

Da programação por blocos ao pensamento computacional: contributos do projeto Gen10s

JOÃO CARVALHO SOUSA¹, MANUEL MEIRINHOS², MARIA RAQUEL PATRÍCIO³, VÍTOR GONÇALVES⁴

jsergio@ipb.pt, ¹ESE/IPB, meirinhos@ipb.pt, ²CIEB, ESE/IPB, raquel@ipb.pt, ³CIEB, ESE/IPB, vg@ipb.pt, ⁴CIEB, ESE/IPB

Resumo

A presente investigação incide sobre a operacionalização da iniciativa Gen10s em quatro concelhos do distrito de Bragança, iniciativa essa que já abrangeu no referido distrito cerca de quatrocentos alunos do 2º Ciclo do Ensino Básico (10/11 anos de idade). O projeto Gen10s impulsiona a inclusão em contexto educativo de atividades de programação em Scratch, uma linguagem de programação visual, por blocos, amigável, que permite criar jogos, animações e histórias interativas. Este ambiente possibilita desenvolver nos alunos um conjunto de aptidões, tais como: competências digitais; competências de comunicação e sociais; competências relacionadas com as áreas disciplinares e não disciplinares, bem como o pensamento computacional, o raciocínio lógico, a resolução de problemas e a criatividade. Além de um breve enquadramento teórico e descrição sucinta (objetivos, escolas e alunos envolvidos, equipamentos e recursos humanos e materiais disponíveis), evidencia-se a análise realizada pelos professores Scratch de cada uma das equipas, e cria-se uma grelha analítica que permita analisar criticamente as produções dos alunos envolvidos.

Palavras-chave:

Programação por blocos, pensamento computacional, Projeto Gen10s, ensino precoce de programação

Abstract

The present paper focuses on the implementation of Project Gen10s in four counties of the district of Braganza, having involved up to the moment, about 400 students of the 5th year of 1st cycle of basic education (age 10/11). Project Gen10s fosters the inclusion of Scratch programming activities in the educational environment. Scratch is a friendly block, visual programming language allowing the creation of animations, games or interactive stories. This environment enables students to develop a set of skills such as: digital literacy skills; social and communicational skills; skills pertaining to curricular and non-curricular areas as well as computational thought, logical reasoning, problem solving and creativity. Apart from a brief theoretical framing and brief description of the experience) goals, schools and students involved, available material/equipment and human and technical resources), a special attention is paid to the analyses undertaken by the Scratch teachers of each team and an analytical grid allowing for the critical analysis of student's productions is proposed.

Key concepts:

Block programming, computational thought, Project Geb10s, early programming teaching

Introdução

Desde o ano letivo de 2017/18 que a Escola Superior de Educação de Bragança, por intermédio do seu Centro de Competências TIC, está envolvida no projeto Gen10s, iniciativa que impulsiona a inclusão de atividades de programação em Scratch em diversas comunidades escolares de Portugal. O Scratch é uma linguagem de programação visual, por blocos, amigável, dirigida sobretudo a crianças com idades a partir do que em Portugal corresponde ao 1º Ciclo de Ensino Básico e que permite criar jogos, animações e histórias interativas. Existe ainda uma versão mais simples, o Scratch Jr., dirigido a crianças mais novas. Este ambiente de programação foi desenvolvido com o objetivo de promover nos alunos um conjunto de aptidões, tais como: competências digitais; competências de comunicação e sociais; competências relacionadas com as áreas disciplinares e não disciplinares, bem como o pensamento computacional, o raciocínio lógico, a resolução de problemas e a criatividade. A presente investigação incidirá sobre a operacionalização da iniciativa Gen10s em quatro concelhos do distrito de Bragança, iniciativa essa que já abrangeu no referido distrito cerca de meio milhar de alunos do 5º ano CEB (10/11 anos de idade).

Destaca-se o contributo deste projeto para (i) a promoção da criatividade e o sucesso escolar; (ii) a escolha de atividades de aprendizagem de programação com Scratch relacionando-as com os conteúdos das disciplinas das turmas envolvidas; (iii) o incentivo à produção e apresentação de trabalhos em grupo, promovendo a igualdade de género e a partilha das experiências ou programas. Termina-se apresentando algumas sugestões para futuras edições deste projeto, incorporando as lições aprendidas.

Metodologia

Este trabalho segue um percurso de carácter qualitativo, essencialmente analítico/descritivo, utilizando por um lado as notas produzidas durante o trabalho de campo pelos professores responsáveis pelo desenrolar do projeto realizado e os depoimentos recolhidos de forma informal junto dos professores de turma que acompanharam este percurso e, por outro, a análise de todo os trabalhos realizados pelos alunos como condição para a conclusão com sucesso deste percurso de aprendizagem.

Foram ainda utilizados registos documentais disponíveis com especial incidência sobre os relatórios institucionais finais das sucessivas implementações como forma de descrever sucintamente os objetivos,

tipo de escolas e alunos envolvidos, recursos utilizados, meios humanos e materiais disponíveis.

