MEDI@ÇÕES
Revista OnLine

IPS Instituto Politécnico de Setúbal
Escola Superior de Educação

Programação tangível e a promoção do Pensamento Computacional: propostas didáticas desenvolvidas no projeto TangIn

FILIPE MOREIRA, ISABEL CABRITA, MARIA JOSÉ LOUREIRO, CECÍLIA GUERRA

filipertmoreira@ua.pt**, icabrita@ua.pt*, zzelouro@gmail.com*, cguerra@ua.pt*

*Centro de Investigação Didática e Tecnologia na Formação de Formadores, Departamento de Educação e Psicologia, Universidade de Aveiro

** DigiMedia – Digital Media and Interaction, Departamento de Comunicação e Arte, Universidade de Aveiro

Resumo

A programação é uma realidade em muitas escolas, para desenvolver o pensamento computacional, promover a inclusão e sensibilizar os alunos para as áreas STEM. Para isso, várias ferramentas têm sido experimentadas, como robôs programados de forma tangível. Neste âmbito, surgiu o projeto TangIn, tendo-se (co)desenvolvido uma *toolbox* para apoiar a programação tangível em contexto educativo. Neste artigo, apresenta-se uma breve resenha sobre o ensino da programação, define-se programação tangível e apresentam-se propostas didáticas para a promoção do pensamento computacional através da abordagem STEM (co)desenvolvidas no âmbito do projeto TangIn.

Palavras-chave:

Programação Tangível; Pensamento Computacional; Robótica; Propostas didáticas; Ensino Básico; STEM

Abstract

Programming is a reality in many schools, to develop computational thinking, promote inclusion and raising pupils' awareness to STEM areas. For this, several tools have been experimented, like robots programmed in a tangible way. TangIn project emerged in this context. During the project a toolbox have been (co)developed to support tangible programming in an educative context. In this article, a brief review on the programming teaching is presented, tangible programming is defined and didactic proposals for the promotion of computational thinking through the STEM approach are presented (co)developed during TangIn project.

Key concepts:

Tangible programming; Computational Thinking; Robotics; Didactical proposals; Basic Education; STEM

Introdução

O domínio de tecnologias é crucial para que as crianças estejam preparadas para os desafios constantes da evolução científica e tecnológica. Acresce que, no quadro complexo e em permanente transformação da sociedade contemporânea, alguns autores têm evidenciado que as tecnologias podem facilitar a inclusão e a aproximação dos alunos às áreas Science, Technology, Engineering e Mathematics (STEM), Ciência, Tecnologia, Engenharia e Matemática (STEM), em português (Kennedy & Odell, 2014).

Não admira, portanto, que o desenvolvimento de competências digitais faça parte das competências essenciais do século XXI, devendo ser considerado como um objetivo educativo universal (Kloos et al., 2018) e desde os primeiros anos de escolaridade.

Em particular, defende-se o desenvolvimento do "pensamento computacional" dos alunos, que a programação, quer seja gráfica quer seja tangível, pode proporcionar. De facto, a programação é uma excelente oportunidade para se compreender como se pode processar a simplificação de processos e a resolução de problemas em qualquer contexto da vida, quer ao nível pessoal, quer profissional, dimensões indissociáveis do referido pensamento.

Na Europa e, designadamente, em Portugal, aprender a programar é uma necessidade educativa bem identificada, dada a carência de profissionais nesta área, tendo-se estimado, já em 2015, que em 2020 existiria um défice de 800 mil profissionais da área da computação/informática (Balanskat & Engelhardt, 2015). Aprender a programar afigura-se, assim, como uma premissa, uma indústria, uma política, uma promessa e um valor (Strawhacker & Bers, 2015).

As primeiras experiências de utilização da programação na educação remetem-nos para a década de 60 do século XX, nomeadamente, através da utilização da linguagem LOGO, desenvolvida no Massachusetts Institute of Technology (MIT) por Seymour Papert e colaboradores (Papert, 1980). A investigação sobre o potencial educativo da programação conduziu a uma mudança de paradigma, passando a defender-se o 'construcionismo', baseado no conceito do construtivismo mas tendo as tecnologias um lugar privilegiado enquanto mediadoras da construção do conhecimento (Clements & Gullo, 1984; Anchieta Silveira, 2016).

No contexto português, as primeiras experiências de ensino da programação ocorreram na década de 80, fortemente potenciadas pelo projeto Minerva (1985-1994) (Ponte, 1994).

