

Um Estudo Exploratório sobre o Uso da Robótica Educacional como Ferramenta de Apoio ao Ensino-Aprendizagem de Lógica de Programação para Alunos da Rede Pública do Ensino Médio

José Augusto M. Vidal
CESAR/Centro de Estudos e
Sistemas Avançados do Recife –
Recife – PE - Brasil
jamv@cesar.org.br

Fábio Ferrentini Sampaio
NCE/UFRJ, Atlantica-Portugal,
IE/ULisboa
UFRJ/Universidade Federal do
Rio de Janeiro – Brasil e Lisboa -
Portugal
ffs@nce.ufrj.br

Nuno Dorotea
IE/ULisboa
Universidade de Lisboa
Lisboa - Portugal
nmdorotea@ie.ulisboa.pt

RESUMO

Este trabalho apresenta um estudo de caráter exploratório sobre as possibilidades do uso da Robótica Educacional no apoio aos educandos da rede pública estadual, no ensino-aprendizagem dos conceitos básicos de programação. Utilizou-se recursos de tecnologia livre, um ambiente de programação visual desenvolvido pelo 1o. Autor e a metodologia Problem Based Learning. O trabalho perdurou por um ano letivo e, como resultados preliminares, observou-se o entendimento de algumas habilidades fundamentais para desenvolvedores de software, tais como: compreensão de lógica sequencial, variáveis, fluxos de controle e repetição condicional, depuração de código, funções e procedimentos¹.

PALAVRAS-CHAVE

Robótica Educacional, Lógica de Programação, Ensino Médio, Educação Pública.

1 INTRODUÇÃO

O colégio estadual em questão é oriundo de uma parceria público-privada na cidade do Rio de Janeiro, com o intuito de apoiar escolas de ensino médio integral, integrado e profissionalizante unindo o ensino público regular tradicional a cursos técnicos em tecnologias digitais.

Um dos cursos técnicos oferecidos é o de Programação para Jogos e Aplicativos e uma das primeiras disciplinas técnicas que

os educandos cursam é a de Lógica de Programação. Esta disciplina visa ensinar, no primeiro ano do ensino médio, os conceitos básicos de programação para que, nos próximos dois anos, o foco recaia sobre o desenvolvimento de jogos/aplicativos.

A disciplina em questão foi projetada de maneira a permitir que o educando desenvolva conhecimentos iniciais sobre o funcionamento dos computadores e formas de operá-lo (programação) utilizando linguagens como o JavaScript associada ao desenvolvimento de pequenos jogos digitais.

É importante salientar que da Silva et al.[6] realizaram uma Revisão Sistemática da Literatura sobre o quão benéfico pode ser o uso de jogos digitais no ensino-aprendizagem de lógica de programação e, ao final do estudo, concluíram que, em 97% dos casos estudados, o uso de jogos pode ser uma ferramenta eficaz para ensinar programação. Desta forma, vincular o aprendizado básico de lógica de programação com o desenvolvimento de jogos digitais parece tornar este processo mais atrativo.

Além de competências técnicas, a disciplina procura fomentar nos educandos algumas competências gerais que são sugeridas pela BNCC[14], tais como:

- Exercitar a curiosidade intelectual, investigar, refletir e analisar de maneira que sejam capazes de elaborar e testar hipóteses, resolver problemas e criar soluções com base nos conhecimentos adquiridos.
- Compreender, utilizar e criar tecnologias digitais de forma crítica, significativa, reflexiva e ética nas diversas práticas sociais para se comunicar, acessar e disseminar informações, produzir conhecimentos, resolver problemas e exercer protagonismo e autoria na vida pessoal e coletiva.
- Agir pessoal e coletivamente com autonomia, responsabilidade, flexibilidade, resiliência e determinação, tomando decisões com base em princípios éticos, democráticos, inclusivos, sustentáveis e solidários.

Como descrito acima, a disciplina foi projetada de maneira que o educando seja apresentado ao que é programação de computadores, além de possibilitar o despertar de algumas competências sugeridas pela BNCC. No entanto, o que se tem

Fica permitido ao(s) autor(es) ou a terceiros a reprodução ou distribuição, em parte ou no todo, do material extraído dessa obra, de forma verbatim, adaptada ou remixada, bem como a criação ou produção a partir do conteúdo dessa obra, para fins não comerciais, desde que sejam atribuídos os devidos créditos à criação original, sob os termos da licença CC BY-NC 4.0.

EduComp'21, Abril 27-30, 2021, Jataí, Goiás, Brasil (On-line)

©2021 Copyright mantido pelo(s) autor(es). Direitos de publicação licenciados à Sociedade Brasileira de Computação (SBC).

notado ao longo dos últimos anos, é que as metodologias empregadas nesta disciplina inicial têm resultado em problemas de evasão e reprovação de alunos, gerando também inquietação entre os professores da escola. Esta inquietação direcionou estes professores a pesquisarem metodologias mais ativas para ensinar programação, já que o “método tradicional”, não tem alcançado resultados satisfatórios².

Como observado por Pontes[21] e Júnior[11], o aluno, geralmente novato, depara-se com uma série de problemas durante seu aprendizado, como a falta de organização do raciocínio, dificuldades no entendimento de questões de lógica relativamente complexas, incompreensão ou desconhecimento da língua inglesa e a falta de motivação gerada pelo medo prévio da disciplina.

Tais problemas estão, na maioria dos casos, atrelados ao que entendemos por metodologias tradicionais de ensino: aquelas que enfocam muito mais a memorização de conceitos em detrimento da prática de programação e resolução de tarefas que fazem parte do cotidiano do aluno, gerando assim uma grande evasão e repetência nos cursos/disciplinas iniciais de Computação[22][23].

Desta forma, pensando em tornar mais efetivo e prazeroso o ensino de programação, a equipe de professores da escola (responsáveis pelo curso técnico de programação) decidiu adotar o uso da Robótica Educacional como uma das ferramentas de apoio à disciplina de Lógica de Programação, ofertada no primeiro ano do ensino médio.

Outras leituras na área da Robótica Educacional (RE) possibilitaram perceber que a mesma permite, aos alunos envolvidos, expressar raciocínios e ideias através da resolução de tarefas de programação de objetos tangíveis, além de promover outros aspectos positivos para a sua aprendizagem como trabalho em equipe, interdisciplinaridade e autonomia[19].

A grande maioria desses trabalhos se apoia na proposta Construcionista de Papert[18], que afirma que a utilização de robôs como instrumento didático oferece muitas vantagens interessantes, tais como:

- O fato de serem objetos tridimensionais reais que se movem no espaço e no tempo, podendo simular comportamentos animais e humanos;
- Alunos aprendem mais depressa quando lidam com objetos tangíveis ao invés da manipulação “pura e simples” de fórmulas e abstrações;
- A motivação intrínseca que as crianças e os jovens têm pelos robôs geram um engajamento maior nas atividades que incluem este recurso.

