

# Utilização do *Scratch* no Ensino e na Aprendizagem da Matemática: Uma experiência de formação

RAQUEL SANTOS  
NEUSA BRANCO

## INTRODUÇÃO

Vários países europeus, incluindo Portugal, estão a integrar a programação nas suas atividades curriculares ou como complemento às atividades escolares (Balanskat & Engelhardt, 2014). A ideia geral é que os alunos aprendam a programar, e também que aprendam programando. Balanskat e Engelhardt (2014) identificam a preparação dos professores para as mudanças curriculares como uma área de investimento essencial para o sucesso efetivo dessas mudanças e dos resultados da aprendizagem. 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.

A formação de professores para a utilização do *Scratch* deve envolvê-los ativamente na sua experimentação e na realização de tarefas de natureza diversificada para a sua exploração. Assim, neste artigo apresentamos um relato de uma experiência de utilização do *Scratch*, realizada na formação contínua de professores dos 1.º e 2.º ciclos, sem experiência ou com experiência pontual neste programa. A proposta envolve tarefas de diferente natureza que visam a introdução aos comandos do programa e a abordagem a conceitos de geometria, com vista à discussão da utilização do *Scratch* com alunos para a promoção da aprendizagem matemática.

## SCRATCH E O ENSINO-APRENDIZAGEM

O *Scratch* é um ambiente de programação que permite às crianças programar e compartilhar histórias interativas, jogos e animações. Segundo os seus criadores (Resnick et al., 2009), este ambiente de programação tem as características fundamentais que as linguagens de programação devem possuir: um chão baixo, facilitando a iniciação e podendo ser utilizada desde o pré-escolar; um teto alto, tornando os projetos progressivamente mais complexos, motivando o seu uso até por adultos; e paredes largas, permitindo diferentes tipos de projetos, independentemente do interesse e contexto. Além disso, pretende ser mais prático e interativo, incentivando à experimentação; ter mais significado, incitando à diversidade e personalização; e ser social, devido à associação com o *website* do *Scratch* que incorpora uma comunidade (Resnick et al, 2009). É uma ferramenta que utiliza blocos de comandos, que se unem como legos, não incorporando a complexidade existente numa linguagem de programação, e que permite a integração de gráficos, imagens, fotos, música e som, fomentando um ambiente de aprendizagem motivante.

Segundo Kordaki (2012), existem 11 tipos de tarefas com o *Scratch* que podem ser realizadas por alunos: a) Atividades criativas livres; b) Resolução de um problema específico; c) Tarefas com soluções múltiplas; d) Experimentação dentro de um projeto; e) Modificação de um projeto; f) Trabalho com um projeto correto, mas com parte do código

incompleto; g) Trabalho com um projeto correto e uma mistura do seu código; h) Trabalho com um projeto com parte do código incorreto; i) Trabalho com um projeto de modo a prever o que acontece; j) Atividades de caixa negra, em que se vê o que acontece e se cria um projeto que replique o observado; e k) Atividades de aprendizagem colaborativa.

A utilização deste programa em aula tem já fomentado investigações sobre as implicações na aprendizagem dos alunos. Estas evidenciam, nomeadamente, que o *Scratch* desenvolve nos alunos a capacidade de resolução de problemas (Calao, Moreno-León, Correa & Robles, 2015), a capacidade de modelação matemática (Calao et al., 2015), o raciocínio matemático (Calao et al., 2015) e o conhecimento matemático, como conceitos de geometria e medida (coordenadas, ângulo e comprimento) (Calder, 2010). O programa fomenta ainda outras capacidades essenciais, como pensamento criativo, comunicação efetiva, análise crítica, experimentação sistemática, *design* iterativo, aprendizagem contínua (Monroy-Hernández & Resnick, 2008) e empreendedorismo (Tinoca, 2014). Os erros que ocorrem na programação parecem ter efeito positivo nos alunos, uma vez que desencadeiam mais experimentação para atingir o movimento desejado (Calder, 2010).

## ABORDAGEM DE INICIAÇÃO AO SCRATCH

Para uma primeira abordagem ao ambiente de programação propusemos duas tarefas. A Tarefa 1 apresenta um pro-

jeto já construído (figura 1), fomentando o contacto com diferentes áreas do programa e a correspondência entre as ações do ator e a sequência de comandos dada.

