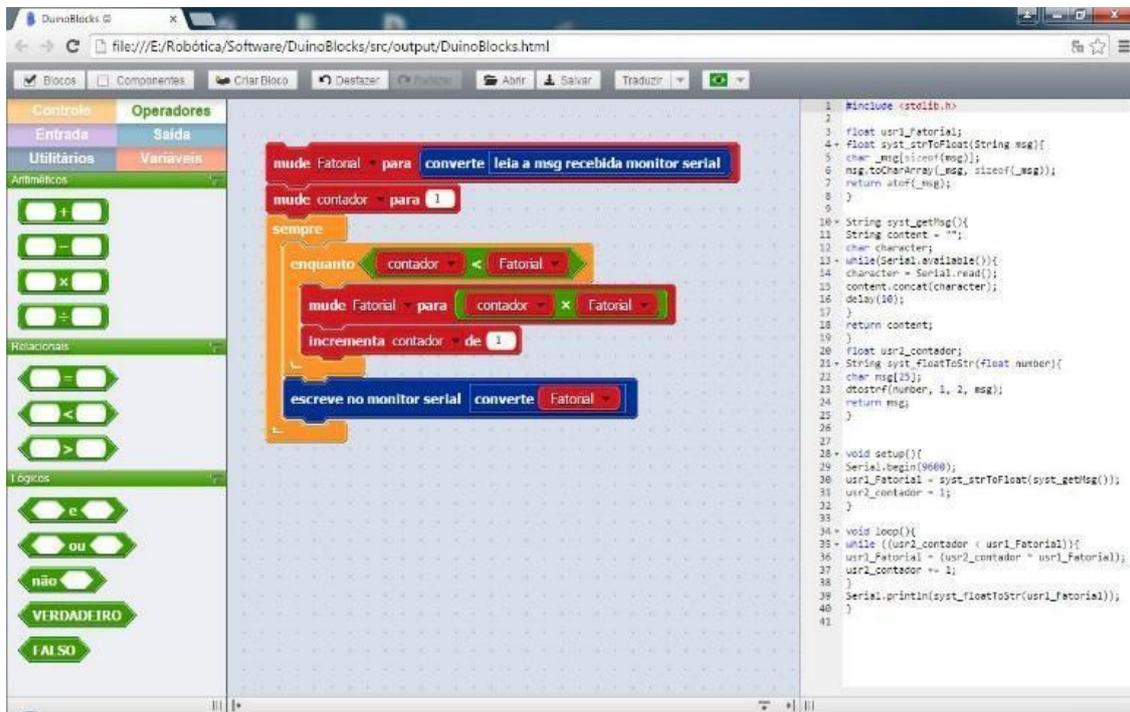
O DuinoBlocks é um ambiente visual de programação desenvolvido para placas Arduino. Sua interface simples, baseada em blocos que podem se conectar, ignora a sintaxe da linguagem de programação, estimulando os alunos a construir seus algoritmos focando apenas em sua semântica. A codificação das categorias de comandos em cores também contribui com a curva de aprendizado dos alunos e sua performance durante a programação. Essas características resultam em um ambiente fácil de ser dominado, consequentemente facilitando também o aprendizado da montagem de algoritmos de programação.

Os blocos se encaixam como um quebra-cabeça, e é possível entender as condições de encaixe observando o formato de cada peça. Por exemplo, operadores relacionais são representados por um hexágono, e suas peças são da cor verde. Essa padronização ajuda os usuários a identificar mais facilmente as peças disponíveis no ambiente e os seus possíveis locais de encaixe (Figura 2).

⁵ Ver <https://www.arduino.cc/>

⁶ Ver <https://www.raspberrypi.org/>

Figura 2 - Tela do DuinoBlocks contendo as suas 3 janelas principais: à esquerda temos as opções de menu de comandos; ao centro a área de programa criado pelo usuário; à direita o código Wiring correspondente ao programa criado pelo usuário (janela opcional)



Fonte: Elaborado pelos autores

O código Arduino correspondente à estrutura de blocos montada pelo usuário é gerado em uma janela no lado direito do ambiente. O código gerado é uma tradução fiel do algoritmo no formato de blocos para o formato de textos. Essa funcionalidade age como uma Pedra de Roseta, permitindo que o usuário possa não só verificar seu código, mas também aprender como construí-lo, utilizando a linguagem nativa do Arduino (Wiring).

3.3. Questões de Pesquisa

A partir do objetivo geral desta pesquisa - apresentado no início desta seção - foram propostas três perguntas de pesquisa tratadas na próxima seção:

- Alunos do ensino médio são capazes de aprender a ementa de um curso universitário de programação, utilizando robótica?
- Esses alunos são capazes de alcançar os níveis superiores da Taxonomia de Bloom revisada?
- A disponibilidade de materiais físicos (robótica) influencia no desempenho dos alunos?