Com base nas dificuldades enfrentadas no ensino de programação e da bibliografia consultada sobre os efeitos na aprendizagem de programação a partir do uso de jogos e da Robótica Educacional, o grupo de professores do curso técnico da Escola, em parceria com uma equipe de investigadores, propõe a reformulação da referida disciplina apoiada no uso

intensivo de atividades práticas com uso da RE e desenvolvimento de jogos.

Desta forma, o objetivo precípua do presente trabalho é o de investigar em que medida a RE, aliada ao uso de jogos, pode ser vista como alternativa viável para o ensino dos conceitos introdutórios de programação para alunos do ensino médio da rede pública.

Este trabalho está dividido da seguinte forma: A seção 1 apresenta a motivação e o problema a ser tratado neste trabalho. A seção 2 expõe, de forma sucinta, o arcabouço teórico no qual está baseada a presente pesquisa. A seção 3 exibe a metodologia de pesquisa que foi utilizada para orientar este trabalho. A seção 4 apresenta o ambiente de programação visual baseada em blocos DuinoBlocksII. A seção 5, por sua vez, descreve o experimento realizado com alunos do primeiro ano do ensino médio. A seção 6 faz uma análise dos resultados obtidos. O trabalho finaliza com a seção 7, na qual são apresentadas as conclusões.

2 ARCABOUÇO TEÓRICO

2.1 Teoria do Aprendizado Experimental

A Teoria do Aprendizado Experimental foi desenvolvida por Kolb em parceria com Ron Fry[13]. A mesma baseia-se no trabalho de estudiosos do século XX, que deram à experiência papel central em suas teorias de aprendizagem, de maneira a desenvolver um modelo abrangente deste processo de aprendizagem experiencial.

Esta teoria é construída sobre seis pilares:

- A aprendizagem é melhor contemplada como um processo e não em termos de resultados obtidos, ou seja, o foco principal deve estar no engajamento dos educandos em um processo que aprimore seus aprendizados e que inclua feedback sobre seus esforços de aprendizagem.
- Todo o aprendizado é reaprender. O aprendizado torna-se mais fácil quando faz o educando pensar além de suas crenças e ideias sobre algum tópico. Desta forma, o assunto pode ser reexaminado, testado e integrado a novas ideias mais refinadas.
- A aprendizagem requer a resolução de conflitos entre modos de adaptação dialeticamente opostos ao mundo. Conflitos, diferenças e desacordos conduzem ao processo de aprendizagem. Neste processo é desejável que se observe modos opostos de pensamentos.
- A aprendizagem é um processo holístico de adaptação ao mundo. Não é só o resultado da cognição, mas também envolve o funcionamento integrado total do indivíduo – pensando, sentindo, percebendo e se comportando.
- A aprendizagem resulta de transações sinérgicas entre o indivíduo e o meio ambiente, ou seja, em termos de Piaget, o aprendizado ocorre através do equilíbrio dos processos dialéticos de assimilar novas experiências em conceitos existentes e vice-versa.
- Aprendizagem é o processo de gerar novos conhecimentos.

² O presente trabalho aqui apresentado é uma síntese da dissertação de Mestrado do 1o. Autor[26]

A Teoria de Aprendizado Experimental é idealizada como um ciclo ou espiral de aprendizado, no qual o educando deve passar pelos quatro processos que são: experimentar, refletir, pensar e agir.

2.2 Robótica Educacional

Um dos aspectos da teoria Construtivista de Jean Piaget[20] considera a manipulação dos objetos a chave para as crianças construírem seu próprio conhecimento. Por sua vez, o Construcionismo de Seymour Papert[17] adiciona à teoria de Piaget, a ideia de que esta construção se dá de forma mais efetiva quando o aprendiz se engaja de maneira consciente na construção de algo tangível.

Papert[18] sugere a criação de uma disciplina de caráter construcionista com suas bases fundamentadas na Robótica Educacional e, desta forma, por ser fortemente baseada em atividades práticas e na manipulação de objetos físicos que fazem parte da realidade dos estudantes, alguns autores como Benitti[3] e Eguchi[7] defendem o emprego da Robótica Educacional como “porta de entrada” para o ensino de programação.

Chin et. al.[5] conclui em sua pesquisa que, de acordo com seus resultados, os educandos ficaram motivados em trabalhar com Robótica Educacional, indicando os benefícios potenciais da integração da robótica em atividades de ensino-aprendizagem, fomentando assim esta prática como motivador e potencializador do processo de aprendizagem.

O problema é que, até pouco tempo, tanto os custos quanto o acesso aos componentes eletrônicos para a criação de robôs inviabilizavam a disseminação da robótica na educação. No entanto, recentes avanços na área de desenvolvimento de hardware e o seu consequente barateamento permitiram uma facilidade no acesso a estes recursos, tornando viável sua introdução na educação.

2.3 Linguagens de Programação Visual

Diferentes autores como Resnick et al.[23] argumentam que as utilizações de linguagens de programação textual acrescentam barreiras desnecessárias no ensino inicial de programação. Tais autores defendem ainda alternativas como o uso de Linguagens de Programação Visuais (LPV), capazes de encapsular as dificuldades existentes nas sintaxes das linguagens textuais, ampliando o alcance do ensino de lógica de programação para diferentes idades.

Algumas iniciativas da utilização de LPVs começaram a surgir nos anos 90, porém, em 2003, o MIT (*Massachusetts Institute of Technology*) lançou o Scratch³. Este ambiente tem como objetivo permitir que qualquer pessoa, de qualquer idade, possa programar. Uma das suas principais características é o uso de LPV através de “blocos lógicos de encaixe” que permitem aos usuários organizarem suas ideias montando e desmontando seu programa de computador, além de visualizar os resultados, imediatamente, através da janela gráfica presente na interface do programa.

O Scratch é uma das principais referências com relação à introdução da lógica de programação, permitindo aos usuários

trabalhar conceitos de pensamento sistemático, desenvolver a criatividade e incentivar o trabalho em equipe[23].

Percebendo-se o valor da utilização de LPVs, não demorou para que fossem desenvolvidas novas ferramentas deste tipo, voltadas para o desenvolvimento de dispositivos robóticos, mais especificamente para placas de prototipagem Arduino⁴, tais como S4A⁵ e mBlock⁶. Também foram desenvolvidas ferramentas mais específicas para o ensino de lógica de programação através da Robótica Educacional, tais como DuinoBlocks[15] e DuinoBlocks4Kids[16].

