Aprender a programar ou programar para aprender matemática?

João Torres

Miguel Figueiredo

Teresa Martinho Marques

Introdução

Em 1967 Seymour Papert criou, no Massachusetts Institute of Technology (MIT) a linguagem de programação *LOGO* que tinha como principal objetivo ajudar os alunos a aprender matemática e, em particular, geometria. Mais do que ensinar alunos a programar, Papert pretendia que o aluno aprendesse enquanto programava. Sessenta anos depois, as linguagens de programação voltam a estar na agenda dos sistemas educativos. Países como o Reino Unido ou a França integraram o pensamento computacional nos seus currículos e em Portugal está em curso, desde o ano letivo 2015/2016, um projeto piloto de Iniciação à Programação no Primeiro Ciclo do Ensino Básico que envolve mais de metade de todos os agrupamentos do país.

Em 2007, também no MIT, Mitchel Resnick, diretor do Lifelong Kindergarten Group, desenvolve uma nova linguagem de programação com finalidades educativas: o *Scratch*. Inspirando-se nas peças de *Lego*, foi criada uma linguagem que permite programar arrastando blocos de comandos que se encaixam uns nos outros. Uma ideia poderosa que simplifica a escrita dos programas, evitando erros de sintaxe e tornando mais fácil a compreensão dos programas que ficam representados graficamente, facilitando a sua leitura e a aprendizagem de conceitos de programação, nomeadamente os ciclos de repetição ou as estruturas de decisão.

O *Scratch* e matemática

Ao contrário do *LOGO*, o *Scratch* não foi desenvolvido tendo como principal finalidade o processo de aprendizagem da matemática. Mais abrangente nas suas finalidades, encontramos referências ao desenvolvimento da criatividade e de competências digitais essenciais no século XXI como (i) Competências de Informação e Comunicação, (ii) Competências de Raciocínio e Resolução de Problemas e Competências Interpessoais e de autodirecionamento (Natalie Rusk & Maloney, sd). Mas o raciocínio, a resolução de problemas e as competências de comunicação também se cruzam certamente com as grandes finalidades do ensino da Matemática e é por isso sem espanto que encontramos o *Scratch* entre os programas informáticos referidos nas Orientações de gestão curricular para o Programa e Metas Curriculares de Matemática para o Ensino Básico dos 1.º ao 9.º anos de escolaridade, recentemente disponibilizadas pela DGE, e nestas se encontre uma referência explícita ao uso do *Scratch*:

> . . . o *Scratch*, que, para além de uma iniciação a uma linguagem de programação, consequentemente envolve o pensamento lógico matemático, a estimação, coordenadas em referencial e variáveis (. . .) (DGE, 2016, p. 4)

Não sendo um programa concebido especificamente para o ensino da Matemática, como é o caso dos programas de geometria dinâmica, como o *Geogebra* ou o *Geometer's Sketchpad*, por exemplo, o *Scratch*, e as linguagens de programação de um modo geral, tem, de facto, potencialidades que podem e devem ser aproveitadas nesse processo. Não é possível programar sem saber matemática e programar pode também ser uma excelente oportunidade para desenvolver e consolidar aprendizagens nesta disciplina.

Vejamos algumas das ideias poderosas que os alunos podem desenvolver enquanto programam.

Referencial Cartesiano

Programar em *Scratch* passa por definir uma sequência de ordens a cada um dos "atores" (sprites na versão inglesa). Esses atores vão posicionar-se e movimentar-se num "Palco" com 480x360 pixels. Embora seja possível contornar a inserção manual das coordenadas de cada ator em cada

momento, os alunos cedo percebem que é mais preciso e cómodo indicar as coordenadas manualmente. Os alunos trabalharão com o referencial cartesiano por necessidade e terão de compreender o significado dos valores estabelecidos para as coordenadas x e y.

Figura 1. Exemplo de programa que posiciona um ator no palco quando a tecla espaço é pressionada

Se, por exemplo, desejarmos que um objeto se desloque de cima para baixo no ecrã, desde a posição y=180 até à posição y=-180, poderemos usar o código representado na figura 2. Claro que depois de executar o programa alguns problemas se poderão colocar. Com que velocidade desce o objeto? É adequada ao projeto que se está a programar?