O material recolhido foi sujeito a uma análise de tipo fundamentalmente interpretativo no que respeita a relatórios e depoimentos e descritivo no caso das produções dos alunos, com base numa metodologia qualitativa. As produções dos alunos, especificamente, são analisadas em conformidade com perspetiva tripla: a) a qualidade e eficácia das soluções desenvolvidas pelos alunos; b) a conformidade das mesmas com os objetivos do projeto e, c) a sua adequação aos conteúdos programáticos das disciplinas curriculares dos alunos envolvidos, um dos objetivos fundamentais do programa. É particularmente no que respeita a esta última análise que se desenvolveu a grelha analítica que é apresentada na conclusão desta investigação.

Os resultados são analisados de forma qualitativa e quantitativa, de forma a permitir uma compreensão profunda das questões relevantes, retirando ilações sobre pontos positivos e problemáticos encontrados, de modo a minimizar futuras disfuncionalidades, comparando os resultados obtidos nos três anos de aplicação do projeto.

Programação por blocos e pensamento computacional

A relação entre ciências da computação e a educação é quase tão longa como a história das primeiras. O primeiro curso conhecido neste domínio, nos EUA, abre em 1962 na Universidade de Purdue (o precursor data de 1953, em Cambridge, com a contratação de Alan Turing para leccionar a incipiente área) e, pouco mais de dois anos depois, a universidade de Dartmouth inicia uma experiência de divulgação de utilização maciça e transdisciplinar de uma linguagem de programação - o BASIC (Kong, Abelson, & Lai, 2018). Se o incipiente entusiasmo gerado no meio educacional pelas potencialidades percebidas das ciências informáticas produziu ou não os resultados esperados é um ponto demasiado complexo para ser abordado aqui, embora haja evidências que apontam para que esses resultados estivera bastante abaixo das expectativas criadas (Norris, Sullivan, Poirot, & Soloway, 2003). Para a questão aqui especificamente abordada, a história começa na segunda metade dos anos 60 com o trabalho de Seymour Papert, mais especificamente com o desenvolvimento do que se vieram a chamar as teorias construcionistas e, particularmente, a criação e divulgação da linguagem LOGO, a partir de 67.

A construção do construcionismo

O que ficou conhecido como construcionismo radica fundamentalmente no trabalho desenvolvido por Seymour Papert no MIT (*Massachusetts Institute of Technology*) a partir de 1963 e sobretudo, a partir de 1967 quando começa a desenvolver a linguagem LOGO.

Seymour Aubrey Papert (1928/2006) licencia-se em filosofia (1949) após o que desenvolve trabalho de doutoramento em matemática (1952) em Joanesburgo e em Cambridge (1959). Entre 1948 e 1963 desenvolve trabalho como investigador no Centro de Epistemologia Genética da Universidade de Geneve onde colabora com Jean Piaget, cujas ideias irão influenciar de forma decisiva o seu futuro percurso.

A partir de 1963 radica-se nos EUA tornando-se investigador e mais tarde professor no MIT – até 96 – onde desenvolveu o seu trabalho mais importante. Em 1970 participou com Marvin Minsky na criação do Laboratório de Inteligência Artificial do MIT e em 1985 foi cofundador com Nicolas Negroponte e o já referido Minsky do MIT MediaLab.

É no âmbito desta atividade de investigação que as primeiras versões da linguagem LOGO são desenvolvidas em colaboração com, entre

outros, Cynthia Salomon. As primeiras versões da linguagem, disponíveis no início dos anos 80, podiam ser adquiridas para várias plataformas – Commodore, Atari, Apple – por preços (não contando com o custo dos equipamentos) que variavam entre \$60 e \$100 dependendo das plataformas e potencialidades (Ames, 2018, p. 184). Devido ao elevado preço envolvido, muitos dos projetos foram patrocinados quer pelo próprio MIT quer por instituições governamentais como o projeto patrocinado pelo governo francês que a partir de 1982, apoia a criação de salas LOGO no Senegal (Ames, 2019, p. 25). As limitações técnicas do LOGO, hoje disponível gratuitamente (Logo Foundation, 2020), levam ao seu quase abandono no final dos anos 80.

O construcionismo dito forte – palavra com N e não com V (distinção importante face ao construtivismo e que o próprio Paper repetidamente referiu) – cuja tentativa de definição seria, nas palavras do próprio, oximorónica (Papert, 1991) está influenciado pelo pensamento de Piaget e MacLuhan. Se a conexão a Piaget é por demais conhecida e estudada, já a ligação a Macluhan – óbvia conhecendo o pensamento de ambos – torna-se explícita, por exemplo, no trabalho de discípulos como Shrutu Dhariwal (Dhariwal, 2008). Para Papert a capacidade do

ambiente em fornecer os “materiais” necessários para a criança desenvolver os seus próprios algoritmos hermenêuticos era fundamental para o processo de construção e o LOGO, tal como o próprio computador, poderia ser uma ferramenta para atingir esse fim. Ambos constituem uma “encarnação” de objetos «com os quais pensar», objetos nos quais se encontra uma intersecção da presença cultural, do conhecimento embebido e a possibilidade de identificação pessoal (Papert, 1981, p. 11). “Na minha visão a criança programa o computador e ao fazê-lo adquire em simultâneo um sentimento de domínio sobre um artefacto da tecnologia mais moderna e poderosa e estabelece contacto íntimo com algumas das ideais mais profundas da ciência, das matemáticas e da arte de construção de modelos” (Papert, 1981, p. 5).