Neste contexto, no capítulo 4 apresentamos o DuinoBlocksII (DBKII), uma Linguagem de Programação Visual baseada em blocos desenvolvida especificamente para a programação de placas Arduino, porém voltada para o desenvolvimento de jogos com a robótica.

2.4 Aprendizado Baseado em Problemas

Aprendizado Baseado em Problemas (do inglês *Problem Based Learning*, PBL) é uma abordagem para estruturar um currículo com o objetivo de estimular o aprendizado provocando os educandos com a apresentação de problemas - sem previamente ter apresentado possíveis formas de resolvê-los - para que busquem as informações e conceitos necessários à implementação da solução[4].

Savery[24] diz que o Aprendizado Baseado em Problemas, como modelo geral, foi desenvolvido no campo da medicina por volta da metade dos anos de 1950 e vem sendo refinado e adaptado para outras áreas de conhecimento até os dias atuais.

Wood[28] apresenta um modelo de como os educandos perpassam pela resolução de um problema usando ABP como metodologia. O primeiro passo é apresentar o problema para um grupo de estudantes e, após isso, os educandos irão discutir e analisar o problema. Nesta etapa do processo, os educandos têm pouco conhecimento e formulam hipóteses de como resolver o problema. Em seguida, são estimulados a pesquisar, individualmente, e buscar informações que os ajudem na busca pela solução. Nesta etapa, os educandos se reúnem novamente com novas informações e compartilham entre si estes conhecimentos adquiridos. Se perceberem que não chegaram a uma possível solução, eles devem voltar ao início do processo para a formulação de novas hipóteses. Caso uma solução seja encontrada, ela é implementada e ao final do processo o grupo reflete sobre como chegaram até a solução.

Atualmente, os educandos, principalmente crianças e adolescentes, estão cada vez mais resistentes a aulas puramente teóricas e de pouca interatividade. Pensando neste contexto, a utilização do Aprendizado Baseado em Problemas no currículo das disciplinas regulares e técnicas pode ser um bom caminho para facilitar e promover o processo de ensino-aprendizagem.

2.5 Pensamento Computacional

O termo Pensamento Computacional (PC) foi cunhado por Papert[17] e aperfeiçoado por Wing em artigo seminal publicado na *Communications of ACM* em 2006[27]. Segundo

⁴ <https://www.arduino.cc/en/Guide/Introduction>

⁵ <http://s4a.cat/>

⁶ <http://www.mblock.cc/try-scratch-arduino-using-mblock/>

³ <https://scratch.mit.edu/>

Wing[27], pensar de maneira computacional é olhar para um problema complexo e quebrá-lo em problemas menores a fim de facilitar a resolução do todo, prevenindo possíveis erros e revisando etapas no processo.

O objetivo de se pensar de maneira computacional não é fazer com que as pessoas mudem seu modo de pensar para o modo de como um cientista da computação pensa, mas sim, habilitá-las a utilizar esta forma de raciocínio lógico na busca de soluções de problemas em diversas áreas do conhecimento[10][2].

O PC pode ser considerado como o entendimento de uma série de habilidades que, de acordo com Grover e Pea[9], grande parte dos pesquisadores e educadores da área da computação tem aceitado. Podem-se destacar algumas destas habilidades, tais como: abstração e generalização de modelos, processamento sistemático de dados, entendimento de controle de fluxo de execução, decomposição de problemas maiores para facilitar sua resolução, pensamento paralelo, recursivo e iterativo, lógica condicional e depuração e detecção sistemática de erros.

Para trabalhar essas habilidades não é necessário apenas o uso da programação para computadores, mas pode-se utilizar esta ferramenta para potencializar o desenvolvimento e exercício de algumas destas competências cognitivas[9], ajudando assim em um dos objetivos deste trabalho.

2.6 Programação de Jogos Digitais

De acordo com Pontes[21], ao participar da construção de um jogo, o estudante explora, testa, controla e prevê os diversos eventos inerentes ao programa, trabalhando seu raciocínio hipotético dedutivo através da proposição e da programação das ações que serão disparadas, de acordo com os eventos ocorridos durante a execução do jogo.

Todo este processo acontece, na maioria dos casos, em um ambiente atrativo, em que princípios como imersão e interação colaboram no processo de ensino-aprendizagem[1].

Kafai[12] afirma que programar jogos digitais trabalha vários aspectos do aluno, tais como trabalho em equipe, percepção de modelos matemáticos, reformulação de conhecimento a partir dos feedbacks e, principalmente, ajuda na construção do raciocínio lógico.

Assim, a atividade de programação para construção de jogos mostra-se como uma alternativa eficiente no processo de ensino-aprendizagem, ampliando o domínio das competências trabalhadas pelo aluno e oferecendo um rico ambiente para descoberta e construção de conhecimento interdisciplinar[21].

3 METODOLOGIA DE PESQUISA

Neste trabalho foi empregado o método de pesquisa-ação, de caráter exploratório, em duas turmas de primeiro ano do ensino médio de uma escola pública no Rio de Janeiro. A pesquisa foi desenvolvida em dois ciclos, onde algumas mudanças foram implementadas no segundo ciclo, a partir dos resultados obtidos no primeiro ciclo.

É difícil reconhecer a origem da metodologia pesquisa-ação, pois as pessoas sempre investigaram a própria prática com a finalidade de melhorá-la[25] e também é difícil definir o

termo, pois é um processo que se apresenta tão natural, sob muitos aspectos e se desenvolve de maneira diferente para diversas aplicações, inclusive na educação.

A pesquisa-ação segue um ciclo no qual tenta-se aprimorar alguma prática através de investigações feitas sobre a mesma, ou seja, há um planejamento para realização de algo, implementa-se a atividade descrevendo o seu processo e avalia os resultados ao final do ciclo, propondo possíveis melhorias para o próximo.

Segundo Tripp[25] algumas características da pesquisa-ação que as difere de práticas rotineiras e pesquisas científicas são:

- Inovadora – discutível, pois o que já foi rotineiro anteriormente foi inovador em alguns aspectos;
- Contínua – pois não se pode, repetidamente, realizar pesquisas sobre a prática rotineira de alguém, mas deve-se trabalhar regular e frequentemente para melhorar um aspecto dela;
- Proativa – Age-se de maneira estratégica, porém não fica presa a grilhões metodológicos científicos que não se encaixam na prática em questão;
- Participativa – Não só o praticante rotineiro está envolvido na pesquisa, mas todos em questão;
- Intervencionista – Diferente da pesquisa-ação, a prática rotineira é naturalista, pois não é oficialmente pesquisada, não havendo intervenções sérias na situação;
- Problematizada – A demanda surge a partir de um problema;
- Deliberada – A pesquisa-ação é deliberativa, pois quando se intervém na prática de alguém, está se aventurando no desconhecido, de modo que é preciso fazer julgamentos competentes a respeito;
- Documentada – Não é rigorosamente científica, porém cada ciclo é documentado de maneira a avaliar como será o próximo;
- Compreendida – Explicar os fenômenos, uma vez que na rotina o importante é que a prática funcione sem maiores explicações;
- Disseminada – Na rotina, o conhecimento tende a permanecer com o prático, diferente do objetivo do pesquisador.