O ensino de programação é, hoje em dia, uma realidade em muitas escolas em toda a Europa. Vários países europeus têm procurado avançar com iniciativas educativas que estabelecem "a introdução da programação de computadores nos primeiros anos de escolaridade. Em 2014, o Reino Unido avançou com o anúncio da introdução como área curricular nos primeiros anos de escolaridade. Mais tarde, em Portugal surge o projeto "Iniciação à Programação no 1º CEB" e que teve como linhas orientadoras o trabalho desenvolvido no projeto EduScratch (Figueiredo & Torres, 2015).

Decorrente desta realidade, vários autores têm relevado o contributo do ensino da programação para o desenvolvimento do pensamento computacional dos alunos, conduzindo projetos de investigação & desenvolvimento (I&D) centrados na conceção, implementação e avaliação de atividades didáticas focadas no potencial educativo da aprendizagem de linguagens de programação, particularmente da programação de materiais tangíveis (ex. utilização de robôs educativos) para o desenvolvimento de competências transversais dos alunos, como o pensamento computacional.

O projeto TangIn "Tangible programming & inclusion", que se propôs (co)desenvolver uma *toolbox* que integra um conjunto de guiões e recursos para o professor e respetivos alunos, com tarefas orientadas para o desenvolvimento do pensamento computacional e da criatividade nos alunos, sensibilizando-os para as áreas STEM (Guerra et al., 2020).

Neste artigo, enfatiza-se a importância de se continuar a investir na disseminação de recursos educativos de qualidade que incentivem ao ensino da programação tangível enquanto mediadora da inclusão e da necessária fusão das áreas STEM.

Importa, então, apresentar as potencialidades educativas da *toolbox* do TangIn, desenvolvidas numa lógica de promoção da inclusão e integração das áreas STEM nos primeiros anos de escolaridade, fortalecendo-se a investigação nesta área específica, para haver um maior entendimento das vantagens e constrangimentos da programação quer em contexto formais quer não formais de aprendizagem.

1. Enquadramento teórico

1.1 Pensamento computacional

O "pensamento computacional" é considerado como uma das competências essenciais que os alunos devem desenvolver, a par da competência de leitura e de escrita e de realização de operações aritméticas (P21's Framework for 21st Century Learning, 2015). O

"pensamento computacional" é promovido a partir da análise das capacidades e limitações dos processos de tratamento de informação, quer estes sejam executados por computadores, quer sejam executados por humanos (Wing, 2012). De acordo com Figueiredo & Torres (2015), o "pensamento computacional" não é inerente ao trabalho com computadores embora, muitas vezes, se estabeleça essa associação quase instintivamente.

Há diferentes conceções de "pensamento computacional", mas a mais frequente prende-se com o "solucionar um puzzle", de modo a reconhecer padrões e desconstruir o problema em partes mais pequenas e simples, de forma lógica e sequencial (Loureiro et al., 2020, p.3).

Para Brennan, Balch e Chung (2014), citados em Figueiredo & Torres (2015), existem conceitos e práticas relacionadas com o "pensamento computacional" que podem ser potenciados quando o aluno programa através de um recurso tecnológico (quer seja digital, quer seja tangível). Na Tabela 1, apresenta-se uma descrição dos conceitos de programação, que têm níveis de complexidade crescente, à luz do significado do ensino da programação nos primeiros anos de escolaridade (Figueiredo & Torres, 2015).

Tabela 1 – Apresentação e descrição de conceitos de programação

Sequências	Sempre que executamos uma série de comandos em programação, eles são interpretados sequencialmente. A ordem pela qual aparecem é importante. Muitas vezes basta trocar a ordem de dois elementos para obtermos resultados completamente diferentes.
Ciclos	A mesma sequência pode ser executada várias vezes. Depois de criar programas com sequências de comandos, os alunos reconhecerão padrões de repetição. A utilização de ciclos, sendo mais exigente em termos de pensamento computacional do que uma simples sequência, tornará os programas mais pequenos, mais legíveis e fáceis de compreender.
Eventos	Acontecimentos que desencadeiam uma determinada ação.
Condições	A programação nem sempre é linear. De acordo com determinadas condições e mediante a utilização de estruturas de decisão, o programa poderá tomar diferentes rumos.
Operadores	Os alunos utilizarão operadores para realizarem operações matemáticas e/ou lógicas.
Dados/ Variáveis	Em programação será necessário armazenar, recuperar e atualizar valores que serãoguardados em variáveis.
Execução em paralelo	Quando executamos um programa, muitas vezes várias ações iniciam-se em paralelo. Será necessário compreender este conceito e programar de modo a que os eventos aconteçam quando necessário e previsto.

