Educação, Formação & Tecnologias (julho-dezembro, 2014), 7 (2), 4-25
Submetido: outubro, 2014 / Aprovado: novembro, 2014

Os futuros professores e os professores do futuro. Os desafios da introdução ao pensamento computacional na escola, no currículo e na aprendizagem

JOSÉ LUÍS RAMOS[i]
Centro de Investigação em Educação e Psicologia, Universidade de Évora**,** Portugal
jlramos@uevora.pt

RUI GONÇALO ESPADEIRO[ii]
Centro de Competência TIC da Universidade de Évora**,** Portugal
rge@uevora.pt

Resumo

O artigo apresenta os resultados de um estudo realizado com o objetivo de explorar alguns dos desafios da introdução ao pensamento computacional no contexto da formação inicial de professores dos ensinos básico e secundário. Um dos principais diz respeito à avaliação das aprendizagens dos estudantes relativas à aquisição e ao desenvolvimento dos princípios do pensamento computacional. Para este efeito foi adotado um quadro de referência baseado na avaliação de artefactos digitais enquanto produtos resultantes do desenvolvimento de projetos no ambiente computacional *Scratch*.

A amostra foi constituída por 44 alunos de licenciatura e mestrado de diferentes cursos de ensino. Foi adotada uma metodologia qualitativa que teve como principais fontes de informação os artefactos digitais programados, os relatórios e entrevistas focus-group. Os resultados são apresentados e discutida a importância de proporcionar experiências de aprendizagem aos futuros professores, para que estes, como futuros profissionais da educação, possam estar preparados para uma utilização pedagógica inovadora dos ambientes computacionais destinados a crianças e jovens.

Palavras-chave: Formação inicial de professores, pensamento computacional, ensino da computação.

1. INTRODUÇÃO

A inovação tecnológica é um dos sinais que marcam e melhor caracterizam os nossos quotidianos e os seus efeitos fazem sentir-se de forma generalizada em todos os sectores da atividade humana. A escola não é imune aos seus efeitos, antes pelo contrário, ela é crescentemente influenciada pela sociedade do conhecimento (UNESCO, 2005) em que o seu *ethos* estimula a uma relação inovadora, aberta e transformadora entre a sociedade e a escola.

As tecnologias têm, neste tipo de sociedade, um papel de grande relevo, até porque os estudantes e os professores são utilizadores de tecnologias em diferentes contextos e para diferentes finalidades. Não é possível (nem desejável) que esta influência termine abruptamente à porta da escola. Tal seria aliás, contraproducente e paradoxal face ao papel da própria escola e das instituições educativas em geral. A escola e a sociedade não têm entre si "fronteiras" e muito menos de natureza *impermeável*, tendo assim que reinventar-se mutuamente, aprendendo a ultrapassar os desafios que são naturalmente comuns.

A sociedade espera da escola respostas equilibradas e relevantes, no que diz respeito à preparação dos seus membros para a integração harmoniosa nas tarefas e funções sociais de cada um. Esperam-se da escola propostas que permitam proporcionar a todos uma educação moderna e atualizada, incluindo propostas que permitam às crianças e jovens aprender a usar a tecnologia de forma inovadora e criativa, aprender a conhecer e a usar as tecnologias, aprender a programar, aprender a ser e a estar informado, aprender a construir novo conhecimento com recurso às tecnologias disponíveis, aprender a informar (identificar o potencial, as vantagens e também os riscos envolvidos) e avaliar de forma crítica o papel das tecnologias na sociedade, na economia, na cultura e nos estilos de vida dos cidadãos. A escola esperará da sociedade e das suas instituições os meios e os recursos humanos e materiais para cumprir o seu papel.

Pensamento computacional: conceitos e princípios

O pensamento computacional tem recebido considerável interesse por parte da comunidade científica e educativa e resulta, em boa parte, da chamada de atenção de Jeannete Wing que, através do texto seminal "Computational Thinking", escrito em 2006 onde a autora reintroduziu o conceito e reclamou o seu uso e adopção por todos os cidadãos, incluindo jovens e crianças, como forma de proporcionar os conhecimentos e capacidades decorrentes das formas e recursos cognitivos próprios das ciências da computação e que, pela sua natureza transdisciplinar e universal, poderia ser útil a todos, recusando a ideia, até aí dada como adquirida, de que estas capacidades apenas seriam destinadas aos cientistas da computação (Wing, 2006, 2008a, 2008b).

No texto fundacional a autora sustentava que *o pensamento computacional é um conceito crucial na educação e que envolve resolução de problemas, conceção de sistemas e compreensão do comportamento humano, baseados nos princípios das ciências da computação* (…). (Wing, 2006).

Mais recentemente, Wing aprofunda o conceito inicialmente formulado e define o pensamento computacional como sendo o conjunto dos *processos de pensamento envolvidos na formulação de um problema e que expressam a sua solução (s) de tal forma que um computador – máquina ou humano- possa realizar eficazmente. Informalmente, o pensamento computacional descreve a atividade mental na formulação de um problema de modo a que possa ser admitida uma solução computacional. A solução pode ser levada a cabo por uma máquina ou um ser humano. Este último ponto é importante. Em primeiro lugar, os seres humanos computam. Em segundo lugar, as pessoas podem aprender pensamento computacional sem recurso a uma máquina. Além disso, o pensamento computacional não é apenas sobre a resolução de problemas, mas também sobre a formulação do problema*". (Wing J., 2014)

A autora recupera assim a memória e a história da tecnologia e a evolução das ciências da computação, retomando a ideia que a computação é sobretudo uma ação humana e a máquina é inventada pelos humanos para suportar e estender essas capacidades, ampliando-as e aplicando-as na resolução de problemas complexos.

Wing considera que o pensamento computacional vai muito além da capacidade de programar "*por se centrar na conceptualização, não na programação e requer pensamento em múltiplos níveis de abstração; é um tipo de pensamento fundamental (reflexivo e teórico), não rotineiro; uma forma de pensar que os seres humanos utilizam, não os computadores, uma forma de pensamento que combina pensamento matemático e de engenharia, refere-se a ideias, não a artefactos, é um tipo de pensamento para todos e em qualquer lugar"* (Wing J. , 2006)

Ao contrário do que pode eventualmente pensar-se, o pensamento computacional não se reduz à ideia de tecnologia, equipamentos, redes, etc. Philips (2008), reforça que a *essência do pensamento computacional é pensar acerca de dados e de ideias e combinar estes recursos para resolver problemas*. E identifica mesmo alguns aspetos que não correspondem ao pensamento computacional:

- *Não é apenas mais detalhes técnicos para a utilização de software*
- *Não é pensar como um computador*

- *Não é programação (necessariamente)*
- *Nem sempre requer um computador*
- *Não é mais uma "coisa" para acrescentar ao currículo.*

Também Wing (2006, 2008a) nos seus trabalhos refere diversas capacidades analíticas que constituem princípios fundamentais da computação e do pensamento computacional: resolução de problemas, pensamento recursivo, pensamento paralelo, abstração, automação, decomposição, modelação, simulação, para referir apenas algumas destas capacidades. Os princípios do pensamento computacional são descritos de forma mais ou menos exaustiva por diversos autores e os que aqui são indicados e explorados na componente empírica deste estudo, não esgotam essa descrição.

