RE@D - Revista de Educação a Distância e Elearning
Volume 5, Número 1
2022

Trabalho com programação para aprendizagem de conceitos de lógica

Working with programming to learn logic concepts

Raquel Santos
Escola Superior de Educação, Instituto Politécnico de Santarém, Portugal
raquel.santos@ese.ipsantarem.pt

Resumo: Ainda que a tecnologia tenha um crescimento exponencial e a programação já não seja uma novidade, a sua integração num contexto escolar, que vá para além do saber programar, é muito reduzida. São muitos os autores que defendem que aprender a programar tem de acontecer em contexto e, de preferência, para que os alunos aprendam conceitos de diferentes áreas. Para isso acontecer em sala de aula, é necessário que os professores passem por experiências semelhantes durante a sua formação inicial. Neste artigo apresenta-se um estudo de caso de uma turma no 1.º ano da Licenciatura em Educação Básica, em que os futuros educadores de infância e professores do 1.º e 2.º ciclo do ensino básico abordaram conceitos de lógica booleana através da programação. Faz-se uma descrição da experiência e apresentam-se as perspetivas, aprendizagens e dificuldades dos estudantes ao longo de toda a experiência. Os resultados são positivos, tanto ao nível do conhecimento de programação, como em relação às atitudes e perspetivas dos futuros educadores de infância e professores dos 1.º e 2.º ciclos do ensino básico quanto à sua utilização em sala de aula.

Palavras-Chave: Formação inicial de professores; Lógica Booleana; Programação; Scratch.

Abstract: Although technology has grown exponentially and programming is no longer a novelty, its integration in a school context, which goes beyond knowing how to program, is very low. Many authors argue that learning to program must happen in context and, preferably, for students to learn concepts from different areas. For this to happen in the classroom, it is necessary for teachers to have similar experiences during their teacher training. This paper presents a case study of a class in the first year of their Teacher Education Program, in which future kindergarten and elementary teachers approached concepts of Boolean logic through programming. A description of the experience is made, and students' perspectives, learning and difficulties are presented throughout the experience. The results are positive, both in terms of programming knowledge and in relation to the attitudes and perspectives of future kindergarten and elementary teachers regarding their use in the classroom.

Keywords: Boolean logic; Programming; Scratch; Teacher training.

Introdução

Com o crescente aumento da tecnologia na sociedade, é necessário preparar cidadãos que, para além de a utilizarem, sejam criadores, criativos, flexíveis e críticos no seu uso. Estudos internacionais (Joint Mathematical Council of the United Kingdom, 2011; Harris et al., 2009; Peralta & Costa, 2007; Ricoy & Couto, 2011) apontam que apesar do

investimento em tecnologia realizado nas escolas e de os professores mostrarem uma atitude positiva quanto ao seu uso e estarem conscientes das suas potencialidades, estes admitem que as tecnologia de informação e comunicação (TIC) assumem um papel secundário nas práticas dos professores, colocando essencialmente um foco no seu uso pelo professor, com recurso a software para criar apresentações, websites, quadros interativos e ferramentas de gestão para melhorar as práticas tradicionais.

Em Portugal, estudos confirmam essa perspetiva e apontam o facto de a formação inicial em TIC ainda ser "deficiente (...) ao nível das aplicações didáticas" das mesmas, apontando imensos constrangimentos para justificar esse facto (Ricoy & Couto, 2011, p. 113). Tinoca (2014) reitera mesmo que, apesar do investimento no desenvolvimento profissional para fomentar práticas inovadoras com recurso às TIC, raramente se investe na formação inicial de professores. Como consequência disso, são as instituições de ensino superior que devem tomar iniciativas nesse sentido (Peralta & Costa, 2007).