4. Análise dos Resultados

4.1. O Desenvolvimento das aulas

Os alunos, de uma forma geral, não estavam acostumados a programar ou trabalhar com circuitos elétricos. Assim, a atenção do professor foi muito necessária no sentido de orientá-los durante as tarefas, principalmente no início do curso. Eles, no entanto, não sentiram grandes dificuldades no aprendizado da programação, uma vez que classificaram entre "muito fácil" e "médio" o grau de dificuldade em aprender programação⁷, conforme respostas à questão do formulário online: "Você achou que aprender programação foi fácil ou difícil?".

A mesma estrutura de aula se manteve em ambas as aplicações: um novo tópico de programação era apresentado para a turma pelo professor, seu mecanismo explicado e desafios e exercícios lançados. Após a apresentação de soluções para o problema inicial proposto, as duplas eram muitas vezes estimuladas a pensar sobre possíveis modificações nos programas a fim de implementar novas funcionalidades (aumento de complexidade do problema).

As atividades e exercícios propostos eram sempre apresentados como provocações, explorando a predisposição natural dos alunos em aprender através de desafios. As duplas sempre reagiram bem a esta estratégia, tanto na Oficina quanto no Curso, algumas vezes chegando a estabelecer competições entre elas na busca de códigos e circuitos mais elegantes/eficientes⁸. Entretanto, era comum observar alunos que já haviam terminado o exercício ajudarem os colegas que ainda estavam tentando resolvê-los.

4.2. Construção de programas com o DuinoBlocks

Durante as aplicações foi percebido um avanço rápido no entendimento e na construção de programas, uma vez que os estudantes não tiveram problemas no domínio do ambiente DuinoBlocks. Ao final do curso os alunos eram capazes de montar seus programas sem dificuldade aparente, parando apenas em dúvidas de lógica ou de performance do algoritmo em questão.

Ao responder à pergunta do questionário online "Você achou que aprender a programar no DuinoBlocks foi fácil ou Difícil?", obteve-se 100% das respostas classificadas entre as categorias muito fácil e médio.

⁷ Nesta pergunta do formulário online foi utilizada uma escala de 05 valores: muito fácil; fácil; médio; difícil; muito difícil.

⁸ Alguns exemplos de exercícios proposto podem ser consultados na dissertação de Mestrado do 1o. Autor em (DARGAINS, 2015)

Embora o ambiente visual baseado em blocos DuinoBlocks tenha sido utilizado, os alunos tiveram, a todo tempo, contato com o algoritmo textual gerado pelos blocos, a partir da funcionalidade “traduzir”. Embora não tenha sido requisitado, muitos deles aventuraram-se em tentar compreender o código textual, desistindo logo em seguida. Os alunos, em seguida, comentaram sobre a facilidade e sua preferência pelo ambiente visual contra o ambiente de programação nativo do Arduino, confirmada através das respostas coletadas no questionário online de opinião.

A reação dos alunos quando percebiam um erro no seu código também foi notada. Ao executar os programas, os alunos não sabiam ao certo, inicialmente, onde estava o erro - no algoritmo ou na montagem do circuito. Era preciso então verificar ambos para encontrar o equívoco, o que demandava certo tempo e, algumas vezes, a atenção do professor ou colegas. Novamente, foi necessário o domínio das categorias superiores da Taxonomia de Bloom revisada para chegar à solução correta. No entanto, isso não inibiu de forma alguma o interesse – inclusive, muitas vezes erros eram considerados pelos alunos desafios a serem conquistados – e a vontade deles de montar um novo circuito a cada novo programa, independentemente de sua complexidade.

O fornecimento imediato de feedback presente na robótica foi de grande valia na busca por erros de algoritmo e montagem de circuitos. Os alunos recebiam, rapidamente, mensagens de erro ou resultados incompatíveis com o esperado, possibilitando-os a buscar correções e melhorias de performance. Esse aspecto foi observado por diversos alunos ao longo das aplicações. A correta interpretação do feedback, no entanto, só é possível após o aluno alcançar a categoria Aplicar da Taxonomia de Bloom revisada, uma vez que é necessário primeiro saber quais comandos foram utilizados, entender como cada um deles funciona individualmente e em integração com os outros comandos, e aplicá-los de forma correta para formar um algoritmo funcional.

4.3. Envolvimento dos alunos nas atividades

Os alunos não apresentaram nenhum bloqueio visível com as questões computacionais (software e hardware), e não demoraram a se acostumar com a terminologia eletrônica e elétrica, bem como com a montagem de circuitos eletrônicos na protoboard.

Durante as aplicações, em diferentes momentos, foi possível perceber e anotar demonstrações de interesse e envolvimento por parte dos alunos, como por exemplo:

- Claras manifestações de alegria quando seu programa/circuito funcionava. Chamavam o professor e outros alunos para mostrar o que produziram e aguardavam ansiosos, esperando o próximo desafio;

- Ao montar um programa ou um circuito particularmente difícil, os alunos frequentemente atribuíam valor sentimental ao projeto, hesitando em desmontá-lo para realizar a próxima tarefa. Em alguns casos os projetos chegaram a receber nomes e foram muitas vezes fotografados para compartilhamento em redes sociais.