### TAREFA 1 – INTRODUÇÃO.

Análise de um projeto já construído no *Scratch*: “1.Intro”. Analise o que o projeto leva o *Scratch* a fazer, tentando interpretar cada um dos comandos utilizados.

Nesta tarefa, os formandos tomaram conhecimento das diferentes áreas do programa e discutiram o que cada comando significa e a ordem em que aparece. Os formandos identificaram o desenho de segmentos de reta, a alteração da cor e o aumento gradual da espessura desses segmentos, bem como a mudança de direção. Reconheceram também o papel do comando que faz surgir os segmentos, mas destacaram o facto de poder ser difícil para os alunos identificar todos os segmentos pelas limitações que o palco pode introduzir, uma vez que durante o desenho de segmentos, se o ator atingir o limite do palco, é desenhado um novo segmento com outra direção. As alterações de cor e de espessura foram apontadas como positivas numa implementação com alunos pois ajudam a evidenciar os segmentos, muito embora não tenham significado matemático. Contudo, os professores apontaram a necessidade de fazer algumas alterações, de acordo com o ano de escolaridade, para melhorar a análise a fazer pelos alunos, por exemplo, mudando o número de repetições, o nú-

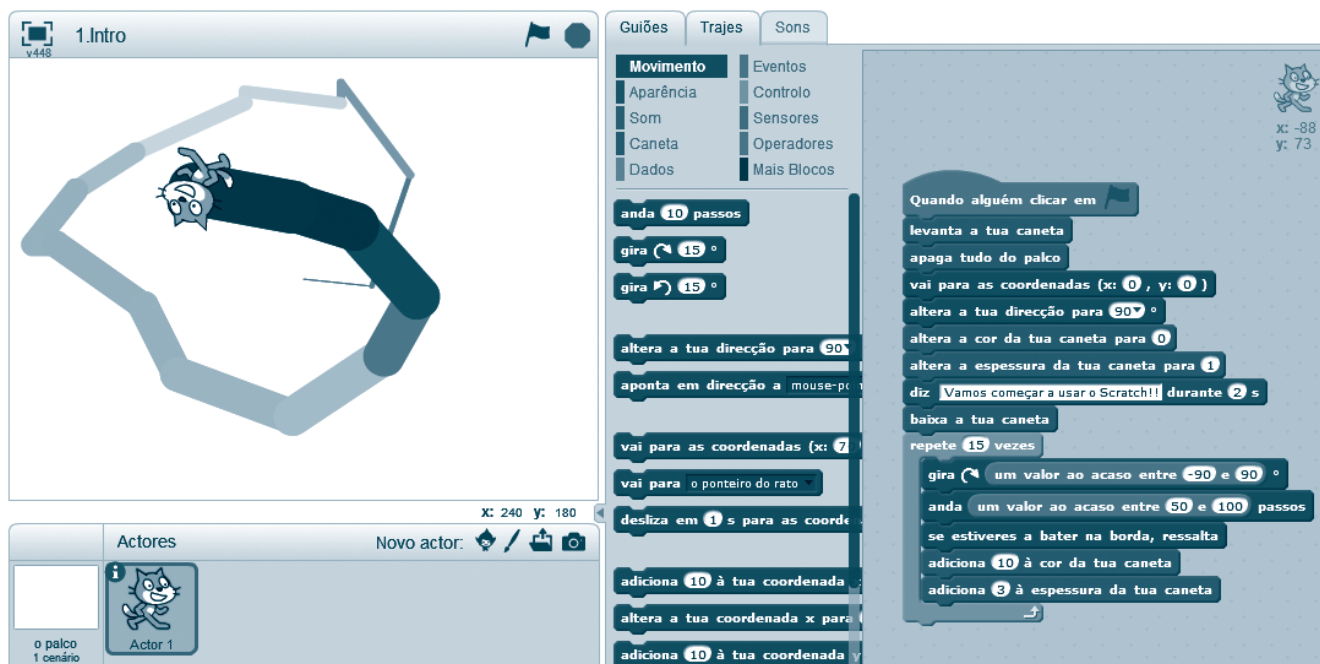


Figura 1. Projeto “1.Intro”