De acordo com estas informações, pode-se começar a reconhecer a situação em que fora aplicada esta pesquisa científica com uma turma de aproximadamente quarenta e cinco indivíduos.

Na situação atual os alunos estudam cerca de dez horas por dia, durante cinco dias na semana. As disciplinas oferecidas são aquelas de base regular proposta pela Secretaria de Educação e outras de cunho profissionalizante. O ano letivo é dividido em quatro bimestres. A proposta aqui apresentada foi aplicada na disciplina de Lógica de Programação. A mesma faz parte do curso profissionalizante e tem duração de aproximadamente três horas por semana durante todo o ano.

Durante o ano foram realizados vários exercícios baseados em PBL, sendo que alguns deles foram utilizados para gerar as notas dos educandos de cada bimestre (escala de 0 a 10,0). Para chegar a estes dez pontos, o professor precisa aplicar três avaliações e suas respectivas recuperações.

A divisão dos pontos fica a critério do professor, porém é recomendado que a pontuação de uma avaliação não seja tão discrepante com relação às outras. Cada avaliação possui parâmetros avaliativos que compõem a nota final e estes parâmetros dependem de cada tarefa que o educando esteja realizando.

Foram trabalhados dois ciclos semestrais de aplicação desta metodologia de pesquisa, de maneira que a implementação mostrou como foi a participação dos educandos durante os ciclos, gerando um relatório de pesquisa sobre os resultados das possíveis melhoras planejadas. Os dados foram analisados e os resultados serão apresentados na seção 6 deste trabalho.

4 A FERRAMENTA DUINOBLOCKSII

O DuinoBlocksII (DBKII) foi desenvolvido com o objetivo de motivar e facilitar o ensino de Lógica de Programação para usuários inexperientes na área de Computação. A ferramenta foi baseada nos trabalhos de Alves[15] com DuinoBlocks e Queiroz[16] com DuinoBlocks4Kids (DB4K) e permite a implementação de diferentes projetos na modalidade da RE, sendo capaz de controlar a maioria das placas de hardware Arduino existentes no mercado. A versão atual é multiplataforma, podendo ser executada nos principais navegadores web instalados na máquina do usuário, sem a necessidade de conexão com a internet.

O DBKII é uma versão similar ao DuinoBlocks4Kids, que fora implementada com, basicamente, os mesmos objetivos. No entanto, o DB4K é voltado para crianças do Ensino Fundamental I, possuindo uma interface com representações visuais mais detalhadas, além de funcionalidades mais concisas e limitadas, de maneira a abranger apenas opções de soluções de acordo com os desafios apresentados para alunos desta faixa etária. Com isso, percebeu-se que o DuinoBlocks4Kids não possuía a quantidade de funcionalidades necessárias para cobrir toda a ementa do curso de Lógica de Programação que o colégio oferecia, tornando seu uso inadequado para o contexto desta pesquisa.

Outro diferencial do DBKII é que ele possui funcionalidades que ajudam no desenvolvimento de jogos com artefatos robóticos. Essa necessidade se dá pelo fato do curso de programação ter como um de seus objetivos o ensino de programação através do desenvolvimento de jogos.

Um dos primeiros conceitos ensinados na implementação de jogos é o de coordenadas cartesianas. O objetivo é ter pontos (X, Y) para desenhos das imagens do jogo na tela do computador / dispositivo móvel. Precisávamos adaptar este conceito para o contexto da Robótica Educacional, no qual o resultado da programação não se dava em uma tela 2D (onde podemos representar as coordenadas nos eixos X – horizontal e Y – vertical na tela) e sim em um artefato físico.

Para isso, pensamos em criar um grid de LEDs na protoboard, em que cada LED representava um ponto X, Y no “plano cartesiano”. A protoboard foi coberta por uma folha de papel em branco, no qual os educandos fizeram as marcações dos eixos X e Y para facilitar o ponto cartesiano que cada LED representa.

E, para programar estes LEDs, foi implementado no DBKII, um bloco de programação que acende/apaga um LED de acordo com sua coordenada cartesiana específica no grid.

Outro diferencial importante do DBKII é a interação do usuário com o artefato robótico. Jogos costumam usar teclado, joystick e mouse/toque para interação do usuário com o software. Já em um dispositivo robótico, uma das principais formas de interação é através de botões.

O problema é que a programação usada nestes botões não é trivial para quem está iniciando no aprendizado de programação e, para isso, foi criado um bloco de programação no DBKII que abstrai toda essa complexidade, resumindo em apenas ficar monitorando o estado do botão (pressionado, não pressionado ou imediatamente não pressionado), ligado a uma porta específica do Arduino.

Desta forma, o educando pode redirecionar sua atenção apenas para a lógica do comportamento do artefato quando o botão for pressionado e não precisa se preocupar em programar o funcionamento do botão propriamente dito.

4.1 Interface

O ambiente DBKII apresenta ao usuário uma paleta de blocos de comandos organizados em sete categorias, sendo capaz de monitorar diferentes tipos de sensores e atuadores que agirão de alguma forma no mundo físico (Figura 1). São elas:

- Controle - possui blocos que controlam o fluxo do programa utilizando-se de estruturas de controle, repetição e delay;
- Operadores - contém operadores matemáticos, booleanos e condicionais de comparação;
- Entrada/Saída - inclui os blocos que fazem a interface com o ambiente externo, lendo as informações captadas por sensores e/ou enviando informações para os atuadores através das portas digitais/analógicas que o Arduino oferece;
- Variáveis - engloba blocos que manipulam variáveis numéricas e textuais no sistema;
- Comunicação - apresenta blocos capazes de ler e escrever valores na janela de comunicação serial, fazendo um interfaceamento em tempo real entre o experimento e o usuário
- (p.ex. auxiliando a depuração de programas);
- Matemáticos - agrega algumas funções matemáticas (trigonometria, radiação, etc);
- Motores - permite o controle de motores DC e Servo;
- Funções - permite a criação de blocos de funções que podem ser implementados pelo usuário.