O reconhecimento de padrões, abstração e orientação especial, em que se tem de transpor a realidade de um objeto para nós próprios (a esquerda e a direita do robô pode ser diferente da nossa), são capacidades que podem ser usadas não só como conceitos introdutórios da programação/código, mas também como método de pensamento e de resolução de problemas, que podem ser aplicados, virtualmente, em qualquer área e nível. A Tabela 2 apresenta as práticas relacionadas com o ato de programação (adaptado de Figueiredo & Torres, 2015).

Tabela 2 – Práticas relacionadas com o ato de programação

Ação iterativa e incremental	Um projeto de programação é desenvolvido por etapas. Apenas quando parte do projeto funciona corretamente, se avança para o desenvolvimento das etapas seguintes, que, muitas vezes, também poderão ser testadas isoladamente.
Teste e depuração	Depois de concluir um programa (ou uma etapa), é necessário testar e certificar-se de que tudo funciona como estava previsto. Muitas vezes, nesta fase, e na partilha de projetos, são encontrados erros que tinham passado despercebidos ao longo do processo de construção do programa e que deverão ser corrigidos.
Decomposição e abstração	Problemas complexos podem ser divididos em problemas mais simples. Por exemplo, se pretendermos desenhar vários quadrados, é possível criar um programa para desenhar um. Desenhar os restantes será simples, reformulando ou repetindo o programa anterior. Mas, se tivermos de desenhar vários polígonos regulares, podemos tentar desenhar um quadrado, um pentágono e um hexágono regulares e depois generalizar, obtendo um programa que permita desenhar um qualquer polígono regular, indicando o número de lados.
Reutilização e reformulação	Os projetos podem ser construídos partindo de outros, sendo apenas reformulados ou adaptados. Pode ainda haver partes do projeto que possam ser programadas como funções ou procedimentos que se repetem ao longo da programação, construindo projetos grandes a partir de partes mais pequenas.

1.2. Programação tangível

As interfaces tangíveis de programação são um conjunto de objetos físicos interligados, cuja manipulação tem impacto direto em ambientes digitais (Bers, 2014; Farr et al., 2010; Loureiro et al., 2018; Wang et al., 2014; Zuckerman et al., 2005). Ou seja, na programação tangível, o criador do programa organiza os objetos físicos de forma a obter o "output" pretendido por parte do robô ou do computador, no caso em que os blocos manipuláveis são associados à aplicação ou *software* do dispositivo (computador, *tablet*, *smartphone*, etc.), que cumprirá os comandos programados, através desses blocos.

Alguns autores têm apresentado várias vantagens do ensino da programação tangível, quando comparada com linguagens gráficas. Convém explicar que, quando se fala de linguagens gráficas, está a falar-se de código, isto é, as várias linguagens de programação por objetos gráficos (*Scratch, Kodu, Blockly, etc*) utilizadas em computador ou dispositivos móveis. As imagens que se seguem ajudam a compreender a diferença destes dois ambientes (Figura 1 e Figura 2).

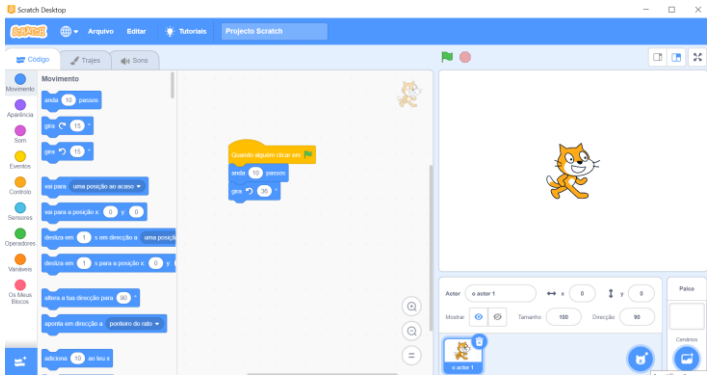

Figura 1 - Linguagem gráfica de programação – Scratch 3

Figura 2 – Blocos de programação tangível do robô MI-GO

A atenção dos investigadores pela área da programação tangível tem sido atraída pela premissa de que as tecnologias de índole tangível podem tornar a atividade de programação bastante mais acessível para as crianças (Nusen & Sipitakiat, 2011; Sapounidis et al., 2019; Sapounidis & Demetriadis, 2012). A programação tangível possibilita a aplicação de objetos físicos de programação numa atividade de aprendizagem atrativa e acessível para crianças mais novas e de modo mais direto e menos abstrato (Loureiro et al., 2020; Loureiro et al., 2018).

A generalidade dos projetos de ensino relacionados com a "introdução à programação" não têm como objetivo formar especialistas/técnicos de programação, mas contribuir para o desenvolvimento do pensamento computacional dos alunos (Wing, 2012).