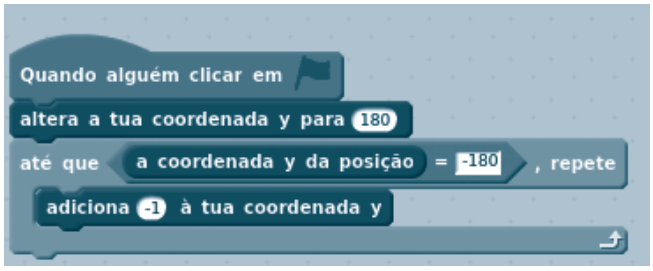

Figura 2. Programar um objeto para "cair" no ecrã

Como se pode aumentar ou diminuir essa velocidade? Todas estas questões são passíveis de ser colocadas pelos próprios alunos. Resolvem-se com matemática e, na maioria das vezes, os próprios alunos colocam-nas e respondem por iniciativa própria.

Números negativos e números decimais

No exemplo representado na figura 2 utilizamos números negativos. A primeira vez para estabelecer o ponto onde o objeto deixaria de descer e a segunda quando adicionamos ´-1´ à posição do objeto. Na realidade não existe em *Scratch* o comando para subtrair um valor à coordenada y (nem x) de um objeto. A solução é adicionar um número negativo. Os números negativos surgem assim naturalmente, para resolver problemas de programação. Da mesma forma, a utilização de números na representação decimal pode surgir como uma necessidade do próprio aluno. Tomemos, mais uma vez, um exemplo. As personagens, ou atores, no *Scratch* podem tomar várias formas ao longo de um programa 'vestindo' vários trajes. Assim, o morcego representado na figura 3 pode, em cada momento, aparecer com o traje 'bat1-a' ou 'bat1-b'. Se alternarmos a sua representação entre cada um destes trajes teremos a sensação de que bate as asas.

Figura 3. Dois trajes para o morcego

Se no interior de um ciclo infinito colocarmos um comando que permita ao morcego ir alterando o seu traje como representado no código da figura 4 teremos o efeito pretendido... ou quase!

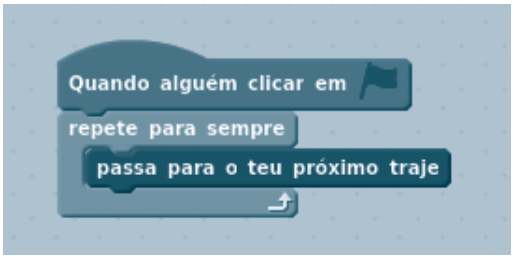

Figura 4. Código para o morcego bater as asas

Na verdade, devido à velocidade de processamento dos computadores, a alteração do traje será tão rápida que não temos a ilusão do bater de asas. Uma das grandes potencialidades deste *software* é a facilidade e rapidez com que podemos testar os nossos programas. Isso permite ao aluno testar e corrigir num espaço muito curto de tempo as suas conjeturas. Coloquemos dentro do ciclo o comando 'espera' que vem preenchido com o argumento '1' a que corresponde uma espera de um segundo, como representado na figura 5.

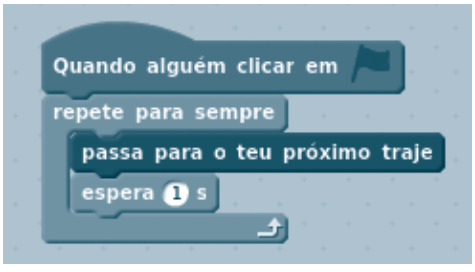

Figura 5. Batimento de asas com o comando 'Espera'

Voltando a testar o programa, o aluno perceberia que agora o morcego bate as asas muito lentamente uma vez que a espera de um segundo é demasiado longa para o efeito pretendido. Alunos do primeiro ciclo, mesmo antes da introdução dos números decimais sentem assim necessidade de representar e usar números compreendidos entre 0 e 1, uma vez que a espera de 0 equivale a retirar este comando e torna o batimento muito rápido e, por outro lado, utilização da espera de 1 segundo torna o batimento demasiado longo. Neste caso, uma espera de 0,2 segundos torna o batimento muito mais realista.