O que parece ser comum a uma grande maioria dos textos que tratam esta temática é a ideia de que o pensamento computacional corresponde a um modo estruturado de raciocínio que é tipicamente usado na resolução de problemas pelos cientistas desta área. A sua aprendizagem é igualmente valorizada pela comunidade científica da computação que sustenta que *pensar é bom para aprender a pensar*. *Sabemos que os alunos de tenra idade são capazes de pensar através de algoritmos. Eles podem usar sequências, análise e testes (…). Como estratégia geral de resolução de problemas, esta capacidade de compreender e descrever os processos no tempo e no espaço (pensamento algorítmico) pode tornar-se uma estratégia que os alunos podem acrescentar ao seu conjunto de ferramentas de resolução de problemas. (…). Além disso, os alunos podem desenvolver hábitos mentais e de perseverança na resolução de problemas, que podem durar uma vida.* (CSTA - Computer Science Teachers Association & Machinery, 2012, p. 11).

Estas capacidades fazem parte das capacidades essenciais dos cientistas da computação. Muitas delas são igualmente fundamentais para outros cientistas de outras áreas. E muitas são igualmente importantes para pessoas comuns, considerando que estas capacidades constituem um conjunto de utensílios mentais que podem e são usadas no quotidiano de muitas pessoas.

Este debate acerca dos conceitos e das características do pensamento computacional e as suas implicações para a educação tem a particularidade de centrar-se não apenas nas capacidades mentais e cognitivas dos indivíduos mas também naquelas capacidades que são associadas à metodologia de resolução de problemas inscrita na matriz das ciências da computação. Por isso, a aprendizagem da computação pode ser caracterizada (mas não limitada) pelos seguintes aspetos:

- Formular problemas numa forma que permita usar um computador e outras ferramentas para o resolver;
- Organizar dados de forma lógica e analisá-los;
- Representar dados através de abstrações, como modelos e simulações;
- Automatizar soluções através de pensamento algorítmico (uma série de passos ordenados);
- Identificar, analisar e implementar soluções possíveis com o objetivo de encontrar a combinação de passos e recursos mais eficiente;
- Generalizar e transferir esse processo de resolução de problemas a uma grande variedade desse tipo de problema. (CSTA - Computer Science Teachers Association & Machinery, 2012).

Estas capacidades apoiam-se num conjunto de atitudes e predisposições, que são dimensões essenciais do pensamento computacional nomeadamente:

- A confiança em lidar com a complexidade
- Persistência em trabalhar com problemas difíceis
- Tolerância à ambiguidade
- A capacidade de lidar com problemas abertos e finitos
- A capacidade de comunicar e trabalhar com outros para atingir um objetivo comum ou solução (Computer Science Teachers Association & Machinery, 2012, p.5).

A capacidade de resolver problemas é igualmente um traço comum, neste esforço de construção coletiva de um quadro de referência do pensamento computacional para a educação.

"*O pensamento computacional é uma abordagem para resolver problemas de uma forma que possa ser implementado com a ajuda de um computador. Os estudantes podem então tornar-se não apenas meros utilizadores, mas os construtores das ferramentas de trabalho. Eles usam um conjunto de conceitos, tais como abstração, recursão e iteração para processar e analisar os dados e a partir daí, criar artefatos reais e virtuais. Esta metodologia de resolução de problemas inerente ao pensamento computacional, pode ser automatizada, transferida e aplicada a outras disciplinas*". (Computer Science Teachers Association & Machinery, 2012, p.4).

A perspetiva adotada pela National Science Foundation (NSF) baseia-se nas sete "grandes ideias" da computação:

- A computação é uma atividade humana criativa.
- A abstração reduz informações e detalhes para se concentrar em conceitos relevantes para a compreensão e resolução de problemas.
- Os dados e informações servem para facilitar a criação de conhecimento.
- Os algoritmos são ferramentas para desenvolver e expressar soluções para problemas computacionais.
- A programação é um processo criativo que produz artefactos computacionais.
- Os dispositivos digitais, sistemas e as redes que os interligam permitem fomentar abordagens computacionais para resolver problemas.
- A computação permite a inovação em outras áreas, incluindo a ciência, as ciências sociais, as humanidades, as artes, a medicina, a engenharia e os negócios (Grover, 2013, p.39).

Formação inicial de professores e ambientes computacionais

Não são apenas as escolas, os professores e os estudantes e respetivas famílias aqueles que devem tomar como seus o tipo de desafios a que nos referimos no início deste texto. Referimo-nos particularmente às instituições em cuja missão se integra a formação das novas gerações de professores. Consideramos que, aos desafios da escola moderna face às aceleradas mudanças neste campo, deverá também corresponder igual esforço das instituições de formação inicial assegurando que os novos professores estão suficientemente preparados para as suas missões de ensino, incluindo uma formação científica e pedagógica sólida, moderna e atualizada.

Neste enquadramento, algumas das competências a desenvolver pelos futuros professores recaem justamente naquelas que capacitam os professores a fazer o melhor uso das tecnologias, no plano da escola, do currículo e da aprendizagem. Como vimos antes, a evolução de uma sociedade baseada no conhecimento, na tecnologia e na inovação, por um lado e a evolução do conhecimento científico e pedagógico relativo aos processos de integração das tecnologias na escola, nas últimas décadas, conduziram-nos à construção de uma importante base de conhecimento sobre as estratégias adequadas à integração das tecnologias na escola. É uma base em construção, aberta à inovação e à mudança mas que constitui uma primeira viagem nos percursos da formação inicial dos novos professores.

Na construção laboriosa e constante desta base de conhecimento estamos hoje (como ontem) expostos a ventos de mudança que varrem e abanam as nossas convicções e nos obrigam a questionar permanentemente o que ensinar, neste contexto, sabendo que, em apenas alguns anos, tudo aquilo que tenha sido objeto de trabalho educativo pode não existir quando estes novos profissionais chegarem à escola que tanto desejaram.

Entre estes ventos de mudança destacamos a emergência de movimentos e iniciativas que reclamam da escola e das instituições de formação, não uma mudança de cosmética, na forma de uma nova

tecnologia ou aplicação, mas pretendem algo mais fundo e duradouro: uma mudança de paradigma.