De acordo com Ramos (2014), quanto mais cedo for iniciado, mais potencial oferece durante a "fase sensível" da formação dos jovens que ocorre até aos 15 anos.

De forma a fortalecer a abordagem STEM, a promoção da inclusão e um conhecimento holístico dos alunos, algumas das "Metas de Desenvolvimento Sustentável" das Nações Unidas que estão intimamente relacionadas com as dinâmicas da Programação Tangível e estas foram privilegiadas no projeto TangIn, a saber:

- assegurar educação de qualidade e inclusiva para todos e que promova aprendizagem ao longo da vida;
- conseguir igualdade de género nos participantes no projeto e empoderar todas as mulheres e raparigas;
- reduzir desigualdade dentro e entre países;
- gerir de forma sustentável as florestas e o combate à desertificação, parar e reverter a degradação das terras, parar a perda da biodiversidade.

O ensino da programação tangível nos primeiros anos de escolaridade é facilitado porque, nas faixas etárias mais jovens, a descoberta do mundo através do toque é de primordial importância na construção da aprendizagem, no conhecimento do mundo e na apropriação que deles fazem. Quanto a outras vantagens atribuídas à programação tangível, destacam-se alguns dos aspetos mais citados no estado da arte:

- facilita a programação colaborativa entre pares (McNerney, 2000; Strawhacker & Bers, 2015; Zuckerman et al., 2005);
- potencia processos de *debugging* (McNerney, 2000), ou seja, procedimentos que consistem em procurar, detetar e corrigir erros;
- ajuda a esbater as diferenças de género verificadas no interesse pelas áreas da computação (McNerney, 2000);
- promove o envolvimento sensorial, uma vez que as crianças aprendem havendo um incremento dos sentidos usados (toque, visão, audição) (Zuckerman et al. 2005; Falcão & Gomes, 2007).

2. Projeto TangIn

Neste quadro, surgiu o projeto TangIn - Tangible Programming and Inclusion (Loureiro et al., 2020). Uma equipa multidisciplinar internacional, com participantes da Bulgária, Espanha, Letónia e Portugal, desenvolveu (concebeu, criou, implementou, avaliou e reformulou) a *toolbox* TangIn[1], a qual contém materiais didáticos de apoio à atividade de professores e respetivos alunos.

Numa primeira fase do projeto TangIn, aplicou-se um questionário a professores dos países pertencentes ao consórcio (Portugal, Espanha, Bulgária e Letónia), cuja análise das respostas permitiu corroborar que os docentes assumiam que tinham pouco conhecimento sobre aspetos relacionados com o ensino da programação, robótica e estratégias que permitissem um uso adequado de tecnologias, nomeadamente tecnologias de programação tangível, na abordagem às áreas STEM (Loureiro et al., 2020).

Tal conclusão veio reforçar a necessidade de se desenvolver a *toolbox* TangIn, que integra conceitos e ferramentas de programação tangível, enquanto suporte de uma abordagem curricular STEM, para potenciar a inclusão e motivação para a aprendizagem de alunos dos 1.º e/ou 2.º Ciclos do Ensino Básico, e que se apresenta em detalhe no ponto seguinte.

2.1 *Toolbox* (caixa de ferramentas)

A toolbox é uma "caixa de ferramentas didáticas" integra uma diversidade de recursos para professores e alunos (do 1º e 2º ciclo), sendo composta, essencialmente, por um conjunto de 21 planos de aula concebidos seguindo uma abordagem STEM e que cobrem diversas áreas tais como a Matemática, Biologia, Geografia, etc., integradas, com recursos de programação tangível.

Os 21 planos de atividades foram concebidos para que, nem os professores nem os alunos tenham de possuir, à *priori,* competências digitais ou aceder a computadores para promover o ensino da programação tangível. Assim, os conceitos de programação são apresentados através da utilização de objetos físicos como o Mi-GO[2] e uma grelha quadricular onde o robô se movimenta, mediante a ação dos blocos de programação.

[1] Ver http://www.tangin.eu/wp-content/uploads/2019/11/EN_Tangin-Teachers-handbook.pdf e http://www.tangin.eu/lesson-plans-toolbox/

[2] Disponível em https://migobot.com

A estrutura de cada Plano de Aula é a seguinte:

- **Sumário**: resumo da atividade e/ou dos temas tratados no plano de aula.
- **Resultados de Aprendizagem**: competências que se espera que os alunos desenvolvam.
- **Ligações com tópicos curriculares**: cruzamento do principal conteúdo do plano de aula com tópicos do currículo.
- **Notas para os professores**: sugestões didáticas para os professores.
- **Plano de Aula**: descrição de cada momento da sessão (ex. contextualização, desenvolvimento, avaliação).
- **Lista de Recursos & Material de Apoio**: lista de recursos necessários para implementar a sessão, bem como material de apoio a imprimir (quando aplicável).