Foi possível constatar, ao longo da Oficina e do Curso, o desenvolvimento do entendimento de programação e robótica dos alunos através de interações entre os membros das duplas. Em diferentes momentos foram observadas correções dos colegas como:

"O LED está invertido, olha" (Aluno D, desafio, aula 4 da Oficina);
"Se você usar um loop aqui (apontando para o algoritmo do colega) dá para resolver o problema" (Aluno E, exercício em sala, aula 3 do Curso);
"Não esquece do resistor, ou o LED pode queimar" (Aluno A, exercício em sala, aula 5 do Curso).
"Você precisa de uma condição verdadeira para continuar no while" (Aluno I, exercício em sala, aula 7 do Curso).

Tais observações indicam um crescimento claro do entendimento do conteúdo programático por parte dos alunos, uma vez que tais demonstrações requerem o domínio da categoria Analisar da Taxonomia de Bloom revisada ao olhar, de maneira crítica, os projetos dos colegas, e sugerir novas formas de abordar o problema em questão. Além disso, tais comentários mostram que os alunos não só aprenderam sobre questões de programação e robótica como estavam também ensinando seus colegas a respeito.

Durante um desses comentários, foi possível observar o aluno explicar uma questão relacionada a um algoritmo ao colega enquanto demonstrava cada passo apontando na tela do computador:

"Para fazer um loop enquanto funcionar você precisa de uma operação relacional, olha. É a mesma coisa que você faz quando compara dois números, tipo se um é maior ou menor que o outro. Enquanto essa operação for verdadeira, o loop continua, e assim que for falsa, o programa sai dela. Você precisa criar uma variável, tipo um contador, e pensar quantas vezes o loop precisa rodar, aí você dá esse valor ao contador. Só não esquece de mudar o valor do contador toda vez que o loop rodar" (Aluno G, exercício em sala, aula 8 do Curso).

O passo a passo indicado pelo aluno demonstra seu entendimento do assunto, através do domínio das categorias Lembrar, Entender e Aplicar da Taxonomia em relação à programação.

Os alunos frequentemente perguntavam acerca dos componentes eletrônicos utilizados na montagem dos circuitos que, embora apresentados apenas superficialmente no início do curso, (por não ser o objetivo maior da Oficina), eram empregados em todos os

projetos. Eles buscaram mais informações também na internet e uns com os outros. Esse comportamento evidencia a curiosidade e uma predisposição deles no aprendizado baseado em projetos.

Um exemplo desse interesse foi notado durante o Curso quando um aluno comentou que ele, em casa, assistiu diversos vídeos tutoriais sobre Arduino e montagem de circuitos na protoboard em um popular portal de vídeos online. Essa atividade foi de iniciativa própria do aluno, conforme mensagem de celular enviada pelo aluno ao professor do Curso.

4.4. As provas

As provas foram aplicadas no último dia de aula da Oficina e do Curso, com o objetivo de avaliar os ganhos cognitivos dos alunos durante as duas intervenções. Para a sua construção foram utilizados como modelo testes aplicados nos últimos três anos no curso de Computação Unificado da UFRJ, analisados conforme a Taxonomia de Bloom revisada.

A análise das provas modelo mostrou que apenas três categorias da Taxonomia eram utilizadas: Entender, Aplicar e Analisar, as mais básicas. O uso dessas três categorias é muito comum em provas de computação, onde a maioria das questões solicita a construção de um programa, a correção de erros de um código ou a apresentação de quais são as saídas esperadas de um programa (SCOTT, 2003). Johnson e Fuller (2006) colocam que, de forma geral, cursos de programação tendem a considerar a categoria Aplicar como a mais importante a ser desenvolvida e inclinam-se a ignorar as categorias superiores a esta.

Tal cenário põe em evidência uma aparente deficiência no ensino de computação, pois o aprendizado se limita a apenas algumas categorias da Taxonomia. Segundo Scott (2003), cursos que englobam todas as seis categorias podem prover melhores resultados de aprendizado e melhores formas de se medir esses resultados.

Assim, na preparação das provas aplicadas neste estudo houve o cuidado de apresentar cada tópico abordando todas as categorias da Taxonomia - de forma a poder-se avaliar o aprendizado de forma mais completa - e, ao mesmo tempo, procurando-se preservar as competências e habilidades necessárias à solução das questões existentes nas provas modelo.

O resultado médio dos alunos que participaram da Oficina foi de 90,00 % de acertos, 2,00 % de respostas parcialmente corretas e 8% de respostas inválidas. Ficou evidente, após a correção, que a maior dificuldade dos alunos foi no tópico laços de repetição. Apenas sete alunos de dez acertaram essa questão. Todos compreenderam com facilidade os demais tópicos, e no geral os resultados foram muito favoráveis.

Após a aplicação na Oficina e coleta de feedback, a prova aplicada durante o Curso sofreu algumas modificações para deixar seus enunciados mais claros. Além disso, foram

adicionadas mais três questões, uma vez que o Curso, mais extenso que a Oficina, permitiu maior aprofundamento no conteúdo.