Geometria, variáveis e muito mais

Embora não sendo concebido especificamente para a aprendizagem da matemática, o *Scratch* mantém algumas das características do *LOGO* e possui um conjunto de comandos que permitem às suas personagens deixar um rasto, podendo assim desenhar à medida que se movimentam, tal como acontecia no *LOGO*. Se nos concentrarmos na lista de comandos destacados no retângulo na figura 6, veremos que a nossa personagem começa por "andar" 60 passos, para depois girar 90° e depois fazer uma pequena espera de 1 segundo. Seguidamente, repete 4 vezes estes comandos, desenhando assim um quadrado com 60 passos de lado.

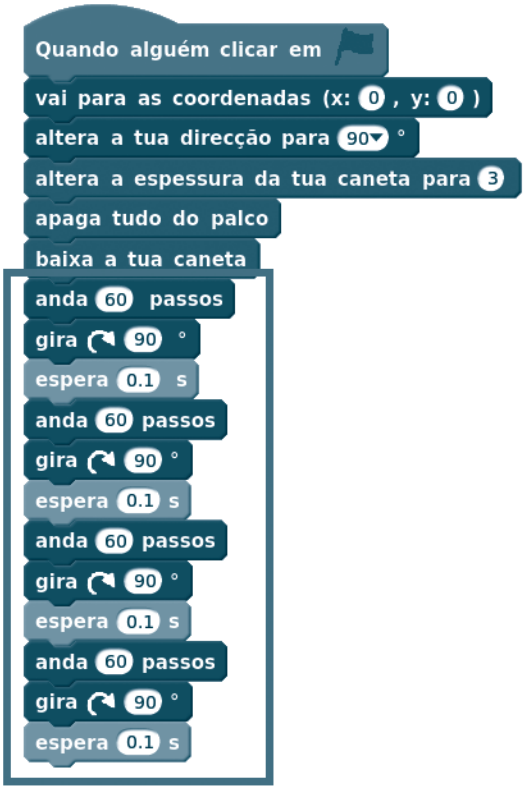

Figura 6. Procedimento para desenhar um quadrado em *Scratch*

Este seria, provavelmente, o conjunto de procedimentos que uma criança do primeiro ciclo utilizaria para programar uma personagem *Scratch* para que desenhasse um quadrado. A matemática está presente, pois só o conseguiria fazer se soubesse as propriedades do quadrado, evidentemente. Contudo, ao analisar o código, podemos verificar que há um padrão que se repete. Poderia, então, o mesmo quadrado ser desenhado utilizando o código que se representa na figura 7.

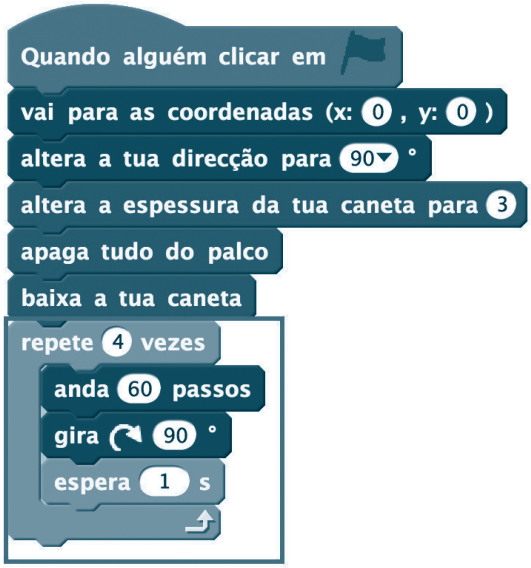

Figura 7. Procedimento para desenhar um quadrado em *Scratch*, utilizando ciclos

Para um programador, o código utilizado na resolução apresentada na figura 7 é mais "elegante" que o utilizado anteriormente, uma vez que é mais curto e simples de ler. Ao nível matemático, também passamos de uma lógica aditiva, em que dissemos à nossa personagem para fazer uma sequência de comandos, para passarmos para uma lógica multiplicativa, em que passamos a dizer que repita 4 vezes uma sequência mais curta, obtendo-se o mesmo resultado final.