A investigação enfatiza ainda que o uso de tecnologia deve incentivar o questionamento (Harris et al., 2009) e deve fomentar uma aprendizagem contínua ao longo da vida (Martins et al., 2017). Ensinar e aprender a programar tem um papel crucial em capacitar os alunos com competências para criar as suas próprias tecnologias digitais e se integrar na sociedade atual (Balanskat & Engelhardt, 2015).

Em Portugal, como em diversos países, já se integra a programação na escola, mas ainda é considerada quase exclusivamente como uma disciplina à parte das restantes (Balanskat & Engelhardt, 2015). Em 2015, a Direção Geral de Educação (Figueiredo & Torres, 2015) promoveu a iniciativa "Iniciação à programação no 1.º ciclo", com o objetivo de desenvolver o pensamento computacional e a criatividade, articulando a programação com as outras áreas curriculares existentes. Também nesse ano, foi iniciado o projeto GEN10S Portugal de modo a "ensinar programação a crianças, promovendo a igualdade de oportunidades na área digital, reduzindo barreiras socioeconómicas e de género", envolvendo mais de 5000 alunos dos 5.º e 6.º anos, de todo o país, e os seus professores (SIC Esperança, 2021). Balanskat e Engelhardt (2015) consideram que a programação é uma área de investimento crucial para uma mudança curricular eficaz, mas que é necessário implementar formação inicial e contínua de professores nessa área.

Quando pensamos num determinado conceito matemático (neste caso a lógica), devemos considerar como ensinar as ideias associadas a esse conceito de um modo que a tecnologia apresente o conceito de uma forma compreensível para os alunos (Niess, 2005). Para trabalhar esses conceitos de lógica fez sentido, assim, que eles emergissem da programação, por ser um contexto em que estes conceitos surgem naturalmente. Adicionalmente, desenvolver formação no ambiente de programação Scratch pode mudar o paradigma de como a tecnologia é utilizada em sala de aula e desenvolver nos alunos o pensamento computacional e a criatividade. Portanto, neste artigo descreve-se uma experiência com estudantes da formação inicial de professores, em que estes aprenderam a programar de modo a fomentar o desenvolvimento de conceitos de lógica Booleana. Apresentam-se ainda as perspetivas, dificuldades e aprendizagens dos

estudantes durante a experiência, onde apontam vantagens e desvantagens do software Scratch.

Enquadramento teórico

O Scratch foi desenvolvido, em 2007, para que qualquer pessoa, independentemente da idade, contexto e interesse, pudesse aprender a programar histórias interativas, jogos, animações e simulações e as pudesse partilhar numa comunidade online (Resnick et al., 2009). É um software que funciona com blocos de comandos que encaixam como as peças de um lego, com uma linguagem simples e bastante atrativo no seu aspeto estético. A linguagem de programação que o Scratch utiliza tem ainda as características essenciais para isso acontecer: a) chão baixo, facilitando a iniciação; b) teto alto, possibilitando o desenvolvimento ao longo do tempo de projetos cada vez mais complexos; e c) paredes largas, motivando pessoas com diferentes interesses e estilos de aprendizagem (Resnick et al., 2009). Foi ainda pensado para ser uma linguagem de programação interativa, que promove a experimentação e significado aos utilizadores, permitindo a diversidade e a personalização, e que fomenta o lado social (Resnick et al., 2009).

Segundo Resnick et al. (2009), ser capaz de programar muda o paradigma de ser apenas consumidor para ser produtor de tecnologia, amplificando as oportunidades de aprendizagem. O objetivo em educação deve ser que os alunos aprendam a programar, mas também que aprendam programando (Figueiredo & Torres, 2015).