O resultado médio da prova aplicada no Curso foi de: 90,63% de acertos, 7,81% de respostas parcialmente corretas e 1,56% de respostas inválidas. Ficou evidente, após a correção, que a maior dificuldade dos alunos foi no tópico laços de repetição. Em uma das questões que implicava no uso de laços, todos os alunos apresentaram algoritmos que não continham laços de repetição, embora todas as soluções resultassem em um programa correto. Todos compreenderam com facilidade os demais tópicos, e os resultados foram ainda mais favoráveis que os obtidos na Oficina.

4.5. Outros aspectos relativos às aplicações da Oficina e do Curso

Problemas de conexão com a internet e questões de configuração de computadores configuraram contratempos recorrentes no início do Curso. Esses obstáculos evidenciam a dificuldade que professores enfrentam nas escolas ao pensar e aplicar atividades que estejam fora do escopo tradicional da educação.

A inflexibilidade inerente da instituição de ensino, portanto, atua como uma barreira que impede a realização de propostas inovadoras e, por fim, a própria evolução do sistema de ensino. Pedroza (2011) discute essa questão em profundidade, apontando causas, consequências e possíveis soluções.

Tanto na Oficina quanto no Curso, a atenção demandada pelos alunos foi imensa. A necessidade de incorporar ao menos um monitor à equipe foi detectada durante a Oficina, porém devido a questões de tempo, orçamento e logística, tal adição não foi possível em ambas as aplicações. Mesmo com os alunos divididos em duplas, o professor viu-se constantemente sendo requisitado pelos grupos. A intensidade da demanda, no entanto, dependeu da fase da aplicação. No início, quando os alunos ainda não estavam familiarizados com o software ou com o hardware, as solicitações foram muito mais frequentes. À medida que as aulas foram progredindo, os alunos se tornaram mais proficientes com os materiais e a demanda por atenção caiu. Outro fator que contribuiu para essa queda foi o comportamento cooperativo entre alunos da mesma dupla e entre duplas distintas, discutido anteriormente nesta seção.

4.6. Considerações parciais

Os resultados gerais da análise apontam para uma melhoria no entendimento de conceitos de tecnologia e engenharia através de suas descrições e explicações acerca dos projetos. Isso pode ser afirmado através de observações feitas durante as aulas, dos resultados das provas e das respostas dos questionários. Todos os alunos atingiram níveis satisfatórios de compreensão em relação aos tópicos abordados de programação

introdutória, alcançando inclusive as categorias superiores da Taxonomia de Bloom revisada.

A análise dos dados coletados nos permitem afirmar, com algum grau de certeza, que problemas no aprendizado de computação não estão necessariamente ligados à idade dos alunos. Nas duas aplicações deste trabalho, alunos do ensino médio exibiram plena capacidade de utilizar conhecimentos aprendidos de programação e eletrônica para criar projetos complexos unindo computação e robótica. Conseqüentemente, é possível afirmar que tais alunos foram capazes de dominar as categorias da Taxonomia de Bloom revisada com sucesso.

Assim, resumidamente pode-se intuir que:

- Os alunos alcançaram as categorias superiores da Taxonomia de Bloom revisada com sucesso;
- Os alunos preferiram aprender programação e realizar exercícios com auxílio da robótica educacional, mesmo que isso implicasse em maior trabalho por parte dos mesmos;
- Os alunos demonstraram preferência a utilizar um ambiente visual de programação ao ambiente textual nativo do Arduino;
- A comparação dos índices de aprovação e evasão entre as turmas participantes deste trabalho e aquelas de Computação I da UFRJ, sugere uma ligação entre as taxas de aprovação, evasão e reprovação e a quantidade de alunos da turma: quanto mais alunos inscritos, menor a taxa de aprovação e maiores as taxas de reprovação e evasão;
- A organização dos alunos em grupos favorece a cooperação não só entre os membros da dupla, mas entre duplas distintas, conseqüentemente aprimorando o aprendizado em geral.

5. Conclusões e Trabalhos Futuros

5.1. O trabalho realizado

Este trabalho propõe uma avaliação dos possíveis ganhos do uso da robótica educativa no ensino de programação introdutória para alunos do ensino médio, sem conhecimento prévio da disciplina.

A partir de uma revisão de literatura envolvendo pesquisas sobre o uso da robótica no ensino de programação, decidiu-se pelo uso da Taxonomia de Bloom revisada como instrumento para orientar a construção das aulas e uma avaliação padronizada, tornando possível a comparação de resultados, alteração e reutilização de projetos em outros cenários.

Para realização do estudo, foram implementadas duas intervenções em escolas públicas

cariocas: a primeira, de curta duração, chamada de Oficina, e a segunda, mais longa, chamada de Curso.

Os participantes eram alunos do 1o. ano do Ensino Médio, com idade entre 15 e 17 anos, sem experiência prévia em programação de computadores (22 alunos no total).