A área opcional a direita, onde é apresentado o código nativo Wiring correspondente ao programa em blocos, permite ao educando inferir sobre o funcionamento da programação textual e, de maneira gradual, se familiarizar com a realidade da programação de computadores.



Figura 1: DBKII - Interface do ambiente: Menu de opções de blocos lógicos à esquerda, área da alocação dos blocos lógicos ao centro e tradução da programação em blocos para a programação textual (em Wiring) à direita.

5 PLANEJAMENTO DO ESTUDO REALIZADO

O estudo preliminar realizado com o DBKII aconteceu durante o ano letivo de 2018, dividido em dois ciclos, com duas turmas de 44 alunos do Ensino Médio (1º ciclo – turma 1003 e 2º ciclo – turma 1004) com conhecimentos gerais semelhantes (alunos oriundos da rede pública do estado) e que não possuíam conhecimento prévio de programação. Os materiais utilizados na disciplina foram 12 kits iniciantes de robótica da empresa Robocore⁷ e 3 kits montados separadamente, porém utilizando os mesmos componentes que os kits iniciantes oferecem.

Cada aula tinha a duração aproximada de 100 minutos e as atividades propostas eram inspiradas na apostila disponibilizada nos kits utilizados e adaptados para a ementa que o curso de Lógica de Programação já possuía. As atividades trabalhadas pelos estudantes da turma (1003) de robótica no 1º ciclo podem ser divididas em: apresentação das ferramentas de trabalho, apresentação de componentes básicos (LEDs, Buzzer, Display de 7 Segmentos, etc.), sistema de coordenadas cartesianas, aleatoriedade, estruturas de controle de fluxo (if-elseif-else e while), interação do artefato com o usuário (Push Buttons e Monitor Serial), depuração de dados e funções e procedimentos.

Em paralelo à turma 1003 que recebia as aulas de programação através da robótica, existia outra turma (1004) de programação, na qual o conteúdo ministrado era da ementa tradicional da disciplina baseado em programação textual e Javascript⁸. A ementa ministrada foi cuidadosamente preparada para que ambas as turmas recebessem o mesmo conteúdo, porém através de ferramentas diferentes, como resumidas a seguir: apresentação das ferramentas de trabalho, apresentação das primitivas básicas (retângulos, círculos, textos, etc.), sistema de coordenadas cartesianas, aleatoriedade, estruturas de controle de fluxo (if-elseif-else e while), interação do artefato com o usuário (mouse e teclado), depuração de dados e funções e procedimentos.

Como exemplo de exercício proposto, a avaliação 2 do 2º Bimestre buscava testar os educandos na utilização dos

conceitos de estrutura de controle de fluxo e entrada de dados, no sistema, pelo usuário.

No caso do exercício feito utilizando Javascript, a ideia era que fossem desenhados três quadrados na tela, um ao lado do outro. O quadrado mais à esquerda deveria estar na cor vermelha e os outros dois, cinza. Os educandos deveriam mapear duas teclas do teclado (A e D) para que o usuário do programa pudesse interagir com o sistema. Se a tecla D fosse pressionada, o quadrado que estava na cor vermelha deveria ficar cinza e o quadrado imediatamente à direita deveria ficar vermelho. Se a tecla A fosse pressionada, o quadrado que estava na cor vermelha deveria ficar cinza e o quadrado imediatamente à esquerda deveria ficar vermelho.

Se o quadrado em vermelho estivesse na extremidade esquerda ou direita e o usuário tentasse ir além deste limite, o quadrado que ficaria vermelho seria o do outro extremo.

No caso do exercício feito utilizando a robótica, os estudantes montaram um circuito com três LEDs, um ao lado do outro. O LED mais à esquerda deveria estar aceso e os outros dois, apagados. Os educandos deveriam mapear duas teclas do teclado (A e D) para que o usuário pudesse interagir com o artefato robótico. Se a tecla D fosse pressionada, o LED aceso deveria apagar e o LED imediatamente à direita deveria acender. Se a tecla A fosse pressionada, o LED que estava aceso deveria apagar e o LED imediatamente à esquerda deveria acender.

Se o LED que estava aceso estivesse na extremidade esquerda ou direita e o usuário tentasse ir além deste limite, o LED que acenderia seria o do outro extremo.

A ideia deste exercício era simular um seletor de opções dentro de um sistema e, apesar dos exercícios utilizarem praticamente a mesma solução para serem resolvidos, existiam pequenas diferenças que facilitavam a sua resolução utilizando a metodologia Javascript. Uma destas diferenças era que no Javascript, o mapeamento de entrada de dados no sistema via teclado não precisava de outro programa, enquanto na robótica é necessário a utilização da IDE nativa do Arduino para receber estas entradas via Monitor Serial.

Outra diferença era que, utilizando Javascript, a configuração inicial era apenas desenhar os quadrados na tela, enquanto na robótica os educandos precisavam montar um pequeno circuito. Esta é uma questão que é interessante ressaltar, pois a robótica apresenta este acréscimo de desafio aos educandos já que, além da programação de computadores, existia a parte eletrônica no processo de montagem do artefato robótico.

No segundo semestre (início do 2º ciclo), o processo das aulas inverteu-se entre as turmas, pois devido a uma prerrogativa da escola, ambas as turmas precisavam receber os mesmos conteúdos ao final do ano letivo. Com isso, a turma 1003 que, no 1º semestre, assistiu às aulas de programação com robótica, passou a assistir às aulas de Javascript e a turma 1004 assistiu às aulas de robótica.

Um diferencial entre o 1º e o 2º semestre, foi a forma como foram ministrados os conteúdos. No 1º semestre, os conteúdos eram ministrados sem conexões entre eles - havia uma aula expositiva e, tanto para exercícios, quanto para avaliações, um problema era apresentado para que os educandos pudessem resolvê-lo. Após as avaliações, um novo conteúdo era

⁷ <https://www.robocore.net/loja/produtos/arduino-kit-iniciante.html>

⁸ <https://developer.mozilla.org/en-US/docs/Web/JavaScript>

apresentado. Desta forma, as aulas teóricas e práticas sobre o uso de estruturas de controle não tinham continuidade com o uso de variáveis, por exemplo.

No 2º semestre foi proposto, para ambas as turmas, um projeto de jogo como "fio condutor" para o ensino/revisão de todo o conteúdo que já havia sido apresentado no 1º semestre. Cada etapa do jogo foi projetada para abranger os conteúdos curriculares que os educandos já haviam experimentado, de maneira separada, no semestre anterior. Desta forma, havia uma ligação entre os conteúdos, já que os educandos viam a sua aplicação sendo parte de um projeto maior e não de pequenos problemas individuais (como fora apresentado no 1º semestre). O projeto tratava de um jogo da memória, onde o objetivo do jogador era memorizar imagens sequencialmente para acertá-las após sua apresentação.