Muito resumidamente, implica que os alunos desenvolvam as suas próprias conclusões a partir de experimentação criativa e interação com objetos sociais. A aprendizagem deverá estar referida a situações/problemas concretos e autênticos permitindo ao aluno sentir que está a ser apoiado de forma a dominar a questão, encorajando o teste de diferentes ideias em diferentes contextos. O resultado esperado será o de uma reflexão quer sobre conteúdo quer sobre processo que conduza a uma apropriação por parte do aluno do problema abordado

(Ackerman, 2001; Papert, 1981; Papert, 1992). O professor, nesta perspetiva, assume, obviamente, muito mais o papel de um facilitador de ocasiões e ambientes de aprendizagem que o de um transmissor de conhecimentos.

O projeto Lego Mindstorms nasceu de uma cooperação iniciada em 1985 entre a LEGO e o MIT (mais especificamente o grupo dirigido por Papert e Resnick) e que começa a dar frutos a partir de 96 constituiu uma primeira tentativa de massificar aquilo que mais tarde ficou conhecido como nome de robótica educativa (Masrill, et al., 2019). Este projeto, permitindo a construção de robots controláveis através de um computador utilizando uma linguagem de programação – Scratch ou outra – está hoje na sua terceira “evolução”, para utilizar a terminologia oficial.

A iniciativa One Laptop per Child, fundada em 2005 também com Negroponte – o computador pessoal de menos de 100 dólares que também ficou conhecido como “the children’s machine” com o objetivo de, como o nome indica, dotar cada criança do mundo com um computador portátil – constituiu um dos seus últimos projetos públicos.

Papert morreu em 2016 em consequência de complicações de um aci-

dente sofrido dez anos antes ao assistir a um congresso sobre Educação Matemática em Hanói.

Desenvolvendo o pensamento computacional

Construído sobre pensamento de Papert – a própria expressão, tanto quanto se pode determinar, aparece pela primeira vez num artigo do próprio Papert (Papert, 1996 (a), p. 116), é popularizado por Jeanette Wing (Wing, 2006) e sobretudo utilizado por Mitchell Resnick (Resnick, 2017) – o conceito de pensamento computacional, extensão daquilo a que Papert chamava “pensamento procedimental” (Papert, 1981, p. 154), pode ser condensado num conjunto de capacidades mentais e práticas para: 1) construir “algoritmos” que levem os computadores a fazerem coisas que nós não conseguimos e; 2) explicar e interpretar o mundo enquanto uma rede complexa de sistemas de informação (Denning & Tedre, 2019, p. 13).

Para adiantar uma definição operacional de pensamento computacional citamos a proposta pela International Society for Technology in Education (ISTE) em colaboração com a Computer Science Teachers Association (CSTA), que o define como um processo de resolução de problemas incluindo (mas não limitado a) as características seguintes (ISTE/CSTA, 2011):

- Formular problemas de forma a permitir a utilização do computador ou outras ferramentas para auxiliar a sua resolução;
- Organizar e analisar dados de forma lógica;
- Representação de dados através de abstrações tais como modelos e simulações;
- Automatização de soluções através de pensamento algorítmico (sequências de etapas ordenadas);
- Identificação, análise e implementação de soluções possíveis com o objetivo de alcançar a combinação mais eficaz de fases e recursos;
- Generalização e transferência deste processo de resolução de problemas para um largo espectro de questões.

Estas capacidades são apoiadas e melhoradas através de um certo número de disposições ou atitudes que constituem dimensões essenciais do pensamento computacional que incluem:

- Confiança em trabalhar com a complexidade;
- Persistência ao trabalhar com questões difíceis;
- Tolerância para com a ambiguidade;
- Capacidade de lidar com problemas abertos;
- Capacidade de comunicar e trabalhar com outros para atingir um objetivo ou solução comum.

Beecher (2017, p. 9), depois de passar em revista diversas definições, sintetiza os diversos pontos discutidos acima em sete “skills” fundamentais – pensamento lógico, pensamento algorítmico, decomposição, generalização e padronização, modelação, abstração e avaliação

– e 5 secundários – representação de dados, pensamento crítico, informatização, automatização, simulação/visualização.

Resnick (2019), sintetiza ainda mais, condensando os diferentes requisitos naquilo a que chama os 4 P da aprendizagem criativa: Projetos, Paixão, Pares e (Play) Brincadeira.

Termina-se esta breve recensão notando que existem evidências interessantes sobre o facto de que, tanto este pensamento computacional, como aquilo a que poderemos designar de forma muito genérica de literacia informática, podem ajudar a aproximar o perfil dos diplomados à saída dos percursos académicos tradicionais com os perfis que são, de facto, procurados pelo mercado de trabalho (Kalas & Winckzer, 2008).