O trabalho utilizou como instrumento de pesquisa o Estudo de Caso, uma vez que este é mais indicado para estudos exploratórios (COHEN; MANION; MORRISON, 2013). A análise qualitativa dos dados foi realizada a partir de passagens de diálogos entre alunos, dos resultados de exercícios e provas e das opiniões dadas pelos participantes. A coleta desses dados foi feita por meio de questionários de opinião, observações do comportamento individual dos alunos na forma de um diário de aula, resultados dos projetos apresentados e testes de conhecimentos.

A partir da análise dos dados, é possível afirmar que o trabalho proposto nesta pesquisa colaborou de forma significativa no entendimento de conceitos sobre lógica de programação, construção de algoritmos, estruturas de programação e montagem de circuitos eletrônicos. Isto pode ser ilustrado no comentário de um dos alunos:

“Eu não sabia bem o que esperar quando entrei no curso, mas gostei muito de poder dar vida aos circuitos que eu montava, do jeito que eu queria. No início foi difícil, mas depois de um tempo passei a perceber os algoritmos por trás das coisas do dia a dia, como em um sinal de trânsito na rua.”
(Aluno B, Questionário de Opinião, Oficina).

Reações positivas, como a apresentada acima, motivaram a direção de uma das escolas participantes a buscar uma forma de dar continuidade à oferta de um curso semelhante. As respostas às questões de pesquisa apresentadas na seção 3 são tratadas a seguir, de forma resumida, conforme o que já foi detalhado na Seção 4.

• **Alunos do ensino médio são capazes de aprender corretamente a ementa de um curso universitário de programação, utilizando robótica?**

A principal característica da Taxonomia de Bloom revisada reside em classificar os objetivos de aprendizagem em ordem crescente de complexidade (níveis), com base nas operações mentais que eles requerem, independente do domínio do conhecimento. Através deles, é possível que os professores definam claramente o nível de conhecimento que eles esperam que os alunos atinjam em um determinado conteúdo, podendo, assim, preparar aulas e avaliações compatíveis com este nível.

A Taxonomia, portanto, indicou de forma sistemática os avanços cognitivos dos alunos ao longo das aulas e exercícios propostos nas intervenções. Além dos comentários, apresentados de forma sintética na seção 4, os resultados das provas mostraram que os

alunos foram capazes de alcançar todos os níveis da Taxonomia com sucesso, em relação a todos os tópicos da ementa da disciplina de Computação I oferecida pela UFRJ.

• **Os alunos são capazes de alcançar os níveis superiores da Taxonomia de Bloom revisada?**

A resposta a esta pergunta é consequência direta do cumprimento da anterior: afirmar que um indivíduo é capaz de aprender, corretamente, um determinado conteúdo, significa dizer que ele conseguiu, com sucesso, alcançar todas as categorias da Taxonomia, em relação a esse conteúdo.

A partir dos resultados analisados podemos afirmar que os alunos foram capazes de dominar todos os tópicos apresentados durante o Curso, ao alcançarem com sucesso todas as categorias da Taxonomia de Bloom revisada.

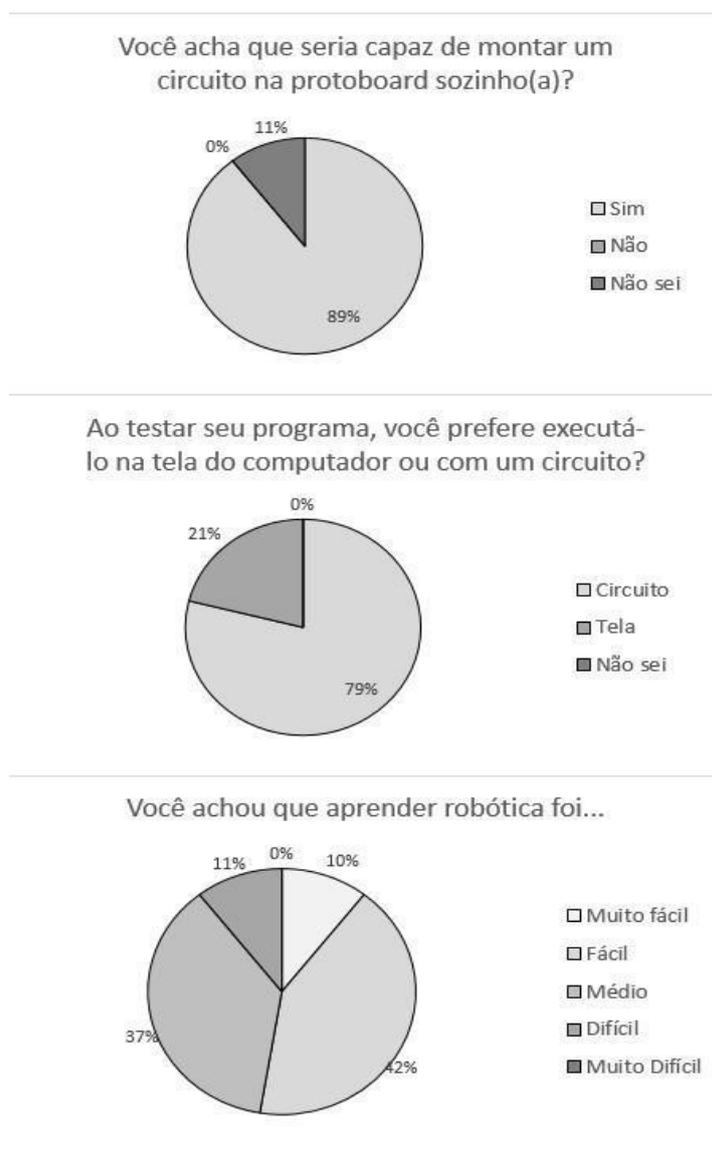
• **A disponibilidade de materiais físicos (robótica) influencia no desempenho dos alunos?**