Há evidências de que a utilização deste programa em sala de aula tem implicações na aprendizagem dos alunos, nomeadamente no desenvolvimento da capacidade de resolução de problemas (Balanskat & Engelhardt, 2015; Calao et al., 2015; Calder, 2010; Resnick et al., 2009), do raciocínio lógico (Balanskat & Engelhardt, 2015; Calao et al., 2015; Calder, 2010; Resnick et al., 2009), da capacidade de modelação (Calao et al., 2015), da motivação (Calao et al., 2015; Calder, 2010), do trabalho colaborativo (Calder, 2010; Resnick et al., 2009) e de conhecimento matemático (Calao et al., 2015; Resnick et al., 2009), na área da geometria e medida, como a capacidade de visualização espacial, coordenadas, ângulos e comprimentos (Calder, 2010). Enquanto programam são desenvolvidas ainda outras capacidades importantes como o pensamento criativo, a comunicação eficaz, a capacidade de análise, a experimentação sistemática, o design interativo e a aprendizagem contínua (Monroy-Hernández & Resnick, 2015). Os erros que ocorrem na programação parecem ainda ter um efeito positivo nos alunos, uma vez que desencadeiam mais experimentação para atingir o movimento desejado (Calder, 2010).

A capacidade de produzir (e não só interagir com) conteúdo interativo é um ingrediente principal para alcançar a literacia digital e se tornar um participante no mundo interativo atual (Monroy-Hernández & Resnick, 2015).

Kordaki (2012) sintetiza as atividades de aprendizagem de Scratch que os alunos podem desenvolver em diferentes categorias: a) Atividade criativa livre; b) Resolução de um problema específico; c) Tarefa com múltiplas soluções; d) Experimentação num projeto;

e) Alteração de um projeto; f) Trabalho com visualização do aspeto final de um projeto em que é dada parte da programação correta; g) Trabalho com visualização do aspeto final de um projeto em que é dada a programação mas não ordenada corretamente; h) Trabalho com visualização do aspeto final de um projeto em que é dada parte da programação incorreta; i) Trabalho com a programação total de um projeto para prever a visualização do aspeto final; j) Trabalho com visualização do aspeto final de um projeto para descobrir toda a programação; k) Atividade de aprendizagem colaborativa.

A lógica Booleana foi desenvolvida em 1947 por George Boole e inclui as operações como a conjunção (... e ...), a disjunção (... ou ...), a negação (não ...) e a implicação (se ... então ...). A lógica Booleana é normalmente representada por tabelas de verdade, de modo a visualizar o comportamento de diferentes operações e o resultado de expressões lógicas. No entanto, essas abordagens tradicionais falham no sentido de dar significado e servir de conexão às experiências dos alunos, que mostram ter dificuldade em imaginar essa ligação e a compreenderem (Weng et al., 2010). Segundo Weng et al. (2010), a aprendizagem baseada em jogos e tecnologia pode ajudar os alunos a atingir essa compreensão, pelo facto de o jogo funcionar como um exemplo da vida real. Estes investigadores implementaram uma abordagem a conceitos de lógica através de jogos com alunos do ensino básico, com recurso ao Scratch, e concluíram que os alunos participaram ativamente nas tarefas e desenvolveram o seu conhecimento de lógica Booleana.

Neste artigo foi também implementada uma abordagem à lógica através da programação, utilizando a linguagem de programação Scratch e a possibilidade de experimentação sistemática e visualização dos efeitos dos operadores, esperando ainda contribuir para que os futuros educadores de infância e professores dos 1.º e 2.º ciclos criem uma ligação entre a linguagem lógica e a linguagem natural.