Até ao Scratch

A linguagem Scratch é desenvolvida por Mitchell Resnick e Yasmin Kafai durante a primeira década do século XXI (Resnick, et al., 2006). Depois de versões experimentais em 2002 e 2005, a primeira versão operacional é apresentada ao público em 2007 (8 de janeiro) e a 3 de janeiro de 2019 é lançada a atual versão, 3.0 que, além de apresentar distintas potencialidades em relação à versão anterior, resolve alguns

problemas de segurança advindos da dependência do Scratch da plataforma ActionScript/Flash, passando a funcionar em HTML5/JavaScript (Resnick M. , 2019; Frang, 2019).

O Scratch é uma linguagem de programação por blocos dirigida sobretudo a crianças e jovens, a maioria dos utilizadores tem entre 8 e 16 anos (existe um recurso similar, o Snap, (Mönig & Harvey, 2020) mais poderoso mas menos intuitivo, desenvolvido pela universidade de Berkeley e disponível em <https://snap.berkeley.edu/> dirigido a um faixa etária mais alta). O utilizador pode criar estruturas complexas e diversificadas – histórias, quizzes, animações, diagramas, figuras geométricas, etc. - a partir de instruções pré-codificadas ou então criar as suas próprias instruções. A plataforma pode ser acedida online a partir do site oficial no MIT (<https://scratch.mit.edu/>) ou descarregada e instalada num computador pessoal, estando disponível para os sistemas Windows, MacOS e Linux e sendo compatível com a maioria dos dispositivos e navegadores Web, embora a transição para o JavaScript tenha provocado alguns problemas de compatibilidade.

Segundo as últimas estatísticas, a comunidade Scratch tinha no momento de escrita deste artigo perto de 55 milhões de utilizadores registados com perto de 54 milhões de projetos submetidos (Scratch,

2020).

Resnick é muito claro nos seus objetivos: se bem que a utilização desta ferramenta para aquisição de capacidades “técnicas” de programação seja possível e importante, o desenvolvimento do pensar criativamente, o raciocinar de forma sistemática e trabalhar colaborativamente são primordiais (Resnick, 2012, p. 56).

O Projeto GEN10s

O projeto GEN10s nasce em Espanha de uma colaboração entre a Fundação Ayuda en Acción e a Google, no ano de 2015 com o objetivo fundamental de desenvolver no público infantil para além da literacia informática geral, a capacidade de pensar computacionalmente. Em vista do seu sucesso, em 2017, a Escola Superior de Educação de Setúbal lança com o apoio da SIC Esperança uma versão portuguesa tendo como finalidades declaradas: 1) proporcionar formação em programação a crianças de comunidades mais carenciadas; 2) combater disparidades de género na área da tecnologia; 3) trabalhar competências pessoais e sociais, promover competências digitais e trabalho em equipa; 4) estimular a criatividade; 5) promover o sucesso educativo e reduzir o abandono escolar; 6) dotar os professores e escolas de ferramentas tecnológicas, incentivando a sua integração nas diferentes

áreas curriculares; 7) promover a sustentabilidade de atividades de programação na comunidade escolar (Projecto GEN10s Portugal, 2020). O sistema de incentivos então montado visa sobretudo valorizar as escolas que priorizam projetos inovadores. Logo no ano seguinte a Escola Superior de Educação de Bragança, através do seu Centro de Competências TIC, associa-se ao projeto com o objetivo de o estender ao Nordeste Transmontano.

O projeto está aberto a qualquer escola que ministre o 5.º ou 6.º anos da Ensino Básico e que reúna as seguintes características: 1) deve garantir um posto de trabalho a, pelo menos, cada dois alunos de cada uma das turmas participantes; 2) cada turma participante deverá ser acompanhada por entre um a três professores (de qualquer área disciplinar); 3) a escola deverá ter espaço físico para acomodar um clube Gen10s (a formar opcionalmente após o termino da ação) que funcione como elemento replicador e disponibilizar o apoio de um professor para este espaço. Cada escola poderá candidatar um mínimo de uma turma, podendo, no máximo, candidatar todas, sendo cada turma apoiada por um grupo de dois professores Scratch (um para turmas com menos de 15 alunos) e pelos professores das diferentes áreas disciplinares envolvidas.

Implementação

A primeira fase consiste na preparação dos professores da turma por elementos do grupo de professores Scratch que integram a equipa pertencente ao instituto Politécnico de Bragança. Nesta primeira fase pretende-se que os professores participantes adquiram alguma proficiência na utilização do Scratch, interiorizem os objetivos gerais do projeto e comecem a construir expectativas sobre formas de aplicação a distintas áreas disciplinares.

Na fase seguinte – que se poderá designar como fase de trabalho de campo – os professores Scratch deslocam-se às turmas onde, durante 5 sessões de 1,5 horas, exploram com os alunos as potencialidades da linguagem (existe uma bateria de exercícios/exemplos já construídos, mas a sua utilização não é mandatória). Durante este período os alunos devem ir definindo um projeto, à sua escolha, a realizar em grupo (2 a 4 alunos) que tenha preferencialmente aplicabilidade numa área disciplinar de sua escolha e interesse. A última sessão de trabalho com os professores Scratch é dedicada à definição final destes projetos. Durante estas nove horas as atividades foram acompanhadas e apoiadas pelos professores da turma responsáveis pelas disciplinas em cujo horário foi ministrada a formação referida acima.