Diferentes estudos, abordados na seção 2.3 da revisão de literatura, indicam que a utilização da robótica educativa é interessante para o ensino de programação. Eles apontam que a robótica é uma facilitadora de aprendizagem de princípios científicos e matemáticos através da experimentação de materiais concretos (ROGERS; PORTSMORE, 2004), incentivadora de classes baseadas em resolução de problemas (ROGERS; PORTSMORE, 2004; NOURBAKHS et al., 2005; ROBINSON, 2005) e promotora de aprendizagem cooperativa (NOURBAKHS et al., 2005; BEER; CHIEL; DRUSHEL, 1999).

Durante as intervenções foi possível verificar que os alunos preferiram aprender programação e realizar exercícios com auxílio da robótica educacional, mesmo que isso implicasse em maior trabalho por parte deles ao precisar planejar e construir os circuitos necessários para executar os programas. Além disso, os alunos demonstraram preferência a utilizar um ambiente visual de programação ao ambiente textual nativo do Arduino. Essas informações podem ser apoiadas pelos gráficos apresentados na Figura 3, construídos a partir das respostas do questionário de opinião.

O fato de os alunos terem sido capazes de aprender de forma satisfatória a ementa da disciplina de Computação I da UFRJ sugere que a disponibilidade de material físico, no formato de kits de robótica, foi benéfica para o seu desenvolvimento cognitivo.

Figura 3 - Respostas dos participantes aos questionários de opinião



Fonte: Autores

O fato de os alunos terem sido capazes de aprender de forma satisfatória a ementa da disciplina de Computação I da UFRJ sugere que a disponibilidade de material físico, no formato de kits de robótica, foi benéfica para o seu desenvolvimento cognitivo.

5.2. Trabalhos Futuros

O principal objetivo deste trabalho foi explorar o uso da robótica educacional no ensino de programação introdutória. Os resultados sugerem que os alunos do primeiro ano do ensino médio que participaram das intervenções são capazes de elaborar e entender algoritmos computacionais a partir da ferramenta de programação visual DuinoBlocks.

Entretanto, os resultados indicam também que outras questões de pesquisa podem ser estabelecidas, encorajando o desenvolvimento de novos estudos para que possam ser mais bem explorados.

Ambas as intervenções trataram de cursos extracurriculares, onde a inscrição foi voluntária. Os participantes estavam, por definição, altamente motivados por se tratar de um interesse pessoal. Além disso, as turmas de cada intervenção foram compostas de aproximadamente um terço da quantidade de alunos em uma sala de aula tradicional de ensino médio. Um aspecto que poderia ser aprofundado é a avaliação da participação de alunos pouco interessados em programação ou robótica, através da aplicação de uma variante do Curso em uma turma completa de ensino médio, durante todo um ano letivo.

Outro aspecto interessante seria o acompanhamento e avaliação do desempenho de um aluno nas matérias associadas à educação STEM, durante sua participação no Curso de programação com robótica, uma vez que estudos indicam que disciplinas como Matemática, Física e Química são favorecidas através do aprendizado de programação (WALCK, 2014).

A última questão a ser sugerida é a avaliação dos resultados do Curso proposto, considerando todos os três domínios da Taxonomia de Bloom revisada para uma melhor avaliação do desenvolvimento intelectual dos alunos. Segundo Reeves (2006), todos os três domínios da Taxonomia são importantes para a avaliação completa do crescimento de um indivíduo. No entanto, devido à complexidade de utilizar todos os domínios e o foco no aprendizado cognitivo dos alunos de programação, a escolha de não contemplar os domínios afetivo e psicomotor foi tomada neste trabalho. Uma análise mais completa dos resultados, à luz das três categorias da Taxonomia, traria um melhor entendimento sobre o ensino de programação introdutória com robótica educacional nas escolas. Desta forma, fazem-se necessárias investigações mais aprofundadas no sentido de desenvolver propostas pedagógicas para introdução de tal disciplina em sala de aula.

6. Referências

ACKERMAN, E. (2001). Piaget's constructivism, Papert's constructionism: What's the difference?. **Future of learning group publication**, v. 5, n. 3, p. 438.

ALVES, R. M. **DuinoBlocks**: Desenho e implementação de um ambiente de programação visual para robótica educacional. 2013. Dissertação (Mestrado em Informática), Programa de Pós-Graduação em Informática, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2013.