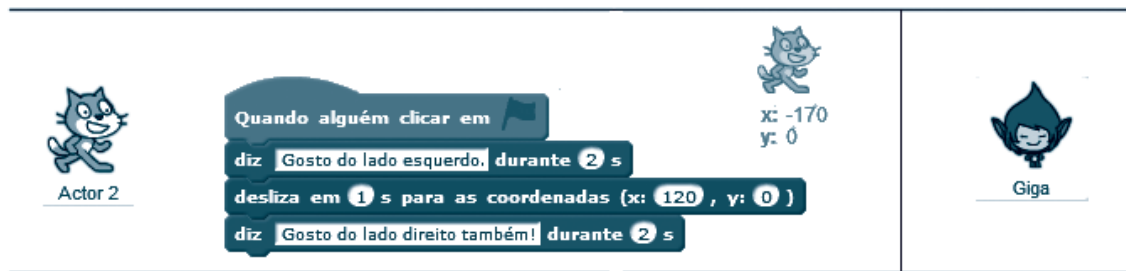


Figura 2. Projeto “2. Erro”

mero de passos, ou o valor a adicionar à cor. Os professores assinalaram ainda a referência às coordenadas na sequência de comandos como mote para uma discussão com alunos sobre o referencial cartesiano, com origem no centro do palco, e as coordenadas do ponto referente à posição do ator.

A Tarefa 2 propõe a análise de projetos já construídos, desta vez de modo a encontrar erros ou omissões na sequência de comandos e a corrigi-los, sendo aqui apresentado o Erro 2.

### TAREFA 2 - À DESCOBERTA DO ERRO (ERRO 2)

Análise de um projeto já construído no *Scratch* de modo a identificar o erro e a corrigi-lo. “2. Erro”. Quando se clica na bandeira verde, o Ator 2 deve começar do lado esquerdo do palco (à esquerda do Giga), dizer algo sobre estar nesse lado, deslizar para o lado direito do palco e dizer algo sobre estar nesse lado (à direita do Giga). Funciona da primeira vez que se corre o programa ao clicar na bandeira verde, mas não nas vezes seguintes.

Este projeto tem dois atores, o Giga que não tem qualquer ação e se encontra posicionado no centro do palco e o Actor 2 que inicialmente está no lado esquerdo do palco e cuja sequência de comandos é apresentada na figura 2.




Esta situação promoveu a análise do posicionamento de um ator no referencial cartesiano e do assumir de uma dada posição num determinado momento. Para corrigir este programa, os formandos identificaram os valores da abcissa e da ordenada que permitem ao Actor 2 posicionar-se inicialmente no lado esquerdo do palco, recorrendo a uma abcissa negativa. Os formandos apresentaram e discutiram duas maneiras de solucionar o erro, colocando um comando de movimento para essas coordenadas no final da sequência de comandos (para que o ator volte à posição inicial no final da sequência) ou no início (para colocar o ator sempre do lado esquerdo no início de cada simulação). Quanto à discussão sobre o trabalho em aula, os formandos apontaram a pertinência de tarefas desta natureza, em que há um erro ou omissão na sequência de comandos que os alunos devem detetar e corrigir, podendo envolver situações com diferentes graus de complexidade e tipos de comando.

## SCRATCH E A CONSTRUÇÃO DE POLÍGONOS REGULARES


A Tarefa 3 propõe a utilização do *Scratch* para trabalhar Geometria, no que respeita às propriedades de polígonos regulares. Apresenta uma sequência de comandos de um ator, o Caranguejo, para a construção de um triângulo equilátero (figura 3) e depois da sua análise sugere-se a construção de sequências de comandos para outros atores para a construção de outros polígonos regulares.

### TAREFA 3 - CONSTRUÇÃO DE POLÍGONOS REGULARES.

1. Abrir o documento *Scratch* “3. Polígono”;
2. Analisar os comandos do caranguejo de modo a perceber como este constrói um triângulo equilátero;
3. Incluir novos atores de modo a que estes desenhem




novos polígonos (  ,  ,  ). Para isso, escolher cada um dos atores na biblioteca, como mostra a figura ao lado.

