

Programação em Scratch para trabalhar a divisibilidade - experiência na formação inicial de professores

Raquel Santos

raquel.santos@ese.ipsantarem.pt
Escola Superior de Educação de Santarém

Contextualização

Nível de ensino: Ensino Superior

Disciplina: Introdução à Teoria dos Números

URL:

Descrição da experiência realizada

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). 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. 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 (Figueiredo & Torres, 2015). Há evidências de que a utilização do software Scratch em sala de aula tem implicações na aprendizagem dos alunos, nomeadamente no desenvolvimento da capacidade de resolução de problemas (Balanskat & Engelhardt, 2015; Calao, Moreno-León, Correa & Robles, 2015; Calder, 2010; Resnick et al., 2009), do raciocínio lógico (Balanskat & Engelhardt, 2015; Calao et al., 2015; Calder, 2010; Resnick et al., 2009), 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 literacia digital e se tornar um participante no mundo interativo atual (Monroy-Hernández & Resnick, 2015). Balanskat e Engelhardt (2015) consideram assim que a programação é uma área de investimento crucial para uma mudança curricular eficaz nas escolas, mas, para que isso aconteça, é necessário implementar formação inicial e contínua de professores nessa área.

Apresenta-se aqui uma experiência realizada numa turma do 1.º ano da Licenciatura em Educação Básica, em que os futuros educadores e professores utilizaram a programação em Scratch para trabalhar conceitos de divisibilidade. Participaram nesta experiência 36 estudantes (E1-E36), sendo 1 do sexo masculino, que frequentaram a unidade curricular de Introdução à Teoria dos Números. Essa unidade curricular aborda quatro capítulos diferentes sendo que um deles inclui conceitos de divisibilidade e outro conceitos de lógica. Neste capítulo de lógica, a docente (e autora da comunicação) decidiu implementar com os estudantes diferentes tarefas de programação em Scratch, ao longo de três aulas, de modo a abordar os conceitos de lógica enquanto aprendiam a programar. Como projeto final, foi proposto que estes estudantes, em grupo (G1-G9), elaborassem programas em Scratch que abordassem diferentes conceitos de divisibilidade:

- Programa para investigar se um número é ou não primo (G1);
- Programa para identificar todos os números primos até um número (Crivo de Eratóstenes) (G4);
- Programa para identificar os divisores de um número (G9);
- Programa para investigar o número de divisores de um número (G3);
- Programa para investigar se um número é perfeito, abundante ou deficiente (G8);
- Programa para explorar a fatorização em números primos de um número (G5);
- Programa para identificar o máximo divisor comum entre dois números (G6);
- Programa para investigar se dois números são primos entre si (G2);
- Programa para identificar o mínimo múltiplo comum entre dois números (G7).

Este projeto final foi realizado maioritariamente fora da sala de aula, colaborativamente e autonomamente, pelos estudantes, ainda que tenha existido dois momentos ao longo do semestre em que os grupos submeteram o trabalho desenvolvido e receberam feedback em aula e por escrito de modo a melhorarem e ultrapassarem algumas das dificuldades. Na última aula do semestre, os grupos apresentaram o projeto final à turma, numa aula aberta à comunidade escolar (figura 1) e submeteram um trabalho escrito onde realizavam um enquadramento curricular do conceito de divisibilidade trabalhado e descreviam o processo de construção do programa (apresentando dificuldades e aprendizagens).

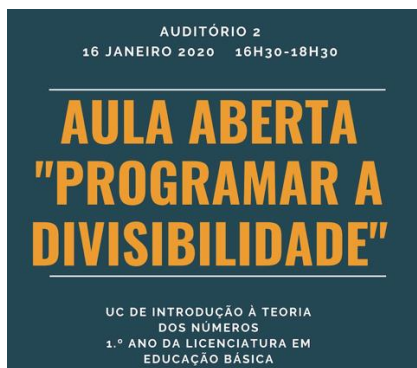


Figura 1. Parte da imagem de divulgação da aula aberta

Alguns exemplos de projetos construídos podem ser encontrados em: <https://scratch.mit.edu/projects/360254041> (G2), <https://scratch.mit.edu/projects/360253055> (G5), <https://scratch.mit.edu/projects/359035335> (G7) e <https://scratch.mit.edu/projects/359857483> (G8).