Metodologia

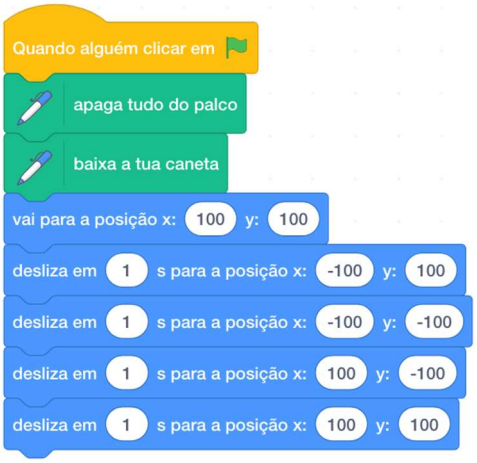
A metodologia adotada é de natureza qualitativa, na modalidade de estudo de caso, seguindo uma abordagem interpretativa. Os participantes são 13 alunos que frequentaram a unidade curricular de Introdução à Teoria dos Números, do 1.º ano do curso de Licenciatura em Educação Básica. Nesta unidade curricular está incluído um capítulo sobre conceitos de lógica Booleana, que se abordou através da programação utilizando o software Scratch. Para isso, os alunos realizaram diferentes tarefas durante três aulas, de modo a desenvolver o seu conhecimento desses conceitos. Depois de cada aula, os alunos preencheram um pequeno questionário, anónimo, com duas questões de escolha múltipla onde avaliaram o quanto cada aula contribuiu para a confiança e para a capacidade de programar em Scratch. Adicionalmente, existiam questões abertas de modo a verificar as vantagens e desvantagens para a sua aprendizagem observadas no software e as dificuldades e aprendizagens identificadas das diferentes tarefas.

A experiência de integração de programação para o ensino da lógica

A primeira aula, intitulada de Iniciação, serviu para que os estudantes tivessem o primeiro contacto com o programa Scratch. Puderam misturar um projeto já

completamente elaborado (figura 1) para perceber alguns tipos de comandos e a sua relação com o que o ator ia realizando no palco (atividade do tipo d na classificação de Kordaki, 2012). Era-lhes pedido, posteriormente, que alterassem alguns dos comandos sem alterar o resultado final, a construção do quadrado. Nessa tarefa, os estudantes puderam trabalhar comandos de movimento (coordenadas e ângulos de rotação), caneta, diálogo e ciclos de repetição.

Figura 1
Programa em Scratch de elaboração de um quadrado.



Para além disso, nessa mesma aula, tiveram que descobrir alguns erros noutros três projetos (figura 2) e corrigi-los (atividade do tipo f na classificação de Kordaki, 2012). Com essa tarefa, puderam trabalhar os eventos de início de projeto, a introdução de variáveis e os comandos de pergunta e resposta. Depois disso, os estudantes tiveram algum tempo para alguma experimentação livre (atividade do tipo a na classificação de Kordaki, 2012).

Figura 2
Erros trabalhados no programa Scratch.

"ErroA": Quando se clica na bandeira verde, tanto o Gobo como o Gato deviam começar a dialogar e a dançar, mas apenas o Gobo fala e dança.

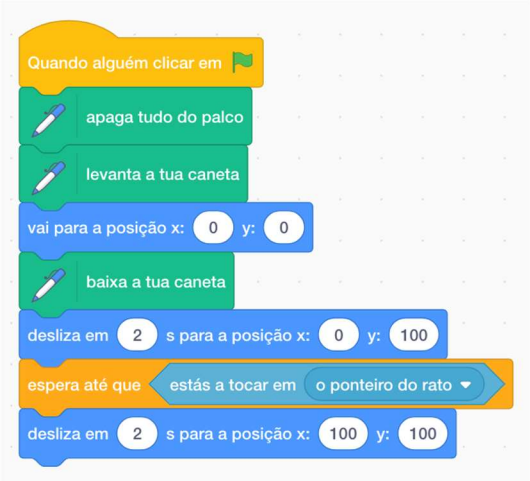
"ErroB": Quando se clica na bandeira verde, o Gato deve começar do lado esquerdo da Giga, dizer algo sobre estar nesse lado, deslizar para o lado direito do palco e dizer algo sobre estar nesse lado. Funciona da primeira vez que se clica na bandeira verde, mas não nas vezes seguintes.

"ErroC": Quando se clica na bandeira verde, é suposto o Gato contar de 1 até ao número que o utilizador escolher, mas o Gato conta sempre até 10, independentemente do número que o utilizador escolhe.