Estes movimentos são protagonizados por instituições de natureza muito diversa, sobretudo empresariais, mas também académicas, científicas, sociais e culturais em que é proposto um alargamento e um aprofundamento da literacia digital, sublinhando a importância de "*desenhar, criar e combinar e não apenas navegar, conversar e interagir*" (Resnick, 2009). Trata-se de uma proposta que pode ser entendida como um "regresso" à introdução do ensino da programação no currículo escolar, no quadro da educação básica e secundária, agora, "embebida" esta pretensão, num conceito mais amplo de literacia digital que corresponde, para outros autores (Wing, 2006) à introdução a disseminação do pensamento computacional no currículo e na aprendizagem, em todos os níveis de escolaridade.

Note-se que não se trata de uma literacia digital apenas assente na aquisição de competências básicas de literacia informacional ou de conhecimentos básicos de manipulação técnica do equipamento. É, antes, uma literacia digital que se funda nos alicerces cognitivos e sociais da sociedade do conhecimento e por isso põe em causa os nossos modos de pensar, agir, as predisposições e atitudes face ao papel da tecnologia na sociedade e na educação. Esta é a mudança de paradigma que está em causa com a introdução do pensamento computacional na educação das crianças e jovens.

As razões e os fundamentos para estas propostas têm a sua raiz em diversos sectores da sociedade, com particular destaque para os sectores económicos e empresariais (com motivações claras de aumentar o número de pessoas qualificadas nestes domínios da tecnologia e da inovação e em falta nestes sectores da economia) para os sectores científicos e académicos (com motivações no que diz respeito à excessiva e preocupante falta de estudantes nestas áreas e com preocupação adicional pela falta de interesse das raparigas no domínios das tecnologias e das engenharias) mas também de sectores políticos e governamentais (como sejam o caso dos governos de muitos países quer na Europa (onde se incluem as iniciativas europeias da Comissão) quer em outras regiões do mundo, conscientes da importância estratégica de dispor de recursos humanos de elevada qualificação nos seus sectores económicos e com reflexos nos mercados de emprego e de trabalho e nas capacidades de competitividade e inovação destes países.

Exemplos de expressão destes movimentos de ideias educativas quanto ao papel da tecnologia na aprendizagem podem ser facilmente encontrados nos programas e iniciativas a nível nacional e internacional de re-introdução do ensino da programação nas escolas do ensino básico e secundário, um pouco por todo o mundo (Guzdial, 2008; Barr, 2011; Royal Academy of Engineering, 2012). Estes movimentos impõem-se hoje em muitos países, embora com requisitos diferentes em muitos casos. Em alguns países a sua introdução é compulsória (por exemplo, Reino Unido) em outros é facultativa (por exemplo, França) e em outros países ainda, tal definição não está concluída ou apresentam definições variadas e combinadas.

Muitos destes movimentos, que genericamente se agregam à volta do conceito de programação ("to code") estão a suscitar uma ampla e efetiva adesão de escolas e de outras instituições educativas e as suas ondas chegam igualmente às instituições de ensino superior. Alguns autores ressalvam que "numa sociedade crescentemente baseada na informação, o pensamento computacional está a tornar-se numa competência fundamental para qualquer indivíduo" (Yadav, 2011), encontrando, neste princípio, o fundamento das suas abordagens.

Os resultados destas iniciativas estão porém condicionadas à ação sobre outras componentes dos sistemas educativos, das quais destacaremos aqui apenas as componentes relativas à formação dos professores uma vez que, para assegurar que os estudantes desenvolvem esta capacidade até ao 12º ano, é indispensável proporcionar aos professores um conhecimento adequado acerca do pensamento computacional e sobre como o integrar nos seus ensinos" (Yadav A. M., 2014).

Sucedem-se os trabalhos e as reflexões sobre as potencialidades desta abordagem no campo da educação e formação (Barr, 2011; Royal Academy of Engineering, 2012).

No caso mais específico da formação inicial de professores, os estudos empíricos não são abundantes e centram-se na exploração de ferramentas computacionais como o *Scratch*. Por exemplo, um estudo envolvendo 35 estudantes da formação inicial de professores conclui que o uso do ambiente computacional *Scratch* tem um efeito positivo sobre as opiniões e atitudes dos alunos em relação ao ensino da programação de computadores e um importante valor educativo, uma vez que no final da intervenção realizada melhorou a percentagem de estudantes que pretendem vir a explorar as TIC nos processos de ensino e aprendizagem, diminuiu a percentagem de alunos com receio e angustia face às suas capacidades de vir a usar um computador no ensino, aumentando a probabilidade de uma integração bem sucedidada e eficaz nas suas práticas educativas (Fesakis, 2009).

Em Portugal este tipo de estudos estão também no seu início. Um exemplo são os trabalhos que se centram na exploração pedagógica de ferramentas computacionais – como é o caso do ambiente computacional *Scratch* – como suporte ao desenvolvimento de competências cognitivas e sociais em crianças e jovens do 1º ciclo de escolaridade (Marques, M., 2009).

A integração das TIC no currículo e na aprendizagem no campo da formação inicial de professores é, em si, mesma, um desafio. Levando em linha de conta que "à medida que o pensamento computacional se vai tornando numa competência fundamental para o século XXI, os professores devem ser expostos aos princípios da computação" (Yadav, 2011).

Em consequência, parece hoje consensual concluir da necessidade de incorporar experiências de aprendizagem adequadas à introdução e ao desenvolvimento do pensamento computacional, no quadro da formação inicial de professores, em todas as áreas curriculares, e não apenas nas áreas do ensino da informática ou computação (Barr, 2011).

Ambientes computacionais e avaliação das aprendizagens

A avaliação das aprendizagens é um elemento crítico em todos os processos relacionados com o ensino e a aprendizagem do pensamento computacional na escola.

Como avaliar as aprendizagens dos estudantes, decorrentes da exploração de ambientes computacionais? *Sem atenção à avaliação, o pensamento computacional não terá muita probabilidade de seguir o caminho de sucesso em qualquer curriculo (...) Mais do que isso, para avaliar a eficácia de qualquer abordagem curricular de integração do pensamento computacional, medidas que permitam aos professores avaliar o que as crianças aprenderam, necessitam de ser validadas* (Grover, 2013).

Para além das dificuldades da definição do conceito de pensamento computacional, já assinaladas, podemos acrescentar outras dificuldades que resultam de muitas mensagens enviadas pela comunidade científica e pelos media e que arrastam consigo sentidos e significados, por vezes contraditórios. Uma dessas ideias é a de que aprender os princípios do pensamento computacional é apenas aprender código ou programação, o que pode ser algo confuso e ter um impacto negativo na sala de aula, podendo conduzir os professores a focarem-se apenas num segmento do currículo e negligenciar a profundidade e a abrangência do pensamento computacional (Selby, 2014).

Pelo seu lado, Werner (2012) reforça a importância da necessidade de desenvolver conceitos, teorias, modelos e dispositivos de avaliação apropriados, quando observa que "um factor que limita a adopção do pensamento computacional no ensino secundário é a falta de avaliações" (ACM/CSTA, 2010; Werner, 2012).