Paralelamente decorre uma outra série de sessões, também de 9 horas de duração, durante as quais os alunos desenvolvem, com o apoio dos seus professores de turma, os projetos que idealizaram. Nestas sessões os professores Scratch não estão presentes, mas estão habitualmente disponíveis para apoiar o trabalho e resolver quaisquer questões ou dúvidas que possam surgir.

A última hora e meia de trabalho conjunto é dedicada à apresentação dos trabalhos realizados pelos distintos grupos.

Tabela 1 - Escolas professores e alunos abrangidos no distrito de Bragança

Ano	Escola	Turmas	Masc.	Fem.	Total	Prof Turma	Prof Scratch
17/18	Brag _a	3	30	24	54	6	2
	Brag _b	6	70	49	119	6	4
	Totais	9	100	73	173	8	6
18/19	Brag _a	1	12	7	19	3	2
	Brag _B	2	28	19	47	6	4
	Totais	3	40	26	66	7	6
19/20	Brag _B	2	29	23	43	5	4
	Dist _a	2	19	22	41	5	2
	Dist _b	2	24	17	41	5	2
	Dist _c	2	11	12	23	3	1
	Totais	8	74	74	148	16	4
17/20	Total Final	16	181	135	316	25	6

Legenda: Brag_a e Brag_b referem os agrupamentos de escolas da cidade de Bragança envolvidos nos diferentes anos letivos. Dist_a, Dist_b e Dist_c as escolas do concelho abrangidas.

Para uma melhor compreensão da tabela é de ter em conta que os números totais não refletem a soma das parcelas apresentadas uma vez que no ano letivo de 19/20 o projeto teve de ser interrompido em três das turmas em que ainda estava a decorrer, em virtude das restrições implicadas pela pandemia de Covid-19 (as duas turmas designadas Dist_c e uma das turmas de Bragança). Os totais finais referem-se aos professores e alunos que, de facto, foram abrangidos pelo projeto e completaram a formação. Em relação aos professores (de turma e Scratch), aconteceu frequentemente que o mesmo professor trabalhava em mais que uma turma o que explica uma aparente disparidade dos números apresentados. Mais uma vez, os totais refletem os números de facto envolvidos.

Para a análise do projeto recorreu-se não só à experiência própria e ao feedback pessoal de todos os restantes professores Scratch bem como aos relatórios de acompanhamento produzidos e às produções finais dos alunos recolhidas.

Como pontos positivos ressalta sobretudo o entusiasmo generalizado dos alunos participantes – muito devido não só à manipulação de uma nova ferramenta de carácter marcadamente lúdico como a uma assinalável quebra da rotina semanal. Também em muitos casos houve

um envolvimento assinalável dos professores de turma nos projetos. Como pontos menos positivos, da parte dos alunos envolvidos sentiu-se sobretudo na definição das temáticas de trabalho e na dificuldade muito generalizada de os ligar a conteúdos disciplinares – esta questão exigirá uma investigação mais detalhada. Essa falha foi geralmente colmatada quer através de sugestões dos professores de escola envolvidos, quer através de exemplos fornecidos pelos professores Scratch. Ainda no que diz respeito aos trabalhos propostos nesta fase foi frequente uma assunção de objetivos por demais ambiciosos que tiveram de ser trazidos à realidade pelas equipas responsáveis. Em contraste, verificou-se em bastantes casos que os projetos apresentados acabaram por se resumir a replicações dos exemplos estudados nas sessões de formação.

Relativamente aos professores de turma, pese embora o envolvimento já acima assinalado, alguns casos houve em que se notou uma falta de informação prévia em relação ao contexto do projeto (por exemplo, ignorando os mecanismos de participação ou os objetivos últimos). Tentou resolver-se este conjunto de questões através de uma atempada formação dos docentes envolvidos sendo de salientar que esta não decorreu com a eficácia que seria desejável – em muitos casos não foi

possível encontrar uma grelha temporal compatível com as limitações de todos os envolvidos.

Decorrente presumivelmente dessa falta de formação atempada, notou-se da parte de alguns docentes de turma o sentimento de que o projeto iria retirar horas à lecionação dos conteúdos curriculares obrigatórios e que, portanto, poderia ser detrimental para o percurso académico dos alunos. Na maior parte dos casos, e muito devido ao facto das sessões decorrerem habitualmente em horas dedicadas a estudo acompanhado, essa relutância foi-se esbatendo com o decurso do projeto. Houve mesmo uma situação de uma professora que se recusou a participar, pelo facto de, nesse ano, a sua disciplina constar nas provas de aferição e, por essa razão, necessitar de tempo para a lecionação dos conteúdos.