Depois de algum contacto e experimentação com o software, na segunda aula os estudantes foram levados a trabalhar a programação de modo a desenvolver os primeiros conceitos de lógica (lógica 1). Autonomamente, os estudantes trabalharam em pares numa tarefa que os levava a experimentar as consequências da negação de uma

proposição, a negação da negação de uma proposição, a conjunção de duas proposições e a disjunção de duas proposições (atividade do tipo k na classificação de Kordaki, 2012). Isto aconteceu uma vez que os alunos foram levados a construir um projeto em que o ator realizava duas ações diferentes, mas só realizava a segunda depois de certas circunstâncias serem cumpridas (usando os conceitos de lógica) (figura 3).

Figura 3
Programa construído em Scratch com duas ações distintas.



Na aula seguinte da unidade curricular, toda esta tarefa foi discutida em grande grupo e os conceitos trabalhados foram explorados, ainda que sem programação.

Na última aula de trabalho de lógica (lógica 2), os estudantes foram levados a construir, novamente em pares, um labirinto em que o ator tinha de o percorrer (figura 4), usando as setas para se deslocar, sem tocar nos limites do labirinto (para não perder vidas) (figura 5) (atividade do tipo k na classificação de Kordaki, 2012).

Figura 4
Labirinto utilizado com cenário do programa em Scratch.

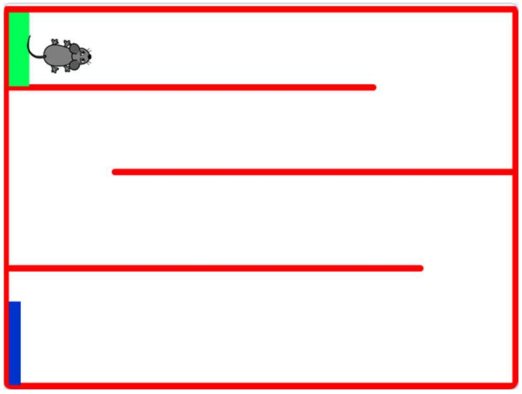


Figura 5
Parte do programa que define o que acontece quando se toca nos limites do labirinto.

Neste cenário havia um tempo limite e um número limite de vidas para chegar ao final do labirinto (figuras 6 e 7).

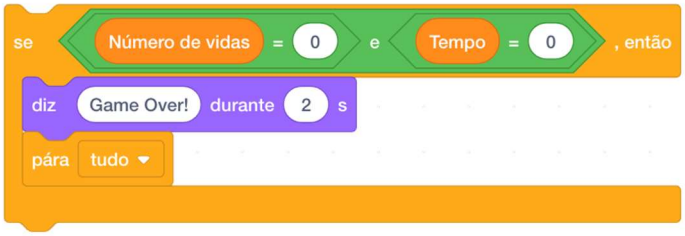
Figura 6
Parte do programa que define o número limite de vidas.

Figura 7
Parte do programa que define o tempo limite.

Posteriormente, os alunos puderam experimentar o que acontecia se definissem diferentes finais de jogo (a) sem vidas ou sem tempo (figura 8); b) sem vidas e sem tempo (figura 9)), fomentando o conhecimento de conjunções e disjunções.

Figura 8
Parte do programa que define o final de jogo quando se fica sem vidas ou sem tempo.

Figura 9
Parte do programa que define o final de jogo quando se fica sem vidas e sem tempo.



Adicionalmente, uma vez que deveria ser tocada uma melodia enquanto o jogo decorria, os estudantes eram incentivados a pensar nas primeiras Leis de De Morgan (negação da conjunção (figura 10) e negação da disjunção (figura 11)).

Figura 10
Parte do programa que define o toque da melodia enquanto o jogo decorria (negação da disjunção).