A constatação desta necessidade e deste problema é por isso um primeiro passo em direção à sua resolução, a partir de um trabalho que

deve ter o seu início na precisão e rigor com que os conceitos relevantes neste domínio devem ser construídos.

Recorde-se que o pensamento computacional deve refletir no seu conteúdo a diversidade de conceitos, teorias e modelos e áreas práticas de aplicação de um campo muito extenso como é o campo das ciências da computação e, neste sentido a sua definição não poderá deixar de ser abrangente, de forma a acolher os diferentes elementos na sua composição. Se acrescentarmos as inúmeras ferramentas e ambientes computacionais disponíveis, destinados especificamente ao uso por crianças e jovens e que podem, cada um à sua maneira, ser usados para estimular diferentes contributos para a aquisição e desenvolvimento dos conceitos e princípios do pensamento computacional (*Scratch*, *Kodu*, *Blocky*, *Alice*, *Angrybirds*, *Minecraft* entre muitos outros), compreendemos a complexidade do empreendimento e as dificuldades que se colocam aos que procuram envolver-se neste tipo de trabalho. Em qualquer caso, a sua definição é o ponto de partida para as tarefas de conceptualização e operação que se impõe à partida neste tipo de trabalho educativo.

A clareza com que é definido o conceito de pensamento computacional é por isso uma prioridade (e uma necessidade) pois na verdade o trabalho educativo a realizar será assente justamente no conteúdo do conceito construído ou adotado e as perspetivas de avaliação da aprendizagem serão igualmente decorrentes desse trabalho. *Uma vez definido de forma adequada o conceito, os instrumentos apropriados de avaliação das aprendizagens podem então ser desenvolvidos*. (National Science Foundation, 2010).

Até ao momento são ainda escassos os trabalhos e as propostas neste domínio da avaliação das aprendizagens no quadro do pensamento computacional.

Podemos ainda assim identificar uma tendência de cariz mais pedagógico e uma de cariz mais tecnológica.

No caso da primeira tendência, a opção tem sido combinar métodos e técnicas de avaliação da aprendizagem. No segundo, a tendência é desenvolver software que permita automatizar uma importante parte da avaliação.

Por exemplo, Franklin (2013) e como exemplo da primeira tendência indicada, descreve uma experiência de apenas duas semanas de trabalho interdisciplinar para 35 jovens em curso de verão, com recurso ao *Scratch*, tendo os alunos exibido competências no final do curso em diversas áreas, incluindo programação orientada a eventos, reinicialização de estado, sincronização entre os elementos áudio e visual, bem como entre os objetos e algumas competências para criar animações complexas, que exigem a integração de vários conceitos, incluindo ciclos.

As técnicas de avaliação utilizadas caracterizam-se pela análise dos artefactos digitais produzidos pelos estudantes, completada com avaliações de projetos entre os alunos e notas de campo dos monitores caracterizando as opções de avaliação por uma "baixa sobrecarga de anotações escritas e sem recurso a testes ou a exames escritos." No caso da análise de projetos, uma grelha de avaliação foi criada e adotada para apoiar a avaliação dos projectos Scratch.

Outras tendências de cariz mais tecnológico têm sido desenvolvidas, como sejam a análise e avaliação automática dos blocos de programação ou ainda soluções automáticas em tempo real. Apresentamos alguns dos exemplos disponíveis de ferramentas de análise e avaliação automática de projetos criados através do Scratch.

O primeiro exemplo designa-se de "Scrape" (Figura 1), ferramenta que identifica e permite calcular os blocos de programação. Está em fase de desenvolvimento e disponível em http://happyanalyzing.com.

FIGURA 1 - ***Scrape*** **-** ferramenta de análise de blocos de programação

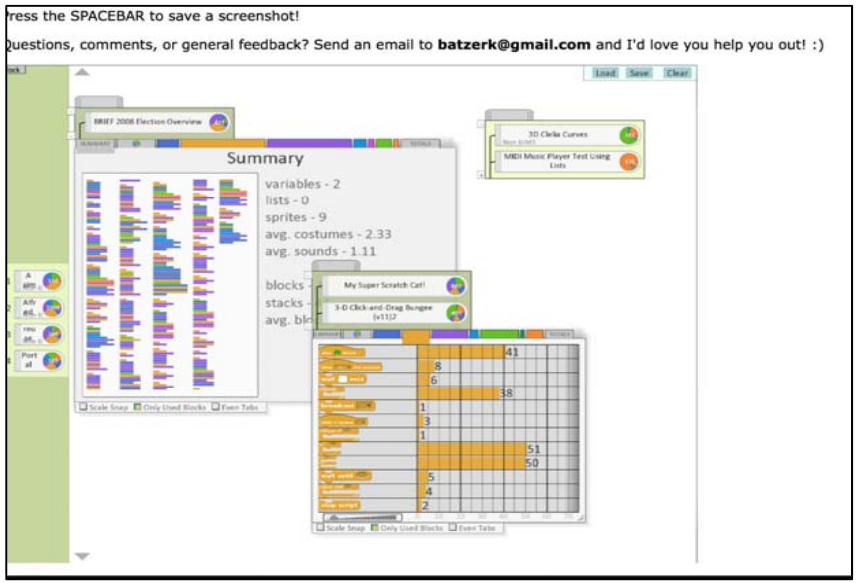
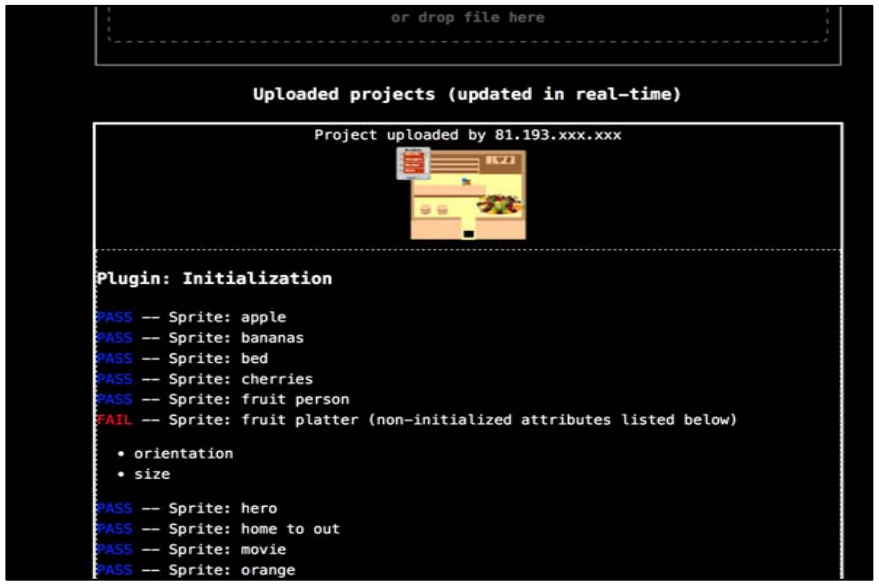