A capa de cada plano de aula expõe o nome da sessão, o leque de idades para a qual o plano é mais adequado e os principais temas tratados (Figura 3).

Figura 3 - Capa dos planos de aulas

Cada plano de aula serve como guião para os professores explorarem determinado tema, utilizando a programação tangível. A *toolbox* possui um manual para professores, isto é, um guia para conhecer os conceitos de programação tangível e de como utilizar os recursos educacionais. Os dois primeiros planos incluem tarefas introdutórios e o objetivo é: apresentar os conceitos de pensamento computacional, programação e robótica através de dinâmicas de role-play (01) e explicar como usar o robot MI-GO e os blocos de programação, incluindo as funcionalidades e características de cada bloco (02). Os

restantes 9 planos centram-se em atividades temáticas, em que cada uma: dirige-se a diferentes níveis de ensino e idades; abrange diferentes tópicos das disciplinas STEM e promove o uso de conceitos de programação tangível. Na Tabela 3 apresenta-se uma breve descrição dos 21 planos de atividades.

Tabela 3 – Planos de aula TangIn: atividades e idades dos alunos

Planos	Descrição	Idades
01_ Introdução à Programação	introdução ao pensamento computacional, programação e robótica, utilizando comandos e dinâmicas de jogo de papéis. simular inputs (entradas) e outputs (saídas) e prever resultados. Dar exemplos de programação e algoritmos na vida quotidiana.	6-12 anos
02_ Introdução ao MI-GO	introdução da história e funcionalidades do MI-GO. Nesta sessão, pretende-se que os alunos: personalizem o MI-GO, usando a sua criatividade e capacidades artísticas, e aprendam a programar o robô através da utilização de blocos e do entendimento da função de cada bloco. No final da sessão, os alunos serão capazes de executar instruções simples.	6-12 anos
03_ Caraterísticas de animais	uso de bilhetes de identidade de animais espalhados na grelha e uso do robô para percorrer determinados caminhos, consoante as características específicas de cada animal.	6-10 anos
04_ Quadrado Mágico	uso do robô para ir a todos as casas, sem repetir nenhuma, num tabuleiro de xadrez, usando apenas o movimento do cavalo, ao mesmo tempo que cria quadrados mágicos.	9-12 anos
05_ Mapas e sinais de trânsito	aprender que os mapas são representações da realidade numa escala diferente. Uso de coordenadas para encontrar a correta correspondência entre escalas mais pequenas e maiores. Identificar sinais de trânsito e compreender o seu significado.	6-10 anos
06_ Minecraft	uso do robô para identificar minerais numa sequência correta, de acordo com as suas propriedades.	8-12 anos
07_ Multiplicação	uso do robô para entender que a aprendizagem das tabelas de multiplicação não depende apenas da memorização, existindo uma lógica. Com a ajuda do MI-GO, pretende-se que os alunos construam tabelas de multiplicação através da contagem de quadrados (áreas), de uma forma divertida.	7-9 anos
08_ Reciclagem	aprender acerca dos diversos tipos de resíduos (lixo), onde devem ser depositados no ecoponto (Plástico, metal, vidro e restos) e sobre a política dos 3 R's Reduzir, Reutilizar e Reciclar)	6-10 anos
09_ Simetria	identificar, caso exista, o eixo de simetria de figuras geométricas desenhadas pelo MI-GO e dividi-las a meio.	7-10 anos
10_ Triângulos	localizar, classificar e agrupar diferentes triângulos em uma figura geométrica desenhada pelo MI-GO.	9-12 anos
11_ Patchwork	introdução à área puzzle. uso de Puzzle de áreas. Rodar figuras do tipo "Tetris" para ver onde se encaixam e pavimentar uma superfície. A resolução de alguns puzzles exige a colaboração entre diferentes grupos.	9-12 anos
12_ Sistema Circulatório	abordagem do sistema circulatório (SC), usando o MI-GO para, por exemplo, ligar os órgãos pertencentes ao SC, separar o sangue venoso do arterial.	7-9 anos