ANDERSON, L. W.; KRATHWOHL, D. R.; AIRASIAN, P. W.; CRUIKSHANK, K. A.; MAYER, R. E.; PINTRICH, P.; RATHS, J.; WITTRICK, M. C. **A taxonomy for learning, teaching, and assessing**: A revision of Bloom's Taxonomy of Educational Objectives. New York: Longman, 2001.

BARKER, B. S.; ANSORGE J. Robotics as a means to increase achievement scores in an informal learning environment. **Journal of Research on Technology in Education**, v. 39, n. 3, p. 229–243, 2007.

BARNES, D. J. Teaching Introductory Java through LEGO MINDSTORMS models. In: TECHNICAL SYMPOSIUM ON COMPUTER SCIENCE EDUCATION. - 33rd.SIGCSE. 2002. Cincinnati, EUA. **Proceedings...** Nova York, ACM, 2002, p. 147–151.

BEER, R.; CHIEL, H.; DRUSHEL, R. Autonomous Robotics to Teach Science and Engineering. **Communications of The ACM**, v. 42, no. 6, p. 85-92. 1999

CHIOU, A. Teaching Technology Using Educational Robotics. In: AUSTRALIAN CONFERENCE ON SCIENCE AND MATHEMATICS EDUCATION. ACSME, 2004, Sidney, Australia. **Proceedings...** University of Sydney, 2004. p. 9-14.

COHEN, L; MANION, L; MORRISON, K. **Research Methods in Education**. New York: Taylor & Francis, 2013.

DARGAINS, A. R. **Estudo exploratório sobre o uso da robótica educacional no ensino de programação introdutória**. 2015. Dissertação (Mestrado em Informática), Programa de Pós-Graduação em Informática, Universidade Federal do Rio de Janeiro, Rio de Janeiro. 2015.

de JESUS, E.; RAABE, A. Interpretações da Taxonomia de Bloom no Contexto da Programação Introdutória. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 20^o SBIE, 2009. Florianópolis. Universidade Federal de Santa Catarina. **Anais...** UFSC, Florianópolis, 2009. p. 1–10.

DiSESSA, A. **Changing Minds: Computers, Learning, and Literacy**. Cambridge: MIT Press. 2000.

EGUCHI, A. RoboCupJunior for promoting STEM education, 21st century skills, and technological advancement through robotics competition". **Robotics and Autonomous System**, v. 75, p. 692-699, 2016.

FAGIN, B.; MERKLE, L. Measuring the Effectiveness of Robots in Teaching Computer Science." **ACM SIGCSE Bulletin**, v. 35, n. 1, 2003.

FLANNERY, L.; SILVERMAN, B.; KAZAKOFF, E.; BERS, M.; BONTÁ P.; RESNICK M. Designing Scratchjr. Support for early childhood learning through computer programming. In: INTERNATIONAL CONFERENCE ON INTERACTION DESIGN AND CHILDREN. Nova York, EUA. **Proceedings...** Nova York, ACM, 2013, pp. 1–10.

GOH, H.; ARIS, B. Using robotics in education: lessons learned and learning experiences. Smart Teaching & Learning: Re-engineering ID, Utilization and Innovation of Technology. In: 1st. INTERNATIONAL MALAYSIAN EDUCATIONAL TECHNOLOGY CONVENTION, 2007, Johor Bharu, Malasia. **Proceedings...** Faculty of Education University Teknologi Malaysia, 2007, p. 1156–1163.

GROVER, S.; PEA, R. Computational Thinking in K12. A Review of the State of the Field. **Educational Researcher**, v. 42, n.1, p.38-43, 2013.

JOHNSON, C.; FULLER, U. Is Bloom's taxonomy appropriate for computer science?. *In: BALTIC SEA CONFERENCE ON COMPUTING EDUCATION RESEARCH*, 2006, Uppsala, Suécia. **Proceedings...** Uppsala University, Suécia, 2017. p. 120-124.

KAYA, E.; HASAN, D. Thinking Intervention on Preservice Elementary Science Teachers' Computational Thinking Teaching Efficacy Beliefs, Interest and Confidence. *In: CONFERENCE OF THE SOCIETY FOR INFORMATION TECHNOLOGY AND TEACHER EDUCATION, SITE 2019*, Las Vegas, EUA. **Proceedings...** Waynesville, Carolina do Norte, EUA, AACE, 2019, p. 1600–1605.

KELLEHER, C.; PAUSCH, R. Lowering the Barriers to Programming: A survey of programming environments and languages for novice programmers. **ACM Computing Surveys**, v. 37, n. 2, p 83–137, 2005.

KRATHWOHL, D.; BLOOM, B.; MASIA, B. **Taxonomy of educational objectives: the classification of educational goals**. Handbook Volume 2: Affective domain. 1964

LEPUSCHITZ, W.; MERDAN, M.; KOPPENSTEINER, M.; BALOGH, G.; OBDRZALEK, D. (Eds.). **Robotics in Education - Latest Results and Developments**. 2017.