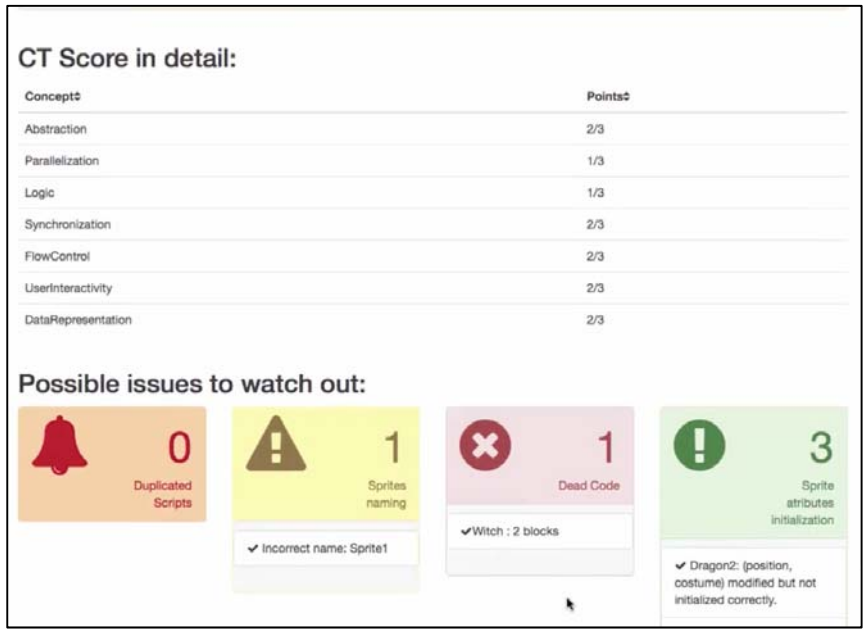
FIGURA 2 - ***Hairball*** **-** ferramenta de análise de blocos de programação *Scratch*



Um segundo exemplo é o "Hairball" (Figura 2), um plug-in que pode ser descarregado para o computador, disponível em https://github.com/ucsb-cs-education/hairball, incluindo uma demonstração online que pode ser acedida em http://hairball.herokuapp.com.

Finalmente um terceiro exemplo é o "Dr. Scratch " (Figura 3) que procura não apenas calcular blocos e identificar comandos mas analisar as sequências de programação inscritas em cada projeto e associar essas sequências e a sua conceção aos princípios do pensamento computacional (Moreno, 2014). Está disponível em http://drscratch.programamos.es .

FIGURA 3 - Princípios de pensamento computacional em Dr.Scratch



Esta ferramenta mostra-se bem mais complexa e apresenta desafios bem mais promissores do que as referidas anteriormente.

O programa associa diretamente o uso dos comandos a princípios do pensamento computacional. Na base da associação está uma matriz de conteúdos e uma escala de utilização dos comandos e blocos de programação, de um nível mais simples e básico a um nível mais avançado (Figura 4).

Um exemplo é a paralelização que, a um nível básico pode ser identificado com dois comandos de bandeira verde mas a um nível mais avançado pode implicar a comunicação entre sprites e sincronizar ações e eventos.

FIGURA 4 - Dr.Scratch - pensamento computacional e níveis de desempenho

Dr. Scratch

CT concept	Basic	Developing	Proficiency
Parallelization	2 scripts on green_flag	2 scripts on key_pressed, 2 scripts on sprite_clicked on the same sprite	2 scripts on when_I_recieve_message, create_clone, 2 scripts when_%s_is_>_%s, 2 scripts on when_backdrop_change_to
Synchronization	wait	Broadcast, when_I_receive_message, stop_all, stop_program,stop_programs_sprite	wait_until, when_backdrop_change_to, when_I_start_as_clon, broadcast_and_wait
Data representation	Modifiers of properties of sprites	Operations on vars	Operations on lists
Conditional logic	if	if_else	logic operations
Interactivity (UI)	Green_flag	key_pressed, sprite_clicked, ask_and_wait, mouse blocks	when_%s_is_>_%s, video, audio
Algorithmic notions of flow control	Sequence of blocks	Repeat, Forever	repeat_until
Abstraction and problem decomposition	> 1 scripts	> 1 scripts and > 1 sprites	def_block

A avaliação dos artefactos digitais criados, de forma manual ou automática e adicionando dispositivos complementares de recolha de informação que permitam uma avaliação compreensiva das aprendizagens realizadas e adquiridas pelos alunos, relativamente ao uso educativo de ambientes computacionais, parece ser a abordagem mais consensual na comunidade educativa.

Exemplos de técnicas que estão a ser associadas à análise dos projetos são as apresentações (incluindo a explicação e justificação oral), a justificação escrita dos projetos (através de relatórios), portefólios, entrevistas individuais ou em grupo, entre outras possibilidades.

O que é comum é a vontade de explorar novas formas e modelos de avaliação mais densos e compreensivos, assegurando uma multiplicidade

de fontes de informação com vista a tomar decisões no que diz respeito à aprendizagem dos alunos neste domínio.

A experiência de trabalho educativo desenvolvida e aqui apresentada, no que diz respeito à aprendizagem, tomou como referencial teórico o quadro conceptual proposto por Brennan & Resnick (2012) considerando as três dimensões principais: os conceitos de pensamento computacional (sequências, ciclos, processamento paralelo, eventos, condições, operadores, dados) as práticas computacionais (trabalhando de forma iterativa e incremental, testando, corrigindo e depurando, reutilizando e refazendo, abstraindo e modulando) e as perspetivas computacionais (expressando, ligando, questionando).

Recorde-se que este quadro conceptual foi o resultado de anos de observação e recolha de dados de diversa natureza por parte dos autores, incluindo observações, entrevistas e análise de projetos e aplicações Scratch disponíveis no local virtual do MIT destinado à comunidade Scratch a nível mundial. Com base neste conhecimento adquirido, o quadro tem sido utilizado para orientar os processos de introdução ao pensamento computacional em diferentes contextos educativos, com recurso ao ambiente Scratch (consta do guia curricular de apoio ao Scratch, disponível em várias línguas) e é um documento de referência para a comunidade educativa sendo utilizado como material de suporte à exploração do ambiente computacional.

O desafio de investigar os resultados obtidos nesta experiência à luz do quadro referido tornou-se também um fator de motivação adicional na medida em que teríamos assim a oportunidade de refletir de forma sistemática sobre o trajeto pedagógico percorrido levando em linha de conta um referencial teórico reconhecidamente consolidado mas ainda muito por explorar do ponto de vista das suas potencialidades investigativas. Este exercício permitiu, ao mesmo tempo, inventariar e delimitar novos territórios de investigação através do levantamento de novas perguntas e hipóteses de investigação, para trabalho futuro.