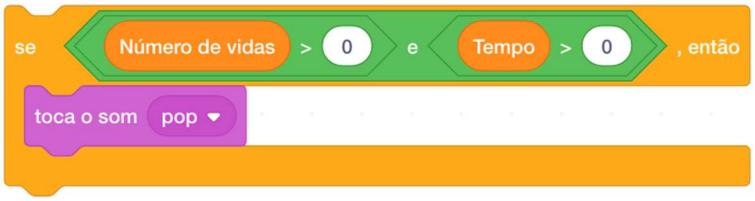
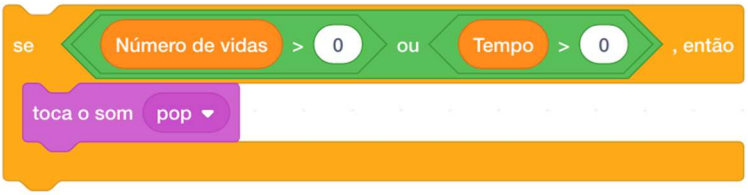


Figura 11
Parte do programa que define o toque da melodia enquanto o jogo decorria (negação da conjunção).



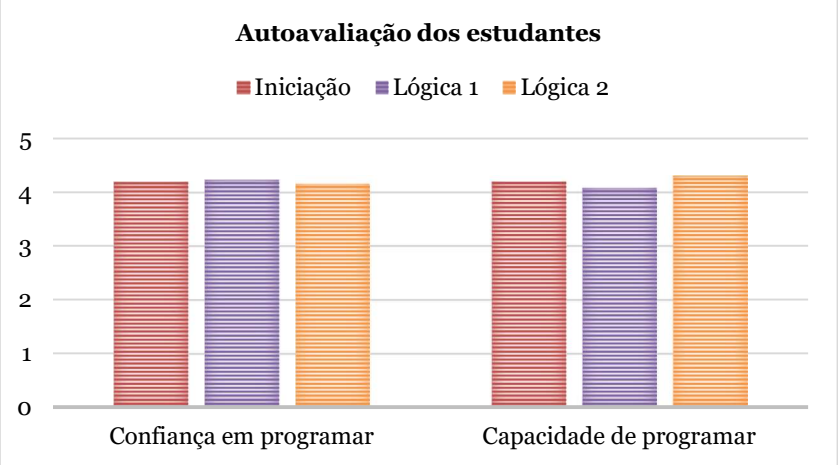
As perspetivas dos futuros professores e educadores

De um modo geral, os estudantes revelaram que as três aulas que tiveram com programação desenvolveram tanto a sua confiança como a sua capacidade de programar. Numa escala de 1 a 5, em que 1 representava "discordo totalmente" e 5 "concordo totalmente", em média os estudantes responderam 4,2, o que revela um

impacto positivo nos estudantes tanto sobre a sua confiança como sobre o seu conhecimento de programação (Figura 12).

Figura 12

Gráfico com as médias das avaliações dos estudantes no final de cada aula.



Analisando as respostas dos estudantes sobre as vantagens do Scratch para a sua aprendizagem, a maioria aponta a facilidade na sua utilização, ser simples, acessível e intuitivo. Também muitos estudantes referem a vantagem de este software dar "(...) oportunidade de usar a criatividade" e de dar "(...) uma certa liberdade porque tem uma gama alargada de personagens, cenários, comandos, personalizações, ...". Outra vantagem muito apontada é o facto de o software permitir a experimentação à medida que se vai programando: "o facto de podermos visualizar todos os passos", "podermos experienciar vezes infinitas o trabalho realizado" e "podemos aprender pondo à prova, e se errarmos reconhecer o nosso erro". Esta capacidade de visualização constante do trabalho parece ser um aspeto do software bastante apreciado pelos estudantes e que, potencialmente, diminui o seu medo de errar na resolução de problemas e aumenta a sua motivação para continuar a explorar: "cativante no sentido em que nos leva a querer explorar mais".