Aspetos de tipo institucional como questões surgidas com a determinação de horários compatíveis (talvez o maior problema encontrado) não são abordadas nesta análise por, na sua grande maioria, terem sido resolvidas satisfatoriamente. Não poderemos também menosprezar alguma lentidão correspondente à baixa largura de banda de uma das escolas. Essencialmente ultrapassada com estratégias de atividades de utilização da versão desktop do Scratch.

Já em relação às produções dos alunos haverá algumas questões a considerar. Por um lado, como já foi dito, houve na altura de definir os projetos de grupo finais, uma certa dificuldade motivada quer pelo facto de ter sido geralmente difícil encontrar espontaneamente uma ligação a conteúdos disciplinares quer por ter sido também muito frequente uma sobrestimativa das potencialidades da linguagem – por exemplo idealizando aplicações por demais complexas – e das capacidades próprias. Notou-se também em diversos casos que, após o término das sessões como os professores Scratch, houve uma diminuição das atividades de programação.

Uma análise preliminar (Carvalho Sousa, Patrício, & Gonçalves, 2019) tinha já revelado estas mesmas tendências apontando para que quase 57% dos trabalhos analisados dificilmente se enquadravam de forma clara em áreas disciplinares (e neste caso com uma grande incidência para a área de história – quase 40% dos trabalhos integráveis tinha optado por essa área o que, pensou-se, foi devido ao carácter marcadamente narrativo da disciplina). Já em relação às tipologias utilizadas verificou-se uma clara opção pelas narrativas (cerca de 51%) seguidas pelos jogos (32%) e dos quizzes (17%). Sem grandes surpresas, verificou-se que esta última tipologia foi a que mais facilmente se

adequou à utilização de conteúdos curriculares sendo a totalidade dos trabalhos integráveis nos objetivos enquanto que os jogos constituíram a categoria de mais difícil utilização nenhum dos projetos analisados obedecendo aos requisitos de integração curricular. Já na área das narrativas os projetos dividiam-se sensivelmente a meio em relação a este requisito sendo que mais de 40% deles lhe obedeciam.

Em relação à estrutura técnica dos projetos a mesma análise apontou para facto de a maioria deles (na casa dos 90%) essencialmente replicações e modificações dos exemplos analisados durante a formação, apenas menos de 10% demonstrando capacidades claras de inovação. Esta análise demonstrou a necessidade de se desenvolver uma grelha analítica que pudesse simultaneamente dar conta da adesão dos trabalhos submetidos às regras gerais do projeto e, da complexidade técnica, conceptual e criativa dos mesmos. É essa grelha que se apresenta e fundamenta a seguir para discussão e que se espera aplicar à análise da totalidade dos trabalhos recolhidos até ao momento.

A análise das produções

Para a construção de uma grelha de análise operacional recorreu-se a três fontes distintas:

1) uma grelha de análise elaborada no âmbito do próprio projeto

Gen10s (Sá, 2019); 2) a estrutura proposta por Brennan e Resnick (2012) e; 3) a operacionalização dos critérios ISTE/CSTA referidos mais acima por Williams (2017, pp. 34, sgs).

A grelha desenvolvida apresenta-se a seguir e, após validação, poderá ser utilizada para uma exploração mais informada dos trabalhos recolhidos.

Quadro 1 - Proposta de parâmetros, especificações e indicadores para uma grelha analítica dos trabalhos recolhidos

Parâmetro	Especificação	Indicador
Integração Curricular	O produto obedece às especificações de poder ser integrado numa área disciplinar frequentada pelo aluno	O trabalho do aluno diz respeito a um conteúdo abordado numa disciplina
Destreza computacional	O produto demonstra domínio das principais técnicas computacionais	Utilização de: sequências, loops, eventos, paralelismo; condições, operadores, gestão de dados
Práticas aplicadas	O aluno aplicou no desenvolvimento do seu produto práticas que demonstram domínio do pensamento computacional	Utilização de: iteratividade, incrementação, teste/debugging”, reutilização de rotinas, modularização
Capacidade Expressiva	Desenho e criatividade	Utilização de elementos originais; utilização original de elementos alheios
Questionamento e conceptualização	O trabalho apresenta uma estrutura lógica e coerente Aborda temas/conteúdos originais, ou de uma forma original.	Clareza de estrutura do projeto Originalidade dos temas Originalidade das abordagens Originalidade dos resultados
Pesquisa de informação	O trabalho utiliza informação pertinente	A informação é adequada A informação foi devidamente pesquisada
Análise de informação	A informação utilizada no trabalho é transformada ou trabalhada de forma a serem atingidos os objetivos propostos	Utilização de operações sobre os dados Utilização dos dados existentes para gerar novos dados

A colaboração dos elementos do CCTIC com as diferentes escolas

participantes no projeto resultou na produção de uma centena de realizações que, num próximo trabalho nos propomos, como foi dito, analisar aplicando a grelha acima de forma a obter informação que permita uma condução mais proveitosa desta iniciativa ou de outras afins.