Tendo como referencial o quadro teórico e conceptual referido, o estudo tem como objetivo geral contribuir para o conhecimento e compreensão dos processos envolvidos nas iniciativas de introdução do pensamento computacional no âmbito da formação inicial de professores do ensino básico e secundário.

Tratando-se de um contexto de formação inicial de professores, importava, nesta investigação, conhecer a perceção que os futuros professores têm acerca da eventual relevância destas aprendizagens em função de cenários de aprendizagem no futuro, aqueles cenários que os futuros professores antevêem virem a constituir os seus contextos profissionais. Também se considerou importante analisar as perceções dos estudantes relativamente à utilidade e relevância deste tipo de processos formativos no seu percurso de formação profissional.

Levando em linha de conta o quadro conceptual referido, foram definidas as seguintes questões orientadoras:

- Que conceitos, práticas e perspetivas computacionais podem ter sido desenvolvidos durante a experiência de exploração pedagógica do ambiente computacional Scracth pelos estudantes- futuros professores?
- Qual a relevância atribuída pelos estudantes-professores ao desenvolvimento do pensamento computacional na escola e no currículo?
- Qual a importância atribuída à exploração do ambiente computacional Scratch como parte do processo formativo dos futuros professores?

2. METODOLOGIA

A intervenção educativa realizada com estudantes no contexto de um programa de formação inicial de professores do ensino básico e secundário foi definida como o campo empírico desta investigação. O contexto da investigação corresponde ao desenvolvimento do trabalho educativo das unidades curriculares de Tecnologia e Educação e TIC em Educação

oferecidas aos estudantes dos cursos de Licenciatura e de Mestrado em ensino (em diversos cursos) numa instituição de ensino superior.

A experiência aqui apresentada foi desenvolvida de acordo com os planos das respetivas unidades curriculares que prevê a realização de 15 sessões de 3 horas semanais durante o semestre, para o conjunto dos conteúdos programáticos previstos, para cada curso. Os estudantes trabalharam em pequenos grupos de alunos (2/3) para conceção e desenvolvimento dos seus projetos e aplicações.

A avaliação da aprendizagem para os tópicos das unidades curriculares referidos anteriormente, foi realizada a partir da abordagem proposta por Brennan & Resnick e implicou a conceção e o desenvolvimento dos projetos e aplicações com recurso ao Scracth seguida da apresentação e entrevista aos autores (artifact-based interview approach).

Neste caso, os estudantes desenvolveram e apresentaram em pares ou grupos de 3 estudantes, o projeto a toda a turma e através do diálogo e das perguntas abertas puderam responder às questões de outros colegas da turma. No total, 7 dos estudantes foram igualmente entrevistados pelos docentes que solicitaram a justificação das opções de conteúdo, pedagógicas e técnicas, desafios que o projeto podia receber para a sua melhoria, análise das potencialidades do projeto na sala de aula, dificuldades sentidas, entre outros aspetos.

Participaram na investigação 44 estudantes dos cursos de formação inicial de professores, sendo (26) do curso de educação básica e (18) do cursos de mestrado para os Ensinos Básico e Secundário (EBS) de Artes Visuais, Biologia e de Geologia, Educação Física, Matemática, Português e de Espanhol/Francês.

A totalidade dos estudantes envolvidos nesta intervenção são estudantes com antecedentes académicos fora das áreas da informática e da computação. Destes estudantes, só uma minoria tem algum conhecimento no domínio da programação e em especial na área de multimédia.

A investigação adotou um desenho de natureza qualitativa com recurso a entrevistas focus-group, análise de projetos e aplicações e análise documental (relatórios dos estudantes).

Uma abordagem compreensiva e interpretativa justifica-se pelo objetivo geral da investigação: contribuir para o conhecimento e compreensão dos processos envolvidos nas iniciativas de introdução do pensamento computacional no quadro da formação inicial de professores de modo a identificar estratégias que sejam bem sucedidas e que possam ser melhoradas e adotadas, bem como as condições que devem ser observadas para que essas iniciativas tenham bons resultados.

Para esta reflexão, invocámos igualmente o quadro de referência sugerido por Brenann & Resnick (2012) cujas dimensões-chave foram usadas para avaliar e refletir sobre a experiência de pensamento computacional vivida pelos estudantes.

Escolhemos este quadro para que nos ajudasse a pensar sobre a aprendizagem que os futuros professores estavam a desenvolver durante a exploração do ambiente Scratch mas também para nos ajudar a pensar sobre a importância/relevância que os futuros professores atribuem a estas aprendizagens, quer como parte dos seus processos formativos, quer para as crianças e jovens, seus alunos num futuro próximo.

Estas circunstâncias permitiram identificar áreas ou temas a explorar ao longo da realização do trabalho educativo e da investigação. As áreas identificadas são as seguintes:

- conceitos, práticas e perspetivas computacionais (enquanto dimensões-chave do pensamento computacional do quadro de referência);
- o pensamento computacional na escola e no currículo das crianças e jovens do ensino básico e secundário;

- o pensamento computacional nos processos formativos dos futuros professores.

Os dados são de natureza qualitativa e foram obtidos a partir de três fontes principais: os projetos e aplicações em Scracth, num total de 24 projetos, os relatórios dos estudantes, num total de 27 e as entrevistas focus-group, num total de 2 entrevistas a dois grupos de estudantes, num total de 7 alunos. Os projetos e aplicações e respetivos relatórios foram entregues pelos estudantes e arquivados na plataforma Moodle.

Os relatórios elaborados por grupos de estudantes relativamente ao seu projeto continham, entre outros, os seguintes elementos: a indicação dos autores, breve descrição do projeto, destinatários do projeto, a justificação (porque foi desenvolvido o projeto, qual a sua utilidade e qual a sua ligação ao currículo, quando aplicável). Se e quando houver ligação ao currículo, explicitá-la (disciplina, programa, conteúdos, competências transversais, etc.), a representação gráfica do algoritmo do programa desenvolvido ou descrição de como foi desenvolvido o programa, passo a passo, a programação – contagem de blocos ou comandos por categoria e as reflexões finais.

As entrevistas focus-group foram realizadas em dois momentos diferentes, depois de terminado o semestre e concluída a avaliação da unidade curricular. A entrevista foi conduzida por um dos investigadores a um grupo por cada curso sendo 4 de Mestrado em Ensino e 3 de Licenciatura, num total de 7 estudantes. As entrevistas foram gravadas em áudio e transcritas para um ficheiro de texto. As entrevistas foram conduzidas tendo como referência um guião previamente elaborado.

O guião foi organizado em três tópicos: experiências prévias de pensamento computacional, importância do pensamento computacional na escola e no currículo no EBS e pertinência do pensamento computacional na formação inicial de professores.

Durante as entrevistas as questões foram colocadas ao grupo de estudantes, sem ordem de intervenção. Cada estudante era livre de responder quando considerasse oportuno e poderia interromper ou dialogar com os colegas ou acrescentar informação.