Esta mudança minimizou o desafio da montagem de circuito sobre cada exercício apresentado, pois eles aproveitavam o mesmo circuito previamente montado com adição de componentes novos, conforme o desenvolvimento do jogo ia evoluindo ao longo do semestre. Desta forma, o enfoque dos educandos ficava muito mais na codificação da solução do problema do que na parte eletrônica do processo.

Detalhando um dos exercícios apresentados neste segundo semestre, a avaliação 1 do 3º Bimestre testava os educandos no desenho de formas, em sequência (programação sequencial), que seriam apresentadas para o usuário memorizar. Utilizando Javascript, os educandos deveriam desenhar quadrados na tela de maneira a formar desenhos. A ideia era apresentar quatro desenhos diferentes, em sequência, com um tempo de espera entre eles e que ficassem se repetindo. Esta funcionalidade representa os desenhos que serão memorizados pelo jogador.

No caso do exercício utilizando a robótica, a ideia era fazer os desenhos serem apresentados através de LEDs, acendendo e apagando os mesmos. Foi montado, previamente, um grid com seis LEDs no circuito. Desta forma, era possível acender apenas os LEDs que representariam os desenhos que deveriam ser memorizados. O objetivo era mostrar quatro desenhos diferentes, em sequência, com um tempo de espera entre eles e que ficassem se repetindo.

Um questionário⁹ foi também apresentado nas turmas com o intuito de ter um feedback dos alunos sobre o andamento das aulas, o que foi mais interessante de aprender (robótica ou Javascript), o que poderia ser melhorado, entre outros.

6 ANÁLISE E RESULTADOS

O ano é dividido em quatro bimestres e em cada bimestre o aluno pode conseguir dez pontos no máximo. Para as turmas participantes desta pesquisa, foram propostos e realizados diferentes exercícios e avaliações que compuseram a nota final do educando. Por sua vez, cada avaliação possui parâmetros avaliativos que compõem a nota da mesma e estes parâmetros dependem de cada tarefa que o educando esteja realizando.

Estas notas geram alguns parâmetros quantitativos que podem revelar informações sobre as avaliações.

Para receber feedback de caráter opinativo dos alunos, foi apresentado aos mesmos um questionário com 9 perguntas, algumas abertas e outras no modelo da escala Likert. O questionário não foi obrigatório e 58 respostas foram obtidas, sendo 35 de alunos da turma 1004 e 23 da turma 1003.

Quando perguntados, em uma escala de 1 (pouquíssimo satisfeito) a 5 (muitíssimo satisfeito), se foi satisfatória a maneira como foi ministrada a disciplina. Considerando os valores de 4 a 5 satisfatórios da tabela de medição, os educandos responderam positivamente (83% para a turma 1003 e 74% para a turma 1004) a este questionamento.

A próxima pergunta procura saber qual abordagem foi mais prazerosa no ensino dos conteúdos da disciplina: Robótica Educacional (RE) ou Javascript (JS). Nos resultados, pôde-se observar que JS foi ligeiramente mais agradável (57%) segundo os educandos de ambas as turmas.

Em função da diferença do número de respostas entre as turmas (35 da turma 1004 e 23 da turma 1003), viu-se a necessidade de detalhar mais os resultados desta pergunta, para realmente validar a superioridade de aceitação do JS. De acordo com os resultados detalhados das turmas, de fato houve a confirmação que JS ficou à frente da RE (1003 – 61% e 1004 – 54%) como forma mais prazerosa de ensino de lógica de programação. No entanto, na turma 1004, que teve o maior número de respostas, a diferença de aceitação foi pequena.

Como a turma 1004 teve o primeiro contato com programação através do JS e só depois, no 2º semestre, trabalharam com robótica, parece que a robótica ajudou a fixar melhor os conteúdos que eles haviam tido no primeiro contato com o JS. Neste sentido, alguns alunos afirmaram que: "Prefiro Javascript, mas com a robótica a lógica ficou um pouco mais fácil".

Já na turma 1003, a aceitação da robótica foi muito menor que o JS. Duas possíveis razões para justificar tal resultado foram os problemas que os educandos, muitas vezes, enfrentavam na infraestrutura do laboratório e a inexperiência deles com tecnologias em geral.

Para que o DBKII conseguisse embarcar o código fonte no Arduino, é utilizado um código em background (Python). Esse código é transparente para os usuários, porém alguns educandos, de alguma forma, conseguiam editar ou até apagar este código das máquinas do laboratório. Isso impedia a execução do código implementado, obrigando o educando a mudar de máquina, causando perda de tempo e frustração.

Esse fato aconteceu muitas vezes ao longo dos ciclos, porém a turma 1003 - provavelmente pela sua inexperiência na manipulação da tecnologia - demoravam mais a perceber esse problema e mudar de máquina. A turma 1004, que viu robótica no segundo ciclo, já estava mais habituada com a infraestrutura e conseguia perceber o problema mais rapidamente.

Outro problema que pôde ser observado é o fato de que para trabalhar com robótica é necessário fazer algumas configurações iniciais, como a escolha da porta de comunicação que o DBKII usará para embarcar o código no Arduino. Era muito comum o educando esquecer esta configuração inicial, mesmo tendo sido orientado a fazê-lo em quase todo o início de aula.

⁹ <https://drive.google.com/file/d/1ZRSE1iqWuLUjj4m9hx11NneUhO95YtTO/view?usp=sharing>

Uma outra possível explicação para essa diferença de aceitação das metodologias é o fato da turma 1003 ter sido apresentada ao JS no segundo ciclo (semestre) e, já tendo alguma experiência em programação (e dos conteúdos trabalhados), foi capaz de dominar mais rapidamente a linguagem. Tal aspecto pode ter gerado nos alunos a impressão que JS era mais fácil. Esta justificativa é corroborada por feedbacks dados pelos alunos quando falavam o quanto foi interessante ver os dois tipos de abordagens trabalhando os mesmos conteúdos: "Interessante Pq (sic) vc (sic) pode achar uma abordagem melhor para vc (sic) do q (sic) outra e descobrir a melhor maneira de aprendizagem (sic)".

Conforme apresentado no capítulo 5, a pesquisa foi dividida em dois ciclos semestrais: no primeiro, os conteúdos foram ministrados com pouca ou nenhuma conexão entre eles, porém no segundo ciclo, os conteúdos foram encadeados de maneira a desenvolver o jogo ao final do ciclo.

No questionamento seguinte, foi perguntado sobre qual destes ciclos fora mais prazeroso o ensino dos conteúdos e 71% das respostas ressaltaram que é importante que haja uma conexão entre os conteúdos ministrados.