Conclusões

O pensamento computacional é, hoje, um movimento educativo à escala global. A linha original iniciada por Papert e continuada por outros investigadores justifica, hoje, a existência de uma grande variedade de recursos de programação para crianças, entre os quais se destaca o Scratch. O Gen10s procura inovar pedagogicamente, procurando envolver as crianças na elaboração de pequenos projetos em grupo, com base no Scratch, procurando ir além da pedagogia tradicional no desenvolvimento da criatividade.

Ao fazer uma análise da implementação do projeto na região de Bragança, salientamos a motivação e envolvimento dos alunos. Alguma resistência – embora tenha sido mais verificada a nível de interações não formais entre os professores Scratch e os professores de turma – tem sido notada por parte de alguns dos docentes envolvidos, mais

centrados numa pedagogia tradicional, para os quais o trabalho realizado em Scratch, reduz o tempo de transmissão de conteúdo ou matéria disciplinar. Esta incoerência entre a metodologia tradicional e a inovação pedagógica de suporte tecnológico, é algo que a escola terá de resolver para que a competência de programação possa ser uma nova habilidade de pensamento que desenvolve habilidades cruciais do século XXI.

Muito deste trabalho baseou-se na perceção por parte de alguns dos intervenientes de que, apesar do interesse e entusiasmo manifestado pela esmagadora maioria dos alunos envolvidos, se verificou uma certa dificuldade em responder de modo totalmente satisfatório aos requisitos originais do projeto. A elaboração de uma grelha de avaliação devidamente testada e validada e a sua aplicação aos trabalhos recolhidos permitirá, pensamos, uma melhor compreensão e uma mais eficaz gestão de futuras edições deste projeto.

Referências Bibliográficas

- Ackerman, E. (2001). Piaget's constructivism, Papert's constructionism: What's the difference. *Constructivisms: Usages et Perspectives en Education*. 1, pp. 85-94. Geneva: SRED.
- Ames, M. G. (2018). Hackers, computers and cooperation: A critical history of LOGO and constructionist learning. *Proceedings of the ACM on Human-Computer Interaction*. 2, pp. 18:1-18:13. New York: ACM.
- Ames, M. G. (2019). *The charisma machine: The life, death, and legacy of One Laptop per Child*. Cambridge, MASS: MIT Press.
- Beecher, K. (2017). *Computational thinking: a beginners's guide to problem-solving and programming*. Swindon: BCS.
- Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *American Educational Research Association Meeting* (pp. 1-25). Vancouver: AERA.
- Carvalho Sousa, J., Patrício, R., & Gonçalves, V. (2019). Uma experiência de ensino precoce de programação. *Ietic 2019* (pp. 53,54). Bragança: IPB.
- Denning, P. J., & Tedre, M. (2019). *Computational thinking*. Cambridge, MASS: MIT Press.
- Dhariwal, S. (2008). *Scratch memories*. Cambridge, Mass: MIT Press.
- Frang, C. (2019, Fevereiro 28). *Porting Scratch from Flash to JavaScript: Performance, interoperability and extensions*. Retrieved from bocoup: <https://bocoup.com/blog/porting-scratch-from-flash-to-javascript-performance-interoperability-and-extensions>
- ISTE/CSTA. (2011). *Operational definition of computational thinking for K-12 education*. Retrieved from International Society for

- Technology in Education: <https://id.iste.org/docs/ct-documents/computational-thinking-operational-definition-flyer.pdf?sfvrsn=2>
- Kalas, I., & Winckzer, M. (2008). Informatics as a contribution to the modern constructivist education. In R. T. Mittermeier, & M. M. Syslo, *Supporting computational thinking* (pp. 229-240). Berlin: Springer.
- Kong, S.-C., Abelson, H., & Lai, M. (. (2018). Introduction to computational thinking education. In S.-C. Kong, & H. Abelson, *Computational thinking education* (pp. 1-10). Singapura: Springer.
- Logo Foundation. (2020). *Logo Resources - Software/Hardware*. Retrieved from Logo Foundation: https://el.media.mit.edu/logo-foundation/resources/software_hardware.html
- Mönig, J., & Harvey, B. (2020). Snap. Berkeley: Berkeley.
- Masrill, M., Hendrik, B., Fikri, H. T., Hazidar, A., Priambodo, B., Naf'an, E., . . . Nseaf, A. K. (2019). The effect of Lego Mindstorms as an innovative educational tool to develop students' creativity skills for a creative society. *International Conference Computer Science and Engineering* (pp. 1-10). Bristol: IOP Publishing.
- Norris, C., Sullivan, T., Poirot, J., & Soloway, E. (2003, Fall). No access, no use, no impact: Snapshot surveys of educational trchnology in K-12. *Journal of Research on Technology in Education*, 36(1), p. 15.27.
- Papert, S. (1981). *Mindstorms - Children, computers and powerfull ideas*. New York: Basic Books.
- Papert, S. (1991). Situating constructionism. In I. E. Harel, & S. E. Papert, *Constructionism: research reports and essays, 1985-1990*. Norwood, NJ: Ablex.
- Papert, S. (1992). *The children's machine - Rethinking school in the age of the computer*. New York: Basic Books.
- Papert, S. (1996 a). An exploration in the space of mathematics education. *International Journal of Computers for Mathematical Learning*(1), pp. 85-123.
- Papert, S. (1996 b). *The connected family - Bridging the digital generation gap*. Marietta, GA: Longstreet Press.
- Projecto GEN10S Portugal. (2020). *Proecto Gen10s Portugal*. Retrieved from Gen10s.org: <http://genios.org.pt/>
- Resnick, M. (2012). Mother's day, warrior cats and digital fluency: Stories from the scratch online coomunity. *Constructionismo 2012* (pp. 52-58). Athens: ACM.
- Resnick, M. (2017, Fevereiro 3). *The seeds that Seymour sowed*. Retrieved from Mit MediaLab: <https://www.media.mit.edu/posts/the-seeds-that-seymour-sowed/>
- Resnick, M. (2019). *Creativity and Learning in the Era of AI [Prefácio à edição coreana de Lifelong Kindergarten]*. Retrieved from medium.com: <https://medium.com/@mres/creativity-and-learning-in-the-era-of-ai-57eea387249d>
- Resnick, M. (2019, Janeiro 3). *The next generation of Scratch teaches more than coding*. Retrieved from edsurge: <https://www.edsurge.com/news/2019-01-03-mitch-resnick-the-next-generation-of-scratch-teaches-more-than-coding>
- Resnick, M., Maloney, J., Monroy-Hernandez, A., Rusk, N., Eastmond, E., Brennan, K., . . . Kafai, Y. (2006). Scratch: Programming for all. *communications of the ACM*, 52/11, pp. 60-67.
- Sá, J. (2019). *Grelha de avaliação proposta para o Projecto GEN10s*. Retrieved Março 12, 2020, from Escola Superior de Educação de Setúbal: http://projectos.esse.ips.pt/genios/geral/lista_arquivos_geral.php