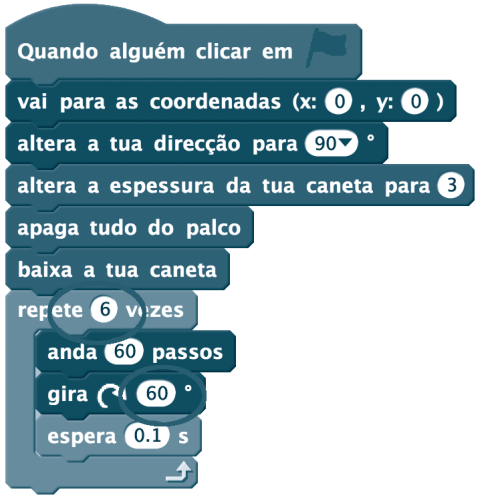

Figura 8. Procedimento para desenhar um hexágono em *Scratch*

Estamos, portanto, perante conceitos matemáticos e o aluno que passe da primeira resolução para a segunda está, certamente, a pensar como um matemático, compreendendo a relação matemática existente entre a multiplicação e a adição de parcelas idênticas.

Modificando apenas dois dos números do programa obteríamos, por exemplo, um hexágono, como se demonstra na figura 8. Facilmente percebemos que teremos de mudar o número de repetições do ciclo, uma vez que o hexágono tem 6 lados e o ângulo que a nossa personagem roda em cada vértice (ângulo externo, suplementar do ângulo interno – conteúdos matemáticos) também é diferente, passando, no caso do hexágono a ser de 60 graus. O aluno poderia agora ser desafiado a encontrar outros pares de valores que permitissem obter outros polígonos regulares e poderia chegar a 3 –> 120 para o triângulo, 5 –> 72 para o pentágono, etc... A procura de regularidades é uma tarefa cognitiva importante no raciocínio matemático.

Quando o aluno compreender que existe uma relação entre estes dois números poderá, utilizando variáveis, passar para uma fase seguinte, onde poderá criar um programa que permita desenhar vários polígonos regulares, bastando, para isso, alterar o valor da variável que controla o número de lados (ver figura 9).