MAJOR, L.; KYRIACOU, T.; BRERENTON, O. P. Systematic Literature Review: Teaching Novices Programming Using Robots. *In: EVALUATION AND ASSESSMENT IN SOFTWARE ENGINEERING ANNUAL CONFERENCE*, 2011, Durham (UK). **Proceedings...** Londres, Institution of Engineering and Technology, 2011, p. 502-513.

MARTINEZ, L. G.; MARRUFO, S.; LICEA, G.; REYNEZ-JUAREZ, J.; AGUILAR, L. Using a Mobile Platform for Teaching and Learning Object Oriented Programming. **IEEE Latin America Transactions**, v. 16, n.6, 2018.

MAUCH, E. Using technological innovation to improve the problem solving skills of middle school students. **The Clearing House**. v.75, n. 4, p. 211–213, 2001.

McCRACKEN, M.; ALMSTRUM, M. V.; DIAZ, D.; GUZDIAL, M.; HAGAN, D.; KOLIKANT, Y. B. D.; LAXER, C.; THOMAS, L.; UTTING, I.; WILUSZ, T. A multi-national, multi-institutional study of assessment of programming skills of first-year CS students. **ACM SIGCSE Bulletin**, v. 33, n.4, p. 125-140, 2001.

NOURBAKHSH, I.; CROWLEY, K.; BHAVE, A.; HAMNER, E.; HSIU, T.; PEREZ-BERGGUIST, A.; RICHARDS, S.; WILKINSON, K. The Robotic Autonomy Mobile Robotics Course: Robot Design, Curriculum Design and Educational Assessment. **Autonomous Robots**, v. 18, n.1, p. 103-27, 2005.

PAPERT, S. **Mindstorms: Children, Computers and Powerful Ideas**. 1980.

PAPERT, S. Situating Constructionism. *In: PAPERT, S.; HAREL, I. (Eds.), Constructionism*. p. 1-11. 1991.

PAPERT, S. **A máquina das crianças: repensando a escola na era da informática**. Porto Alegre: Artmed. 1994.

PEDROZA, S. A evolução da educação: necessidade de uma nova gestão escolar. *In: CONGRESSO IBERO-AMERICANO DE POLÍTICA E ADMINISTRAÇÃO DA EDUCAÇÃO*. Porto. 2011.

PRENSKY, M. Digital Natives, Digital Immigrants Part 1. **On the Horizon**, v. 9, n. 5, p. 1 – 6, 2001.

REEVES, T. How do you know they are learning?: the importance of alignment in higher education. *International Journal Learning Technology*, v. 2, n. 4, 2006.

RESNICK, M. Sowing the Seeds for a More Creative Society. **Learning & Leading with Technology**, v. 35, n.4, p. 18-22. 2007.

ROBINSON, M. Robotics-driven activities: Can they improve middle school science learning? **Bulletin of Science, Technology & Society**, v. 25, n. 1, p. 73-84, 2005.

ROGERS, C.; PORTSMORE, M. Bringing Engineering to Elementary School. **Journal of STEM Education**, v. 5, n.3, 2004.

SCOTT, T. Bloom's Taxonomy Applied to Testing in Computer Science Classes. **The Journal of Computing in Small Colleges**, v. 19, n. 1, p. 267-274. 2003.

SLAVIN, R. **Educational Psychology: Theory and practice** (6th Ed.). Boston: Allyn and Bacon, 2000.

STAGER, G. A Constructionist Approach to Teaching with Robotics. **Proceedings 9th IFIP World Conference on Computers in Education**. 2009.

VAL, S.; PASTOR, J. Robotics in Education. *In: Advanced Research in Scientific Areas*, 2nd. ARSA, 2012. Slovakia. **Proceedings...** University of Patras, 2012. p. 1917-1921.

VALENTE, J. A. Por que o computador na educação. *In: VALENTE, J. A. Computadores e Conhecimento: repensando a educação*. Campinas: NIED, UNICAMP. 1993.

WALCK, S. Learn Physics by Programming in Haskell. *In: INTERNATIONAL WORKSHOP ON TRENDS IN FUNCTIONAL PROGRAMMING IN EDUCATION*, TFPPIE 2014. Holnand. **Proceedings...** Open Publishing Association, Holland, 2014. p. 67-77.

WHALLEY, J., LISTER, R.; THOMPSON, E.; CLEAR, T.; ROBBINS, P.; KUMAR, P.; PRASAD, C. An Australasian Study of Reading and Comprehension Skills in Novice Programmers, using the Bloom and SOLO Taxonomies. *In: AUSTRALASIAN CONFERENCE ON COMPUTING EDUCATION*, 8TH, 2006, Hobart, Australia. **Proceedings...** ACM, Nova York, EUA, 2006. p. 243-252.

WING, J. M. Computational Thinking. **Communications of the ACM**, v. 49, n. 3, p. 33-35. 2006.

WING, J. M. Research Notebook: Computational Thinking-What and Why?. **The Link Magazine**, p. 20-23, Spring 2011.

ZILLI, S. R. **A Robótica Educacional no Ensino Fundamental: Perspectivas e Práticas**. 2004. Dissertação (Mestrado em Engenharia), Universidade Federal de Santa Catarina/SC, Florianópolis, 2004.