Scratch. (2020, Maio 3). *Community statistics at a glance*. Retrieved from Scratch: <https://scratch.mit.edu/statistics/>

Williams, H. (2017). *No fear coding. Computational thinking across the k-5 curriculum*. Portland. OR: ISTE.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 9/3, pp. 33-36.

Notas curriculares

Manuel Meirinhos, Professor coordenador do departamento de Tecnologia Educativa e Gestão da Informação, mestre em Tecnologia Educativa, doutor em estudos da Criança, especialidade Tecnologias de Informação e Comunicação, diretor do mestrado TIC na Educação e Formação, Coordenador do CCTIC ESE/IPB, membro da equipa que organizou a Conferência Ibérica iETIC e outros eventos científicos, membro integrado do CIEB (Centro de Investigação em Educação Básica), membro do conselho científico de várias conferências e revistas científicas nacionais e internacionais, investigador em vários projetos de investigação, autor de vários artigos em atas de eventos científicos e revistas internacionais, ampla experiência na formação inicial e contínua de professores.

João Sérgio de Pina Carvalho Sousa, Licenciado em Línguas e Literaturas Modernas (UP), Bacharel em Engenharia Electrotécnica (UP). Prof. Adjunto do depº de Tecnologia Educativa e Gestão de Informação da ESEB. Provas públicas na área da Literatura Infantil e Juvenil – Estruturas Narrativas da Banda desenhada, Docência: Ucs de TIC em CTESPS Licenciaturas e Mestrados, Docência de UCs de mestrado em S. Tomé e Príncipe (cerca de 5 anos) UCs ligadas à Edição não linear de vídeo. Organização do IETIC e INCTE (três anos). Diretor de curso CTESP Multimédia desde a sua criação (15 anos mais ou menos). Diversos artigos e capítulos publicados na área da análise estrutural de redes sociais e da análise de conteúdo. Recentemente convidado para participar numa mesa redonda na U Vigo sobre a utilização de tecnologias disruptivas no ensino das TIC.

Maria Raquel Patrício, Professora adjunta no Departamento de Tecnologia Educativa da Escola Superior de Educação (ESE) do Instituto Politécnico de Bragança (IPB). Doutorada em Ciências da Educação na especialidade de Tecnologia Educativa pela Universidade do Minho. Investigadora do Centro de Investigação em Educação Básica (CIEB) do IPB. Membro da equipa do Centro de Competências TIC da ESE/IPB e formadora Scratch no Projeto GEN10S. Integra comissões científicas e organizadoras de congressos nacionais e internacionais. Autora e co-autora de artigos em revistas e atas nacionais e internacionais.

Vitor Gonçalves, Professor Adjunto no Departamento de Tecnologia Educativa e Gestão da Informação da Escola Superior de Educação (ESE) do Instituto Politécnico de Bragança (IPB). Doutor em Engenharia Electrotécnica e de Computadores e Mestre em Tecnologia Multimédia pela Faculdade de Engenharia da Universidade do Porto. Foi formador(a) Scratch no Projeto GEN10S.pt. É investigador do Centro de Investigação em Educação Básica (CIEB) do IPB, membro do GPITIC - Grupo de Pesquisa Interdisciplinar em Tecnologia da Informação e Comunicação e coordenador do Gabinete de Relações Internacionais da ESE-IPB.