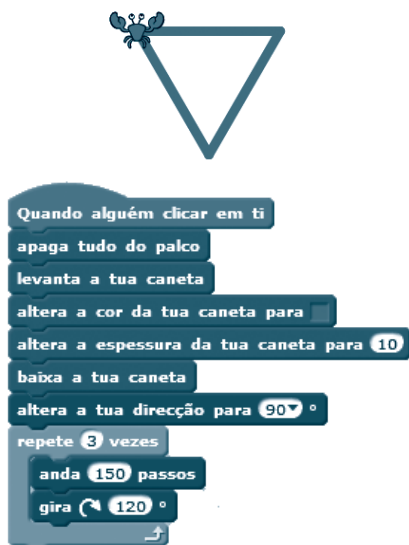


4. Para que todos os atores caibam no palco, é necessário encolher as imagens. Para isso, selecionar “reduzir” (  ) e clicar em cima dos diferentes atores.



5. Arrastar os comandos do caranguejo para cada um dos outros atores. Isto faz uma cópia dos comandos.
6. Analisar agora que alterações são necessárias para a construção de cada um dos polígonos em cada um dos diferentes atores:

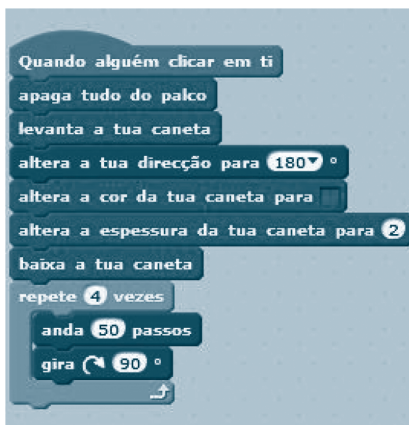
- a.  Carocha (polígono regular com 4 lados);
- b.  Peixe (polígono regular com 5 lados);
- c.  Hipopótamo (polígono regular com 6 lados).



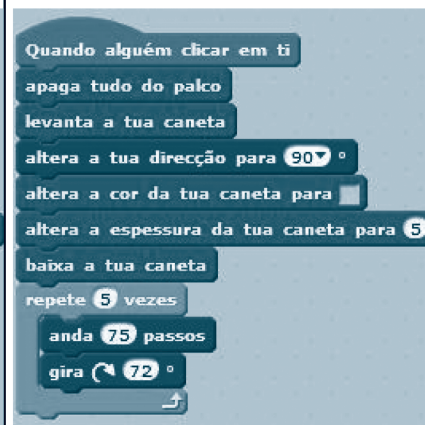
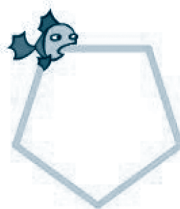
**Figura 3.** Construção de um triângulo equilátero

Nesta tarefa os formandos identificaram os diferentes tipos de comandos utilizados na construção do triângulo. Referiram que, ao implementar esta tarefa em aula, pode ser discutido o conceito de polígono regular e a classificação de

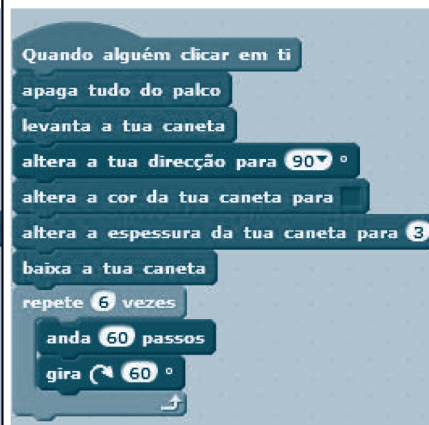
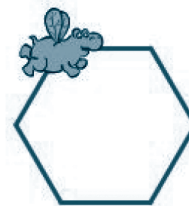
cada polígono. Os professores fizeram alterações nos valores dados para identificarem a sua influência na construção obtida, nomeadamente no número de passos colocado em cada ciclo de repetição verificando que este determina o comprimento do lado do polígono. Da discussão na sessão emergiram dois aspetos a analisar em aula com os alunos: i) o número de lados de cada polígono que influencia o número de repetições a indicar no comando, e ii) a amplitude do ângulo da rotação a realizar pelo ator de modo a se obter o polígono regular pretendido. Em relação ao primeiro aspeto, os formandos destacaram que os alunos se podem esquecer de indicar no mínimo tantas repetições quanto o número de lados do polígono de modo a obter uma linha poligonal fechada. Além disso, alguns formandos encontraram como obstáculo a posição inicial do ator. Se este estiver próximo dos limites do palco, podem não obter uma linha poligonal fechada devido à falta de espaço para respeitar o comprimento dos lados indicado. O segundo aspeto envolveu uma discussão mais aprofundada sobre as propriedades dos polígonos, nomeadamente sobre a análise da amplitude do ângulo da rotação que é realizada no triângulo equilátero,  $120^\circ$ , que corresponde à amplitude do ângulo externo. Assim, analisou-se a possibilidade de chegar a esta amplitude do ângulo externo do triângulo equilátero a partir da amplitude do ângulo interno ( $\frac{180^\circ}{3} = 60^\circ$ ), verificando que estes são suplementares  $180^\circ - 60^\circ = 120^\circ$ , ou pela soma das amplitudes dos ângulos externos ( $\frac{360^\circ}{3} = 120^\circ$ ). Também para os restantes polígonos (figuras 4, 5 e 6), os