13_ Ângulos	introdução ao conceito de ângulo, usando o MI-GO para distinguir entre ângulo reto, ângulo agudo e ângulo obtuso.	6-10 anos
14_ Ligando Pontos	uso de puzzles para aprender/introduzir o conceito de ciclos, através da ligação entre diversos pontos, aumentando gradualmente o nível de complexidade.	8-12 anos
15_ Comboio espacial	abordagem do sistema solar (SS), usando o MI-GO para, por exemplo, desenhar circuitos e criar horários de comboios.	6-10 anos
16_ Ciclo da Água	abordagem do ciclo da água (CA), usando o MI-GO para, por exemplo, potenciar um jogo competitivo entre os alunos.	6-10 anos
17_ Palavras	introdução de uma atividade do tipo "Scrabble", usando o MI-GO para criar palavras a partir de um tema à escolha.	6-12 anos
18_ Constelações	desenhar constelações com o MI-GO usando cartas e mapas à escala para medir ângulos e comprimentos e converter às dimensões definidas.	9-12 anos
19_ Cálculos	emparelhar cartas com operações aritméticas básicas e números na grelha como soluções possíveis.	8-10 anos
20_ Países e Bandeiras	emparelhar cartas com bandeiras com países e capitais.	6-10 anos
21_ Unidades de medida	medir grandezas da sala de aula e usar os dados para criar mapas à escala.	8-10 anos

Os planos de aula não devem ser considerados como imutáveis e fechados, permitindo aos professores a liberdade de adaptar à dinâmica da sua aula e explorar o que considerarem mais relevante, tendo em conta o perfil dos alunos e objetivos de aprendizagem a atingir com as atividades didáticas (Tabela 4).

Tabela 4 – Matriz dos planos e tópicos curriculares

Tópicos	Planos da *toolbox* TangIn 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Pensamento computacional	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Algoritmos	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Itinerários	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Rotação	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Robótica	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Cálculos	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Raciocínio lógico	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Coordenadas					x																
Sinais de trânsito					x																
Minerais						x															
Áreas							x														
Recurso natural						x		x													
3 R's								x													
Reflexão e Simetria									x												
Triângulo										x											
Decomposição de figuras										x											
Proporção áurea (golden ratio)										x											
Pavimentação											x										x

Sistema circulatório humano											x								
Figuras geométricas regulares												x							
Ângulos internos e externos												x					x	x		
Regularidades e padrões													x							
Loop													x	x						
Velocidade														x						x
Horário														x						
Ciclo da água															x					
Constelações																	x			
Operações e propriedades																		x		
Expressão algébrica																		x		
Países europeus																			x	
Mapas																			x	

Reflexões finais / Conclusões

A utilização de robôs de programação tangível, como o MI-GO, permite que as crianças interajam, negoceiem entre elas e participem ativa e autonomamente na sua aprendizagem, enquanto "brincam para aprender e aprendem para brincar" (Zuckerman et al., 2005)

Tendo em conta estas premissas, durante o projeto TangIn foi possível contribuir para a (co)construção de 21 planos de aula cuja resolução apela à utilização de robôs de programação tangível, em particular o robô Mi-GO, de uma forma cativante, divertida, lúdica e colaborativa. A *toolbox* TangIn disponibiliza 21 planos de atividades de ensino e aprendizagem prontos a usar, com atividades atrativas e divertidas para os alunos, estimulando o seu interesse dos alunos pelas áreas STEM.

O projeto TangIn visou contribuir para um ensino da programação promotor do desenvolvimento de capacidades transversais dos alunos envolvidos nas atividades de exploração da *toolbox*, já referenciadas na UNESCO, e salientada por Figueiredo e Torres (2015), tais como:

- a comunicação/expressão através da partilha dos projetos desenvolvidos no grupo/turma e outros elementos da comunidade escolar, permitindo que os alunos expressem como construíram o programa e defendam as suas ideias;
- a colaboração dos alunos e o reconhecimento da vantagem de criar com e para os outros, explorando programas anteriormente construídos no projeto (e disponíveis na toolbox) como forma de inspiração para a criação de novos programas.

- o questionamento dos alunos sobre a forma como outros programas terão sido desenvolvidos, como foram solucionados determinados problemas, levando-os a questionar como funcionam diversas situações do mundo real.

O consórcio do projeto TangIn desenvolveu uma estratégia de disseminação da *toolbox* com o objetivo de promover o envolvimento de outros docentes, empresários e decisores políticos dos diferentes países envolvidos (Portugal, Bulgária, Letónia e Espanha) que não estiveram diretamente envolvidos no projeto. De acordo com esta premissa, realizaram-se quatro eventos multiplicadores (um em cada país envolvido), que visavam fornecer aos participantes ferramentas que lhes permitissem compreender a importância da articulação entre áreas curriculares distintas, da introdução à programação e da promoção da inclusão, para a formação de crianças competentes para enfrentar o século XXI (Guerra et al., 2020).