Outro aspecto importante a ser questionado foi a validação da metodologia de PBL, de maneira que a cada conteúdo ministrado, um ou mais exercícios eram realizados. Aproximadamente 70% das respostas mostraram satisfação quanto a esta estratégia.

Foram ministrados os seguintes conteúdos na disciplina: coordenadas cartesianas, loop principal de jogo, variáveis, tipos de dados e aleatoriedade, estrutura de controle de fluxo "if-else", input de dados via mouse / Push Button, depuração de dados, funções e procedimentos, input de dados via teclado e estrutura de repetição "while".

A próxima pergunta pedia para os educandos responderem, entre RE e JS, qual conteúdo foi mais interessante de ser aprendido. Os resultados mostram que JS fora preferido nos seguintes conteúdos: input de dados via mouse / Push Button, depuração de dados, input de dados via teclado e estrutura de repetição "while".

É possível que estes conteúdos tenham tido a preferência de JS, pois trabalhar com depuração de dados e input de dados via teclado tinham a necessidade do uso de um passo extra que era a comunicação serial com o Arduino, o que tornava um pouco mais complexa a visualização do resultado final. O uso do Push Button como input de dados também trazia um trabalho extra de montagem no circuito, já que na programação textual não havia necessidade, pois o mouse já estava "conectado" ao sistema. Apenas o uso da estrutura de repetição "while" não tinha diferença entre as abordagens, não deixando claro o motivo da preferência pela abordagem textual.

A próxima pergunta foi relativa às linguagens de programação utilizadas: DBKII (visual baseada em blocos) ou JS (textual). As respostas mostram superioridade (61%) da programação textual tradicional. Detalhando a pergunta em ambas as turmas, pode-se perceber uma porcentagem similar de preferência (1003 – 61% e 1004 – 57%) pela programação textual.

Abaixo serão listados alguns aspectos relevantes observados nesta pesquisa:

- As avaliações foram feitas durante os momentos dos exercícios que valem notas, sendo propostas como uma atividade prática a ser resolvida pelos alunos em duplas ou trios;
- Na turma 1003, foi ministrada RE no 1º semestre (ciclo 1) e JS no 2º semestre (ciclo 2) e as médias foram, do 1º ao 4º Bimestres, respectivamente 5.8; 5.2; 5.2 e 5.1. A média final e o desvio padrão foram respectivamente 5.325 e 8.09 e a média do 1º ciclo usando RE foi de 5.5 enquanto a média do 2º ciclo usando JS foi de 5.15;
- Na turma 1004, foi ministrada JS no 1º semestre (ciclo 1) e RE no 2º semestre (ciclo 2) e as médias foram do 1º ao 4º Bimestres respectivamente 6.0; 5.1; 6.4 e 5.5. A média final e o desvio padrão foram respectivamente 5.75 e 6.93 e a média do 1º ciclo usando JS foi de 5.55 enquanto a média do 2º ciclo usando RE foi de 5.95;
- Os dados acima levaram em consideração todos os alunos que foram matriculados nas turmas, porém alguns alunos não frequentaram as aulas ou desistiram no meio do ano letivo. Retirando esses outliers, os novos valores de média da turma 1003 foram, do 1º ao 4º Bimestres, respectivamente 6.2; 5.7; 5.7 e 5.7. A média final e o desvio padrão foram respectivamente 5.825 e 5.69 e a média do 1º ciclo usando RE foi de 5.95 enquanto a média do 2º ciclo usando JS foi de 5.7; retirando os outliers, os novos valores de média da turma 1004 foram do 1º ao 4º Bimestres respectivamente 6.4; 5.3; 6.8 e 6.0. A média final e o desvio padrão foram respectivamente 6.12 e 4.53 e a média do 1º ciclo usando JS foi de 5.85 enquanto a média do 2º ciclo usando RE foi de 6.4;
- Pôde-se observar que as médias semestrais nas quais os conteúdos foram ministrados através da RE e JS não foram tão discrepantes, variando menos de um ponto para ambas as turmas;
- Outro ponto importante a ser observado é que as médias nos semestres nas quais a RE foi aplicada foram ligeiramente melhores que as médias nos semestres em que JS foi ministrado;
- Alguns alunos dominaram rapidamente o processo de programação com blocos e passaram a construir seus programas utilizando a programação textual (Wiring), economizando tempo na depuração do código e se familiarizando cada vez mais com os tipos de códigos que eles vão se deparar no futuro em novas disciplinas que irão participar;
- Com relação à interface, todos os educandos souberam utilizar bem suas funcionalidades, porém notou-se alguma dificuldade em visualizar certas mensagens emitidas pelo DBKII na área de output da ferramenta.

7 CONCLUSÕES

O presente trabalho teve como objetivo investigar possibilidades do uso da RE e do desenvolvimento de jogos no aprendizado de programação por estudantes do ensino médio.

Os resultados das notas dos trabalhos obrigatórios ao longo do curso não nos permitem afirmar que houve uma diferença significativa entre as duas abordagens empregadas. Entretanto, as opiniões dadas pelos alunos, no questionário utilizado, sugerem que a Robótica Educacional, juntamente com o ambiente de programação visual DuinoBlocksII proporcionam cenários mais prazerosos e engajadores para o aprendizado de lógica de programação e desenvolvimento do pensamento computacional.

Também nos parece relevante afirmar o quão difícil é para o professor desenvolver atividades de caráter mais experimental em cenários de aprendizagem onde temos:

- Falta de apoio no gerenciamento dos recursos de hardware e software disponibilizados no laboratório de informática (exclusão do DBKII das máquinas pelos alunos, mau funcionamento do DBKII devido a alunos que conseguiram apagar arquivos fundamentais para execução do mesmo, problemas com os equipamentos dos laboratórios, perda de componentes dos kits de robótica que impediam grupos de realizar certas atividades, etc.).
- Turmas de tamanho excessivo (cerca de 40 alunos), onde muitos alunos eram leigos em informática básica;

7.1 Trabalhos futuros

É necessário que sejam feitos novos estudos com um controle maior sobre as variáveis observáveis (projeto e sequência pedagógica) a fim de investigar o potencial do DuinoBlocksII e da Robótica Educacional, auxiliando na implementação de diferentes estratégias pedagógicas.

Uma vez que boa parte dos alunos não possuía kits de robótica, não foi possível propor atividades extraclasses que os motivassem a avançar em alguns projetos desenvolvidos. Nesse sentido, a integração do DBKII com outras tecnologias passíveis de serem acessadas a distância, e sem a necessidade do educando possuir os kits de robótica, pode ser uma solução para problemas dessa natureza.