**Figura 4.** Construção de um quadrado.



**Figura 5.** Construção de um pentágono regular.



**Figura 6.** Construção de um hexágono regular.

formandos identificaram que o ator realiza a rotação de centro no vértice e de ângulo de amplitude igual ao ângulo externo do polígono. Sobre a sua implementação em aula, os formandos referem que esta tarefa constitui uma oportunidade para realizar a generalização da amplitude de um ângulo externo de um qualquer polígono regular.

Durante a sessão, os formandos mostraram-se bastante motivados, fazendo por vezes muito mais do que era pedido, com a curiosidade de ir experimentando diferentes comandos. Refletiram também sobre a possível implementação com alunos, notando que o programa pode fazer toda a diferença, até para alunos que manifestam mais dificuldade em Matemática, uma vez que incita à experimentação, mesmo quando se erra. Foi também apontada a vantagem de ser uma ferramenta atrativa, utilizar imagens apelativas e permitir vários tipos de projetos, mais simples ou mais complexos, envolvendo, neste caso, ideias matemáticas importantes (sistemas de coordenadas, polígonos regulares, ângulos, grandezas e medidas).

## CONSIDERAÇÕES FINAIS

Nesta experiência de formação os professores envolveram-se com muita motivação e entusiasmo na realização destas tarefas. Experimentaram e identificaram o papel da criação e análise de projetos no *Scratch* com vista ao desenvolvimento de capacidades transversais e de conhecimentos matemáticos, alguns subjacentes aos vários projetos e outros desenvolvidos com mais profundidade por um projeto específico. Nestas situações e noutras que o professor de Matemática pode propor aos seus alunos com recurso ao *Scratch*, deve promover a capacidade de análise e fomentar a discussão de modo a emergirem os conceitos matemáticos que estão subjacentes a comandos específicos ou a projetos, nomeadamente no que respeita a conceitos de geometria e medida, como aponta Calder (2010).

Não tendo estes formandos prática na utilização do *Scratch*, esta experiência evidenciou a sua grande vontade de saber mais sobre tecnologia e sobre a sua utilização para a aprendizagem dos alunos, devendo por isso a formação de professores proporcionar-lhes desafios que correspondam às suas expectativas, como se verificou neste caso.

### Referências

Balanskat, A., & Engelhardt, K. (2014). *Computing our future: Computer programming and coding - Priorities, school*

*curricula and initiatives across Europe*. Bruxelas: European Schoolnet.

Calao, L., Moreno-León, J., Correa, H., & Robles, G. (2015). Developing Mathematical Thinking with Scratch. *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.

Kordaki, M. (2012). Diverse categories of programming learning activities could be performed within Scratch. *Procedia – Social and Behavioral Sciences*, 46, 1162-1166.

Monroy-Hernández, A., & Resnick, M. (2008). Empowering kids to create and share programmable media. *Interactions*, 50-53.

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., ... Kafai, Y. (2009). *Communications of the ACM*, 52(11), 60-67.

Tinoca, L. (2014). EduScratch: a case study on the development of key competences in Europe. *European Journal of Curriculum Studies*, 1(1), 23-32.

**Raquel Santos**

ESCOLA SUPERIOR DE EDUCAÇÃO DE SANTARÉM

**Neusa Branco**

ESCOLA SUPERIOR DE EDUCAÇÃO DE SANTARÉM

UIDEF, INSTITUTO DE EDUCAÇÃO, UNIVERSIDADE DE LISBOA