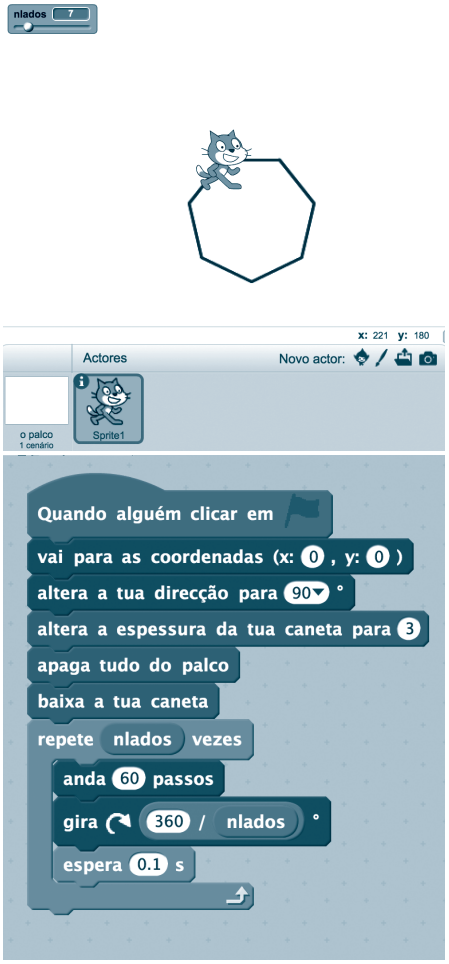

Figura 9. Generalização do desenho de polígonos usando variáveis

Numa conferência TED[1], a determinada altura Mitchel Resnick conta a história de uma visita que fez a alunos que utilizavam o *Scratch*. Um aluno tinha criado um jogo onde um peixe grande, controlado pelo utilizador, comia peixes mais pequenos, que se moviam no ecrã. O aluno queria saber quantos peixes tinham sido comidos e pediu ajuda ao investigador. Mitchel explicou-lhe que o poderia fazer criando uma variável que seria incrementada cada vez que um peixe era comido. Com alguma ajuda, passado poucos minutos, o aluno criou uma variável e a contagem começou a aparecer no ecrã. No final, agradeceu ao investigador dizendo-lhe: "Obrigado, obrigado, obrigado! Obrigado por me ensinar o que são variáveis". Resnick refere muitas vezes em encontros que serão poucas as ferramentas e contextos de trabalho em que os alunos desejam aprender conceitos matemáticos e agradecem quando estes lhes são ensinados, permitindo-lhes avançar na complexidade dos seus projetos.

Criar programas relacionados com a aprendizagem da matemática

A matemática estará presente na maioria dos projetos dos alunos, mas pode também ser o tema dos próprios projetos. No exemplo representado na figura 10, uma aluna do 5.º ano criou um jogo onde o jogador é desafiado a controlar um cavalo com as setas do teclado para tocar em todas as bolas que representem múltiplos de sete. Um contador, à esquerda no cimo do ecrã, será incrementado com 1 unidade por cada múltiplo encontrado. Além de toda a matemática envolvida, por exemplo para movimentar o cavalo, neste caso, o aluno teve que encontrar os múltiplos que queria representar. Neste caso concreto, por exemplo, o jogador tinha apenas 45 segundos para encontrar os múltiplos e, mais uma vez, a matemática estava presente.

Figura 10. *Projeto* Múltiplos de Sete

No exemplo representado na figura 11, outra aluna (6.º ano) criou um jogo, que designou por "Fração intrusa" em que o jogador terá que clicar sobre as frações e, em cada uma delas, ouvirá uma gravação que lhe indicará se a referida fração é a que quer descobrir: equivalente a quatro quintos. Mais uma vez, a aluna teve que mobilizar conteúdos matemáticos para conceber o jogo.

Desafiar os alunos para criarem jogos ou aplicações relacionados com os conteúdos da disciplina de matemática pode ser uma maneira de garantir que os utilizam. No entanto, a matemática estará, quase com toda a certeza, presente em todos os projetos que realizarem em *Scratch*.

Teresa Marques (2009) realizou um estudo em que utilizou o *Scratch* com uma turma de 5.º ano durante um ano letivo. Reproduzimos de seguida algumas das competências matemáticas trabalhadas com os alunos ao longo desse ano letivo que sintetizou num quadro apresentado na página 135 do seu trabalho (Tabela 1)

Figura 11. *Projeto* Fração Intrusa

Tabela 1. Síntese de conceitos e competências de programação explorados no *Scratch* e competências e conceitos matemáticos trabalhados na Turma X (adaptado de Marques (2009))

Competências para resolução de problemas e para a concepção de projectos • raciocínio lógico • decomposição de problemas complexos em partes mais simples • identificação e eliminação de erros • desenvolvimento de ideias, desde a concepção até à concretização do projecto • concentração e perseverança **Noções básicas sobre computadores e programação** • os programas indicam ao computador exactamente o que deve ser feito, passo por passo • criar programas para computador em Scratch não exige perícia especial, apenas raciocínio claro e cuidadoso **Conceitos específicos de programação:** Sequência, iteração (ciclos), instruções condicionais, variáveis, execução paralela, sincronização, interacção em tempo real, lógica boleana, números aleatórios, gestão de eventos, desenho de interface do utilizador, estruturas de dados.
Sequência - Para criar um programa *Scratch* é preciso pensar de forma sistemática na ordem de execução das instruções. No estabelecimento de prioridades de cálculo (expressões numéricas), ou de passos de resolução de um problema, o entendimento desta ideia de prioridade trabalhado com as sequências é fundamental.
Iteração (ciclos) e também sequências – repete e para sempre podem usar-se para repetir um bloco de instruções. Operação multiplicação (adição de parcelas idênticas). Todas as operações - **Números e cálculo** (primeira abordagem aos números racionais e ao conceito de percentagem – comandos que permitem redução e ampliação), **Medida, Geometria e Álgebra** (uma vez que a sequência de instruções combinadas que se repete nos ciclos pode ser de qualquer natureza ou domínio matemático). Alguns aspectos trabalhados pelos alunos: movimento obtido com recurso à variação das coordenadas (posição de um ponto no plano); movimento linear obtido com recurso a um valor inteiro ou fraccionário, ângulos internos e externos, ângulos suplementares; combinação dos ângulos com movimento linear para obtenção de "linhas curvas"); tempo (valor inteiro ou fraccionário), velocidade, direcção, sentido...