Ainda na vertente positiva, os estudantes, quando referiram as aprendizagens que realizaram com as tarefas, mencionam que a programação implica "entender a lógica por trás dos comandos, as ordens dos comandos, como é que se relacionam, etc." e, de um modo geral, fomenta capacidades como "raciocínio, resolução de problemas, procura de soluções, análise de informação". Em relação à aprendizagem dos conceitos de lógica, muitos estudantes indicam que a programação foi bastante útil para "aprender de forma mais assertiva, devido à aplicação prática, as proposições, sobretudo a Lei da Dupla Negação" e para a "realização de um labirinto utilizando comandos com disjunções e conjunções aprendendo assim melhor as suas funções". Nota-se, assim, que poder experimentar estes conceitos de lógica em programação, como na construção do labirinto, possibilitou que compreendessem de modo mais significativo e profundo esses conceitos.

No que diz respeito às desvantagens que apontam no Scratch e dificuldades nas tarefas, a maioria dos estudantes não encontra ou não menciona. Uma desvantagem funcional referida foi o facto de os comandos encaixarem tipo legos e a forma como se pode eliminar ou modificar um comando que esteja numa cadeia de comandos: "É complicado alterar a ordem dos comandos, poderia existir uma forma mais rápida de o fazer". De realçar, ainda, que alguns estudantes apontam que a diversidade de comandos pode levar à dispersão e que se demonstra mais trabalhoso, talvez por ter de se experimentar muitos comandos até encontrar o que faz o pretendido: "muitas opções em cada comando, sendo um pouco aliciante à distração", levando a que seja "trabalhoso devido ao facto de ter muitos comandos com que se trabalhar".

Relativamente às perspetivas dos estudantes sobre a utilização deste programa no seu futuro profissional, apesar de não lhes ter sido perguntado diretamente para refletirem sobre este assunto, quando referiram vantagens, alguns estudantes mencionaram este aspeto, principalmente depois das segundas e terceiras aulas: "programa didático para no futuro trabalhar com crianças" e "criação de um jogo didático para ser utilizado com as crianças". Contudo, parece ficar a perspetiva de utilizar este programa com crianças no sentido de as colocar a jogar, em vez de colocar as crianças a programar, o que certamente não vai ao encontro do potencial máximo do software.

Considerações finais

Os resultados da experiência de ensino e aprendizagem apresentados, da exploração de conteúdos de lógica Booleana através da programação, estão de acordo com a investigação no que concerne ao desenvolvimento da capacidade de experimentação sistemática e de design interativo criativo (Monroy-Hernández & Resnick, 2015), do raciocínio, da capacidade de resolução de problemas, do pensamento computacional (Resnick et al., 2009), da capacidade de aprendizagem através do erro e da motivação (Calder, 2010). A abordagem de ensino parece ter contribuído ainda para fomentar a confiança nos estudantes em programação e uma atitude positiva e intenção favorável na utilização deste software com crianças num futuro profissional, ainda que na perspetiva de colocar os futuros alunos a utilizar e não a programar.

De salientar ainda as dificuldades que os estudantes mostraram relativamente ao modo de encaixar e desencaixar os blocos de comando e na possibilidade de dispersão devido ao número elevado de opções que o Scratch proporciona. Este último obstáculo pode tentar ser colmatado com a construção de tarefas para crianças que os orientem de algum modo.

Conclui-se com o facto de, apesar de ter sido uma experiência muito curta para estes estudantes, a abordagem apresentada ter mostrado aspetos positivos para a aprendizagem de lógica através da programação (Weng et al., 2010). É necessário continuar a dar oportunidade aos futuros professores para explorar e aprender Matemática com recurso à tecnologia, de modo a fomentar a confiança e o conhecimento de tecnologia e de Matemática (Niess, 2005). Só colocando os professores, durante a sua formação inicial, a passar por experiências de integração da tecnologia, onde têm

um papel ativo na sua aprendizagem, podemos esperar uma mudança nas práticas nas escolas.