A análise dos projetos e aplicações Scratch foi realizada considerando as dimensões identificadas por Brennan & Resnick: conceitos, práticas e perspetivas computacionais (Tabela 1). Para cada uma destas dimensões, as categorias de análise foram pré-definidas quer pelos comandos existentes no ambiente programação Scratch quer pelos resultados da investigação realizada por Brennan & Resnick (2011).

Os projetos foram analisados um por um, a partir dos *scripts* de programação concebidos pelos alunos, analisando, contando e agrupando os blocos por tipo de projeto, conceitos computacionais envolvidos e associando-os a princípios do pensamento computacional, igualmente propostos pelos autores referidos.

A presença de blocos de programação pertencentes às diferentes categorias de comandos permitiu identificar o tipo de projeto e o tipo de conceitos computacionais envolvidos nos diferentes tipos de projetos. O objetivo desta operação foi identificar os processos mentais usados pelos estudantes, "reconstituindo" o projeto desenvolvido, a partir da associação destes processos aos *scripts* de programação usados pelos estudantes. Foram utilizados os relatórios dos estudantes como fonte de informação relativamente aos processos usados na conceção e desenvolvimento dos projetos, particularmente úteis na investigação das práticas e perspetivas computacionais.

Os relatórios e as entrevistas foram objeto de análise de conteúdo, tomando como unidade de análise todas as frases com sentido, redigidas pelos estudantes incluídas nos relatórios e todas as frases com sentido ditas durante as entrevistas em grupo. Todo o conteúdo das respostas dos estudantes foi considerado no *corpus* de análise, depois de eliminados os ruídos de comunicação.

TABELA 1 - Matriz de dimensões, categorias de análise e indicadores

Temas/ Dimensões	Categorias de análise	Indicadores
Conceitos Computacionais	Categorias de comandos: Movimento, Aparência, Som, Caneta, Controlo, Sensores, Operações, Variáveis	Blocos de programação de cada categoria de comando presentes nos projetos Scratch.
Práticas computacionais	Incremental e interativo; Testando e depurando; Reutilizando e misturando; Abstração e modularização.	Expressões nos testemunhos (discurso escrito e falado): por exemplo: " fazer experiências", tentativa e erro", "descobrindo"
Perspectivas computacionais	Expressar, Conectar, Questionar	Expressões nos testemunhos (discurso escrito e falado): "Criar diálogo", "criar personagens", "recriar um evento", "criar um jogo", "movimentar", "escrever", "expressar", "comunicar ", "em conjunto", "Colocar questões" "explicar", "ensinar", "responder" , "fazer novas cores", "demonstrar, "interagir", "colegas"

3. RESULTADOS

Descrevem-se e analisam-se os resultados obtidos com referência à matriz apresentada na Tabela 1.

Na dimensão "conceitos computacionais", os tipos de projectos e a respetiva percentagem do total, foram os seguintes:

a) projetos de iniciação - projetos e aplicações que integram um número muito reduzido de comandos, bem como revelam limitada aplicação de princípios e conceitos computacionais (20,8%) ;

b) histórias interativas - projetos e aplicações que contém uma narrativa de conteúdo educativo com recurso a comandos que implicam interatividade da parte do destinatário como p.e. entrada de dados, opções de escolha no fio da história, etc. (16,7%);

c) *quizzes* - projetos e aplicações que contêm questões/ perguntas e utilizam as respostas dos destinatários na avaliação dos conhecimentos sobre o conteúdo em questão (29,2%);

d) simulações - projetos e aplicações que representam processos físicos, naturais ou comportamentos humanos (33,3%) .

No conjunto dos projetos e aplicações desenvolvidas pelos estudantes, os projetos que resultaram em *quizzes* e simulações foram em maior número do que as histórias interativas e os projetos de iniciação (Tabela 2).

No que diz respeito à aquisição dos princípios e conceitos computacionais por parte dos estudantes, os valores obtidos mostram que no processo de criação dos projetos Scratch, os estudantes usaram predominantemente os comandos de aparência e controlo e muito raramente os comandos de caneta (exceção para os estudantes de Artes Visuais). Recorde-se que comandos de aparência e controlo permitem manipulação dos trajes, controlo das personagens e de todas as ações de execução do programa.

TABELA 2 - Categorias de comandos por tipo de projetos (%)

	Movimento	Aparência	Som	Caneta	Controlo	Sensores	Operações	Variáveis
Histórias	9	33	8	0	42	1	3	4
Iniciação	12	31	6	1	42	1	2	6
Quizzes	1	26	10	0	28	12	12	11
Simulações	12	28	3	1	43	6	2	4

As histórias interativas implicaram o uso de comandos de som, de movimentos e algumas das histórias também recorreram a comandos de variáveis. Os projetos de iniciação fizeram uso dos comandos de movimento, para além dos comandos da aparência e de controlo.

Os projetos que se concretizaram em forma de *quizzes* fizeram uso significativo de comandos como os sensores (pela necessidade de obter respostas às perguntas, contagem de valores e de tempo, etc.), as operações (necessidade de uso de operadores lógicos, aritméticos, juntar, sortear aleatoriamente algarismos, etc.) e dos comandos de som (uso do som para perguntar, realizar narrativas e "oferecer" recompensas por acertos e sons desagradáveis para erros).

Finalmente, nas simulações criadas, para além dos comandos de aparência e de controlo, os estudantes fizeram uso significativo de comandos de movimento, para demonstração de gestos técnicos nos projetos dos estudantes de Educação Física e Desporto, e de sensores (para controlo de posições no palco, movimento de objetos, contagem de tempo, etc.).

Depois da análise por tipo de projetos e comandos utilizados, os projetos ainda foram organizados por tipos de projetos e princípios do pensamento computacional, cujos resultados podem ser observados na Tabela 3.

Os conceitos em maior evidência durante a conceção destes projetos foram as sequências, os eventos e o paralelismo. Esta tendência era esperada, uma vez que se trata de aspetos essenciais a todos os projetos: pensar numa ordem para executar as instruções dadas, criar eventos (qualquer instrução seguida de comando de controlo dá lugar a um evento, no palco) e uma grande maioria dos projetos exigia a execução de instruções em paralelo.

Os conceitos menos desenvolvidos foram as iterações ou ciclos, em quase todos os tipos de projeto mas também as condições, operadores e variáveis (com exceção dos *quizzes*).

TABELA 3 - Tipos de projetos e conceitos computacionais (%)

Tipos de projetos	Sequências	Ciclos	Eventos	Paralelismo	Condições	Operadores	Dados (vars)
Histórias	100	75	100	100	50	75	75
Iniciação	100	40	100	60	20	40	40
Quizes	100	14	100	86	100	100	86
Simulações	88	75	100	100	63	50	50

Se observarmos agora os conceitos desenvolvidos por tipo de projeto, podemos identificar alguns "padrões" (sendo o número de projetos muito reduzido, podem apenas ser interpretados no contexto deste estudo).