Gestão de eventos e desenho de interface do utilizador – resposta a eventos despoletados pelo utilizador ou por outra secção de um programa e desenho de interfaces de utilizador interactivas – por exemplo usando *sprites* clicáveis para criar botões. Mesmo projectos que não ilustram directamente conceitos matemáticos usam-nos. O controlo de gráficos e eventos pode ser uma actividade matemática intensa que envolve conceitos diversos, dependendo do contexto e da situação. Para a construção de histórias e diálogos, por exemplo, os alunos necessitam de controlar os tempos e momentos de intervenção fazendo, por sua iniciativa, os cálculos necessários. Para conceber alguns cenários propostos foi necessário recorrer às coordenadas no plano e descobrir modos de garantir o rigor de determinadas distâncias.

Conclusão

Mais do que um programa concebido para a aprendizagem da matemática, o *Scratch* apela à criatividade dos alunos fornecendo um ambiente rico onde podem utilizar ideias poderosas, testando-as e obtendo imediatamente os resultados para as suas conjeturas, desenvolvendo também competências de resolução de problemas e a autonomia e persistência necessárias para enfrentar os desafios colocados pela programação (transferindo essas competências para outras situações). O aluno perceberá ainda que para programar têm que ser mobilizados conceitos matemáticos, dando sentido à matemática que aprende.

Se programar é uma competência fundamental no séc. XXI, a matemática tem um papel imprescindível para que o aluno consiga programar. Programar pode ser assim um ótimo pretexto para aprender matemática criando necessidade de a compreender como já há muito Papert (1997) alertava:

> Será que estamos mesmo à espera que as crianças se mantenham passivas perante os currículos pré-digeridos do ensino básico, quando já exploraram o saber contido nas auto-estradas da informação de todo o mundo e se abalançaram a realizar projetos complexos, procurando por si próprias o conhecimento e os conselhos de que necessitaram para os pôr em prática? (Papert, 1997, p. 226)

Nota

[1] http://www.ted.com/talks/mitch_resnick_let_s_teach_kids_to_code?language=pt

Referências

DGE (2016, Agosto). Orientações de gestão curricular para o programa e metas curriculares de matemática ensino básico dos 1.º ao 9.º anos de escolaridade. Acedido em: http://www.dge.mec.pt/ sites/default/files/Basico/Metas/Matematica/ documento_orientador-_ensino_basico.pdf (Acedido em setembro de 2016)

Natalie Rusk, M. R., & Maloney, J. (sd). Competências de aprendizagem para o Séc. XXI. Lifelong Kindergarten Group MIT Media Laboratory. Acedido em: http://projectos.ese.ips.pt/eduScratch/index.php/recursos2/finish/34-textos/370-competencias-de-aprendizagem-para-o-sec-xxi-documento-mit/0 (Tradução para português de Teresa Marques, Acedido em março de 2016)

Papert, S. (1997). A família em rede. Lisboa: Relógio D'Água.

Marques, M. T. P. M. (2009). Recuperar o engenho a partir da necessidade, com recurso às tecnologias educativas: Contributo do ambiente gráfico de programação Scratch em contexto formal de aprendizagem. (Tese de Mestrado). Faculdade de Psicologia e de Ciências da Educação da Universidade de Lisboa.

João Torres
Escola Superior de Educação/Instituto Politécnico de Setúbal
Miguel Figueiredo
Escola Superior de Educação/Instituto Politécnico de Setúbal
Teresa Martinho Marques
Agrupamento Vertical de Escolas de Azeitão