As perceções dos professores envolvidos neste projeto, sobretudo ao longo de vários eventos multiplicadores promovidos pelo consórcio do projeto TangIn nos vários países (Portugal, Espanha, Bulgária e Letónia) é que é, de facto, trata-se de um recurso educativo que potencia a introdução de conceitos de programação tangível eo desenvolvimento do pensamento computacional (Loureiro et al., 2020).

No entanto, urge continuar a investir na avaliação da qualidade educativa da integração deste recurso em contexto de sala de aula, sobretudo através da continuidade do projeto *follow-up TangIn*, com vista a verificar se estes planos potenciam a promoção do trabalho colaborativo e a comunicação entre os alunos e pode, em alguns casos, facilitar a inclusão dos alunos (ex. questões de género relacionadas com a heterogeneidade dos grupos).

Referências Bibliográficas

Anchieta Silveira, J. (2016). Construcionismo e inovação pedagógi-ca: uma visão crítica das concepções de Papert sobre o uso da

Balanskat, A., & Engelhardt, K. (2015). *Computing our future: Computer programming and coding - Priorities, school curricula and initiatives across Europe*. http://www.eun.org/pt/resources/detail?publicationID=661

Bers, M. U. (2014). Tangible Kindergarten: Learning How to Program Robots in Early Childhood. In *The go-go guide for engineering curricula, prek-5* (Cary I. Sn). Corwin. https://us.corwin.com/en-us/nam/the-go-to-guide-for-engineering-curricula-prek-5/book241521#contents

Brennan, K., Balch, C., & Chung, M. (2014). Creative computing: Scratch curriculum guide. *Harvard Graduate School of Education.* Retrieved from scratched. gse. harvard. edu/guide.

Clements, D. H., & Gullo, D. F. (1984). Effects of Computer Programming on Young Children's Cognition. *Journal of Educational Psychology*, *76*(6), 1051–1058.

Falcão, T. P., & Gomes, A. S. (2007). Interfaces Tangíveis para a Educação. *Simpósio Brasileiro de Informática Na Educação - SBIE*, *1*(1), 579–589. https://doi.org/10.5753/CBIE.SBIE.2007.579-589

Farr, W., Yuill, N., & Raffle, H. (2010). Social benefits of a tangible user interface for children with Autistic Spectrum Conditions. *Autism*. https://doi.org/10.1177/1362361310363280

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

Guerra, C., Moreira, F., Loureiro, M. J., & Cabrita, I. (2020). Programação tangível para a inclusão e promoção das stem- contributos para a formação contínua de professoreS. In *Revista APEduC* (Issue 01). https://unric.org/pt/17-objetivos-de-desenvolvimento-sustentavel-entram-em-vigor-a-1-de-janeiro/

Kennedy, T. J., & Odell, M. R. L. (2014). Engaging Students In STEM Education. *Science Education International*.

Kloos, C. D., Munoz-Merino, P. J., Alario-Hoyos, C., Estevez-Ayres, I., Ibanez, M. B., & Crespo-Garcia, R. M. (2018). The hybridization factor of technology in education. In *2018 IEEE Global Engineering Education Conference (EDUCON), Global Engineering Education Conference (EDUCON), 2018 IEEE* (pp. 1883–1889). https://doi.org/10.1109/EDUCON.2018.8363465

Loureiro, M. J. de M. N., Moreira, F. T. T., & Senos, S. (2018). *Introduction to Computational Thinking With MI-GO* (pp. 110–137). https://doi.org/10.4018/978-1-5225-5867-5.ch006

Loureiro, M. J., Guerra, C., Cabrita, I., Moreira, F. T., Gonçalves, D., & Queiroz, J. (2020). *Teachers' training handbook - tangible programming and inclusion in educational context*. UA Editora.

McNerney, T. S. (2000). Tangible Programming Bricks : An approach to making programming accessible to everyone. *Media, June 1983*.

Nusen, N., & Sipitakiat, A. (2011). Robo-Blocks: A Tangible Programming System with Debugging for Children.

Papert, S. (1980). Computers for children. In Mindstorms: Children, computers and powerful ideas.

Ponte, J. P. da. (1994). Introduzindo as NTI na Educação em Portugal DEPGEF.