Referências

Balanskat, A., & Engelhardt, K. (2015). *Computing our future - Computer programming and coding -Priorities, school curricula and initiatives across Europe*. European Schoolnet. http://fcl.eun.org/documents/10180/14689/Computing+our+future_final.pdf/746e36b1-e1a6-4bf1-8105-ea27c0d2bbe0

Calao, L. A., Moreno-León, J., Correa, H. E., & Robles, G. (2015). Developing mathematical thinking with Scratch: An experiment with 6th grade students. In G. Conole, T. Klobučar, C. Rensing, J. Konert, & E. Lavoué (Eds.) *Design for Teaching and Learning in a Networked World* (pp. 17-27). Springer International Publishing. https://doi.org/10.1007/978-3-319-24258-3_2

Calder, N. (2010). Using Scratch: An integrated problem-solving approach to mathematical thinking. *APMC, 15*(4), 9-14. https://files.eric.ed.gov/fulltext/EJ906680.pdf

Figueiredo, M., & Torres, J. (2015). *Iniciação à Programação no 1.º Ciclo do Ensino Básico. Linhas orientadoras*. Direção-Geral da Educação. https://www.erte.dge.mec.pt/sites/default/files/Projetos/Programacao/IP1CEB/linhas_orientadoras.pdf

Joint Mathematical Council of the United Kingdom (2011). *Executive Summary. Digital technologies and mathematics education.* JMC. https://www.jmc.org.uk/documents/JMC_Report_Digital_Technologies_2011_Summary.pdf

Martins, G., Gomes, C., Brocardo, J., Pedroso, J., Carrillo, J, Silva, L., ..., & Rodrigues, S. (2017). *Perfil dos alunos à saída da escolaridade obrigatória*. Lisboa: Ministério da Educação – Direção-Geral da Educação. https://dge.mec.pt/sites/default/files/Curriculo/Projeto_Autonomia_e_Flexibilidade/perfil_dos_alunos.pdf

Monroy-Hernández, A., & Resnick, M. (2015). Empowering kids to create and share programmable media. *Interactions, 15*(2), 50-53. https://doi.org/10.1145/1340961.1340974

Niess, M. (2005). Preparing teachers to teach mathematics with technology. *Teaching and Teacher Education, 21*(5), 509-523. https://doi.org/10.1016/j.tate.2005.03.006

Peralta, H., & Costa, F.A. (2007). Competência e confiança dos professores no uso das TIC. Síntese de um estudo internacional. *Sísifo. Revista de Ciências da Educação, 3*, 77-86. http://hdl.handle.net/10451/7028

Kordaki, M. (2012). Diverse categories of programming learning activities could be performed within Scratch. *Social and Behavioral Sciences, 46*, 1162-1166. https://doi.org/10.1016/j.sbspro.2012.05.267

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan. K., …, Kafai, Y. (2009). Scratch: Programming for all. *Communications of the ACM, 52*(11), 60-67. https://doi.org/10.1145/1592761.1592779

Ricoy, M.C., & Couto, M.J. (2011). As TIC no ensino secundário na matemática em portugal: a perspectiva dos professores. *Revista Latinoamericana de Investigación en Matemática Educativa, 14*(1), 95-119. http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1665-24362011000100005&lng=es&nrm=iso

SIC Esperança (2021, fevereiro 25). *Gen10s Portugal*. https://sicesperanca.org/gen10s-portugal/

Tinoca, L. (2014). EduScratch: a case study on the development of key competences in Europe. *European Journal of Curriculum Studies, 1*(1), 23-32. http://pages.ie.uminho.pt/ejcs/index.php/ejcs/article/view/13

Weng, J., Tseng, S., & Lee, T. (2010). Teaching Boolean logic through game rule tuning. *IEEE Transactions on Learning Technologies, 3*(4), 319-328. https://doi.org/10.1109/TLT.2010.33