Analisando os trabalhos escritos submetidos, podemos verificar que os estudantes mencionam algumas dificuldades iniciais na construção dos programas, que após receberem feedback da docente ou pesquisando autonomamente foram ultrapassadas: “Após alguns dias a tentar este método e a não chegar a nenhuma conclusão, decidimos pedir ajuda à professora e chegámos à conclusão que este processo não era o mais adequado” (G2); “Ao início o trabalho deu-nos algum trabalho, só conseguíamos criar as falas, mas depois de alguma pesquisa e de alguns conhecimentos de amigos, conseguimos elaborar o nosso projeto” (G3); “De forma a ultrapassar as dificuldades encontradas, recorremos à ajuda da professora e à consulta de alguns suportes digitais, tais como websites” (G5).

Para além dessas formas de apoio, os obstáculos parecem também ter sido ultrapassados recorrendo a esquemas auxiliares de programação (figura 2) e ao processo de tentativa e erro: “Assim, recorremos a muitas tentativas-erro, o que originou uma confusão, no que diz respeito à organização dos comandos no Scratch” (G2); “Testamos o programa e reparamos que havia um erro que era o facto do morcego a dar o número de divisores incorreto (...). Ao revermos o código novamente para tentarmos descobrir o erro conseguimos perceber que o erro vinha da variável “iterador” (...)” (G3).

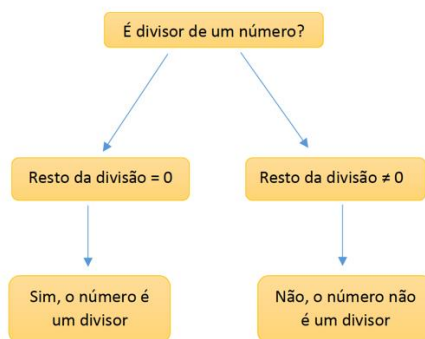


Figura 2. Esquema auxiliar de programação do grupo 3

As descrições que os estudantes fizeram da construção dos programas evidenciam também que existiram aprendizagens significativas tanto ao nível de conhecimentos de programação, como do próprio conhecimento dos conceitos de divisibilidade: “este projeto aumentou a nossa capacidade de utilizar o programa Scratch, permitiu uma melhor compreensão do conteúdo utilizado para a execução do mesmo (números primos ou não primos) e aumentou os conhecimentos a nível do conteúdo e da “arte de programar”” (G1); “No decorrer deste trabalho, podemos concluir que aprofundámos os nossos conhecimentos sobre divisibilidade (mínimo múltiplo comum) e também sobre lógica – programando em Scratch” (G7).

Este tipo de projeto proporcionou também o aumento da motivação e o divertimento em aprender: “Em suma, nós pensamos que apesar de existirem sempre aspetos a melhorar, foi um trabalho bem concedido e bastante divertido” (G3); “Foi um projeto complicado e demorado, mas gostámos muito de o realizar” (G9).

Adicionalmente, alguns grupos mostram a capacidade de perspetivar a utilização deste recurso para fomentar a motivação e o conhecimento de futuros seus alunos: “Futuramente, acreditamos que este programa será um bom meio para lecionar diversos conteúdos, a nível da Matemática, uma vez que, facilitará a compreensão dos mesmos e estimulará o processo de aprendizagem, devido ao seu fator apelativo” (G1).

Conclui-se que esta experiência culminou numa apresentação onde foi evidente o orgulho, o entusiasmo e o desenvolvimento do seu conhecimento de programação, assim como o de alguns conceitos de divisibilidade.

Referências bibliográficas

- Balanskat, A., & Engelhardt, K. (2015). *Computing our future - Computer programming and coding -Priorities, school curricula and initiatives across Europe*. Brussel: European Schoolnet. Disponível em 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. In G. Conole, T. Klobučar, C. Rensing, J. Konert, É. Lavoué (Eds.), *Design for Teaching and Learning in a Networked World* (pp. 17-27). Springer International Publishing.
- Calder, N. (2010). Using Scratch: An integrated problem-solving approach to mathematical thinking. *APMC*, 15(4), 9-14.
- Figueiredo, M., & Torres, J. (2015). *Iniciação à Programação no 1.º Ciclo do Ensino Básico. Linhas orientadoras*. Lisboa: Direção-Geral da Educação.
- 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.