Sapounidis, T., & Demetriadis, S. N. (2012). Exploring children preferences regarding tangible and graphical tools for introductory programming: Evaluating the PROTEAS kit. *Proceedings of the 12th IEEE International Conference on Advanced Learning Technologies, ICALT 2012*, 316–320. https://doi.org/10.1109/ICALT.2012.48

Sapounidis, T., Demetriadis, S., Papadopoulos, P. M., & Stamovlasis, D. (2019). Tangible and graphical programming with experienced children: A mixed methods analysis. *International Journal of Child-Computer Interaction*, *19*, 67–78. https://doi.org/10.1016/j.ijcci.2018.12.001

Strawhacker, A., & Bers, M. U. (2015). "I want my robot to look for food": Comparing Kindergartner's programming comprehension using tangible, graphic, and hybrid user interfaces. *International*

Journal of Technology and Design Education, *25*(3), 293–319. https://doi.org/10.1007/s10798-014-9287-7

Wang, D., Wang, T., & Liu, Z. (2014). A tangible programming tool for children to cultivate computational thinking. *The Scientific World Journal*, *2014*. https://doi.org/10.1155/2014/428080

Wing, J. M. (2012). *Computational Thinking*. https://www.microsoft.com/en-us/research/wp-content/uploads/2012/08/Jeannette_Wing.pdf

Zuckerman, O., Arida, S., & Resnick, M. (2005). *Extending tangible interfaces for education*. 859. https://doi.org/10.1145/1054972.1055093

Notas curriculares

Filipe T. Moreira é professor do 1.º e 2.º Ciclo do Ensino Básico, com ampla experiência na participação e desenvolvimento de projetos na área das TIC (nacionais e internacionais), principalmente relacionados com a introdução à programação e robótica nos primeiros anos de escolaridade. Destes projetos destaca-se: o PAprICA – Potenciar Aprendizagens com a Internet das Coisas que visa o desenvolvimento de recursos para a utilização de IoT em contextos educativos; e o TangIn - projeto que consistiu no desenvolvimento de recursos educativos para a utilização de robôs programados de forma tangível para a promoção de conhecimentos de CTEM. Atualmente é professor do 2.ºCEB no Ensino Público e também membro do DigiMedia – Digital Media and Interaction, Departamento de Comunicação e Arte, Universidade de Aveiro e colaborador ativo do Centro de Competência TIC da mesma Universidade.

Isabel Cabrita é Professora Associada no Departamento de Educação e Psicologia da Universidade de Aveiro (UA). É doutorada em Didática (Matemática e tecnologias digitais); membro do Centro de Investigação Didática e Tecnologia na Formação de formadores (CIDTFF); coordenadora do lem@tic – laboratório de Educação em matemática e do Centro de Competência TIC da UA (ccTIC-UA); diretora do Mestrado em Ensino de Matemática no 3.º CEB/Ensino Secundário e membro da direção do Programa Doutoral em Multimédia em Educação e do Mestrado em Educação e Formação; assessora Editorial da Revista Indagatio Didactica. Tem diversas publicações, principalmente, na área da educação em matemática e tecnologias digitais e coordenou(a)/participou(a) diversos projetos de I&D e programas de formação inicial, pós-graduada e contínua de professores. (http://orcid.org/0000-0003-0255-7577)

Maria José Loureiro é Doutorada em Didática, área de e-Learning (2007), mestre em Ciências da Educação, Tecnologia Educativa - Mestrado Europeu (1994), licenciada em Ensino de Francês/Português e docente do ES e 3.º CEB (1982 - 2002).
Exerceu funções de orientadora de Francês nos estágios integrados nas universidades de Aveiro e de Coimbra, na década de 90.
É formadora na área das TIC com registo de certificação pelo Conselho Científico Pedagógico de Formação Contínua (CCPFC), desde 1996.

Desempenha funções de professora requisitada no Centro de Competência TIC da ERTE/DGE, tendo também integrado polos e CC dos Projetos Minerva, Nónio século XXI e CRIE, na Universidade de Aveiro, onde ingressou como assistente convidada em 2001/02.
Integra grupos de investigação e comissões científicas de revistas, publicações da especialidade e congressos nacionais e internacionais. É membro do Comité Executivo do ICEM (Conselho Internacional de Educação para os Media).

Cecilia Guerra é investigadora contratada no Centro de Investigação em Didáctica e Tecnologia na Formação de Professores (CIDTFF) no Departamento de Educação e Psicologia (DEP) da Universidade de Aveiro (Portugal). É doutorada em Multimédia em Educação (2012), Mestre em Comunicação e Educação em Ciência (2007) e Licenciada em Ensino de Biologia e Geologia (2002). Participou em vários projectos de investigação e eventos científicos (com funções nos Comités de Organização e Científico). Tem várias publicações científicas nas áreas de Didáctica das Ciências, Tecnologia Educativa e Formação de Professores. É uma das autoras de duas Marcas Nacionais - "The Human being and natural resources" (Nº435547) and "SCoRE – Science Communication for Researchers in Education" (Nº MN622205 and Nº MN622206). Actualmente, está a realizar um estudo intitulado "Sustainability of educational innovations in higher education: insights for the political, institutional and research arenas".
https://orcid.org/0000-0002-2560-165X