Uma outra possibilidade de ajuste à presente pesquisa é tornar a manipulação dos circuitos eletrônicos mais restrita, de maneira que os educandos se preocupem mais na programação dos artefatos do que na montagem integral dos mesmos.

Por último, mas não menos importante, verificamos ser necessária a adição de novos blocos de comandos no ambiente DBKII, a fim de torná-lo mais completo, possibilitando a utilização de novos componentes que ajudem no desenvolvimento do Pensamento Computacional dos alunos.

7.2 Validade da pesquisa e seus resultados

O desenho e implementação da presente pesquisa levou em conta o conjunto de questões apontadas por Hartmut[8] que orientaram a avaliação da validade interna da mesma. A seguir apresentamos algumas delas:

- As perguntas da pesquisa foram claramente formuladas.
- Os dados foram coletados em todos os contextos, tempos e pessoas sugeridos pelo delineamento.
- Detalhamento da análise levou em conta resultados não esperados e contrários aos esperados.

- A discussão dos resultados levou em conta possíveis alternativas de interpretação.

REFERÊNCIAS

- [1] Gláucio Aranha. 2006. Jogos Eletrônicos como um conceito chave para o desenvolvimento de aplicações imersivas e interativas para o aprendizado. *Ciências & Cognição* 7 (2006).
- [2] Valerie Barr and Chris Stephenson. 2011. Bringing computational thinking to K-12: what is Involved and what is the role of the computer science education community? *ACM Inroads* 2, 1 (2011), 48–54.
- [3] Fabiane Barreto Vavassori Benitti. 2012. Exploring the educational potential of robotics in schools: A systematic review. *Computers & Education* 58, 3 (2012), 978–988.
- [4] David Boud and Grahame Feletti. 2013. The challenge of problem-based learning (Abingdon).
- [5] Kai-Yi Chin, Zeng-Wei Hong, and Yen-Lin Chen. 2014. Impact of using an educational robot-based learning system on students' motivation in elementary education. *IEEE Transactions on learning technologies* 7, 4 (2014), 333–345.
- [6] Thiago Reis da Silva, Taina Jesus Medeiros, and Eduardo Henrique da S Aranha. 2014. Jogos digitais para ensino e aprendizagem de programação: uma revisão sistemática da literatura. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, Vol. 25. 692.
- [7] Amy Eguchi. 2010. What is educational robotics? Theories behind it and practical implementation. In *Society for information technology & teacher education international conference. Association for the Advancement of Computing in Education (AACE)*, 4006–4014.
- [8] Hartmut Gunther. 2006. Pesquisa qualitativa versus pesquisa quantitativa: esta é a questão? *Psic: Teor. E Pesq.* [Online]. (2006), vol.22, n.2 [cited 2021-01-19], pp.201-209.
- [9] Shuchi Grover and Roy Pea. 2013. Computational thinking in K-12: A review of the state of the field. *Educational researcher* 42, 1 (2013), 38–43.
- [10] David Hemmendinger. 2010. A plea for modesty. *ACM Inroads* 1, 2 (2010), 4–7.
- [11] José Carlos Rocha Pereira Júnior, Cleli Elena Rapkiewicz, Carla Delgado, and José Antonio Moreira Xexeo. 2005. Ensino de algoritmos e programação: uma experiência no nível médio. In *XIII Workshop de Educação em Computação (WEI'2005)*. São Leopoldo, RS, Brasil.
- [12] Yasmin Bettina Kafai. 1995. *Minds in play: Computer game design as a context for children's learning*. Routledge.
- [13] David Allen Kolb. 1984. *Experience as the source of learning and development*. Upper Saddle River: Prentice Hall (1984).
- [14] MEC/CONSED/UNDIME. 2018. Base Nacional Comum Curricular (BNCC). http://basenacionalcomum.mec.gov.br/images/BNCC_EI_EF_110518-versaofinal_site.pdf, month=oct, last accessed=Oct19,2020
- [15] Rafael Alves, & Fábio Ferrentini Sampaio. 2014. Duinoblocks: Desenho e implementação de um ambiente de programação visual para robótica educacional baseado no hardware arduino. In *Anais dos Workshops do Congresso Brasileiro de Informática na Educação* (Vol. 3, No. 1, p. 11).
- [16] Rubens Queiroz, Fábio Ferrentini Sampaio, & Monica Pereira dos Santos. 2016. DuinoBlocks4Kids: Ensinando conceitos básicos de programação a crianças do Ensino Fundamental I por meio da Robótica Educacional. In *Anais dos Workshops do Congresso Brasileiro de Informática na Educação* (Vol. 5, No. 1, p. 1169).
- [17] Seymour Papert. 1980. *Mindstorms: children, computers, and powerful ideas*. (1980).
- [18] Seymour Papert. 1993. *The children's machine: Rethinking school in the age of the computer*. ERIC.
- [19] António Sérgio Gomes Pereira. 2017. *A robótica no ensino e aprendizagem de programação* [em linha]. Lisboa: ISCTE-IUL, 2017. Dissertação de Mestrado. [www.hdl.handle.net/10071/15076](http://hdl.handle.net/10071/15076).
- [20] Jean Piaget. 1973. *To understand is to invent: the future of education*. (1973).
- [21] Herleson Paiva Pontes. 2013. *Desenvolvimento de jogos no processo de aprendizado em algoritmos e programação de computadores*. *Proceedings of the XII Simpósio Brasileiro de Games e Entretenimento Digital (SBGames)*. São Paulo (2013).
- [22] Rui Gonçalo Espadeiro & José Luís Ramos et al. 2015. *Pensamento computacional na escola e práticas de avaliação das aprendizagens. Uma revisão sistemática da literatura*. (2015).
- [23] Mitchel Resnick, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver,

- Brian Silverman, et al. 2009. Scratch: programming for all. *Commun. ACM* 52, 11 (2009), 60–67.
- [24] John Savery and Thomas Duffy. 1995. Problem based learning: An instructional model and its constructivist framework. *Educational technology* 35, 5 (1995), 31–38.
- [25] David Tripp. 2005. Pesquisa-ação: uma introdução metodológica. *Educação e pesquisa* 31, 3 (2005), 443–466.
- [26] José Augusto Mendes Vidal. 2019. Um estudo exploratório sobre o uso da Robótica Educacional como ferramenta de apoio ao ensino-aprendizagem de lógica de programação para alunos da rede pública do Ensino Médio. *Dissertação de Mestrado*. PPGI - Universidade Federal do Rio de Janeiro.
- [27] Jeannette M Wing. 2006. Computational thinking. *Commun. ACM* 49, 3 (2006), 33–35.
- [28] Diana Wood. 2003. Problem based learning. *BMJ* 326, 7384 (2003), 328–330.