Os dados obtidos através da análise destes projetos mostram que os projetos de iniciação fazem um uso mais reduzido e limitado destes

conceitos e mostram que o uso de comandos, incidem sobretudo nos conceitos computacionais relacionados com as sequências, os eventos e o pensamento paralelo. Os projetos de iniciação usaram pouco os ciclos, os operadores e os dados e muito pouco as condições ("se" e "se senão", p.e.).

Os projetos de iniciação são projetos mais elementares e que representam estádios de aprendizagem do ambiente Scratch de escassa complexidade. Os comandos associados a ciclos, condições, operadores e dados ou variáveis são pouco utilizados.

As histórias interativas e as simulações constituem um estádio um pouco mais exigente do ponto de vista da exploração do programa e dos comandos mais complexos. Os comandos associados aos ciclos, bem como a condições, operadores e variáveis estão presentes em dois terços do total de projetos de histórias interativas e cerca de metade dos projetos das simulações.

Os projetos que incidiram em *quizzes* interativos foram os projetos em que os estudantes mais desenvolveram os conceitos computacionais incluídos no quadro de referência utilizado, com exceção dos ciclos, blocos de instruções que se poderiam repetir. Estes projetos fizeram um uso mais acentuado de condições, operadores e variáveis e menor uso das iterações ou ciclos.

No que diz respeito às práticas computacionais, invocando ainda o quadro de referência de Brennan & Resnick, os testemunhos dos estudantes escritos nos relatórios mostram que as práticas computacionais relacionadas com os processos incrementais e interativos e as constantes operações de teste e depuração, foram adotadas em todos os tipos de projetos.

Os valores referentes à reutilização de outros projetos e misturas são pouco significativos nas histórias e nos projetos de iniciação uma vez que os estudantes preferiam construir algo mais pessoal e original e não recorreram com grande frequência a projetos já existentes, ainda que estimulados para tal. As práticas de abstração e modularização foram mais significativas nos projetos de simulações e histórias interativas.

Na dimensão "perspetivas computacionais", a análise dos projetos e do conteúdo inscrito nos testemunhos dos estudantes quer nos relatórios quer nas entrevistas, permitiu " trazer ao de cima " os indicadores que funcionaram como sinais ou marcas das perspetivas computacionais desenvolvidas pelos alunos.

O contexto da sala de aula e o trabalho por pares constitui um ambiente propício ao desenvolvimento destas perspetivas, sendo comum a troca e a partilha de ideias, a discussão sobre os caminhos a percorrer em conjunto pelo grupo, favorecendo o aprender com os outros e para os outros. Os projetos e aplicações de tipo histórias interativas e as simulações proporcionaram com mais evidência a perspetiva assinalada por Brennan e Resnick, a conexão ou ligação com os outros, como pudemos verificar através dos testemunhos dos estudantes. Expressões como "trabalhar em conjunto", "colegas" ou o discurso escrito no plural, indicando algo que foi realizado em grupo, são marcas dessas perspetivas computacionais. Assim, nas histórias interativas e nos projetos de iniciação foi possível encontrar as marcas da perspetiva centrada na "Expressão" e também algumas marcas de "Conexão" .

Nos projetos desenvolvidos como *quizzes* a perspetiva desenvolvida foi a oportunidade de questionar, neste caso, as conexões ou interações entre os pares de estudantes na abordagem ao universo dos conteúdos e da aprendizagem e a forma como a computação pode inspirar este processo de "inventário dos saberes".

Da análise realizada aos documentos resultaram três dimensões que parecem constituir elementos importantes para compreender as perceções dos estudantes, relativamente à experiência de aprendizagem com recurso ao ambiente computacional Scratch: 1) reconhecimento da importância do pensamento computacional no quadro dos processos formativos; 2) a complexidade e a riqueza dos processos de aprendizagem neste tipo de

ambientes; 3) a importância de refletir e descobrir estratégias de trabalho em que seja possível abordar os conteúdos ou temáticas das áreas e disciplinas, com recurso a ambientes e ferramentas computacionais. Vejamos um breve resumo das contribuições dos estudantes.

A expectativa de mudanças aceleradas nos cenários de aprendizagem que irão enfrentar e a noção de que os mais novos estão bastante mais familiarizados com a tecnologia e com a cultura digital do que eles próprios são as razões porque consideram que devem estar bem preparados para enfrentar estes desafios. Esta expectativa de mudança faz ter presente a perceção da necessidade de se adaptarem a novas metodologias e ambientes de ensino e de aprendizagem que certamente estarão povoados de tecnologia. Este tipo de atividades deve por isso fazer parte integrante dos seus processos formativos. Por exemplo, A1 refere que " (…) *foi uma mais-valia para mim e para o meu futuro, como futura professora… porque nos vai ser bastante útil, visto que o futuro está todo ligado com a tecnologia"*.

Os estudantes percebem a complexidade dos processos formativos na aprendizagem dos princípios e das práticas de pensamento computacional. Por exemplo, um estudante refere que "*Falando agora no processo de aprendizagem com o Scratch, inicialmente sentimos muitas dificuldades no manuseamento do programa pois era uma realidade completamente nova, mas ao longo do tempo e com a ajuda de colegas e professores conseguimos desenvolver essas capacidades, fazendo com que compreendêssemos a sua utilidade, e o seu aproveitamento da melhor forma para o futuro profissional*".

Os estudantes, enquanto futuros professores, atribuem predominância aos conteúdos educativos nas suas conceções de ensino e aprendizagem e tais perceções refletem-se no conteúdo dos projetos desenvolvidos. "*Este projeto consiste num jogo didático, onde é abordada a área da matemática com vista a desenvolver os conteúdos de números e operações e geometria …O conteúdo foi pensado com base no programa de matemática* (EB 07). Num outro registo, os estudantes referem que "*os conteúdos abordados são os reis e navegadores e os fatos associados à expansão marítima, bem como as datas. Pretendemos que as crianças adquiram este conhecimento"*. (EB11).

4. CONCLUSÕES

Retomando o conteúdo das questões orientadoras desta investigação tendo como base o conhecimento e as evidências recolhidas durante o trabalho empírico, propomos as seguintes conclusões, naturalmente limitadas ao contexto desta investigação.

Para a primeira questão orientadora: que conceitos, práticas e perspetivas computacionais podem ter sido desenvolvidos durante a experiência de exploração pedagógica do ambiente computacional Scracth pelos estudantes-professores?

Os conceitos computacionais envolvidos com maior frequência na criação dos projetos foram as sequências, os eventos e o pensamento paralelo. Os conceitos menos envolvidos foram as iterações ou ciclos, as condições, operadores e variáveis.

A utilização de categorias de comandos e de dimensões-chave do pensamento computacional parecem poder relacionar-se, embora com algumas cautelas, com a complexidade e exigência dos projetos criados: projetos mais simples, recorrem aos comandos de eventos da categoria de controlo e à aparência (combinados com som e movimento, em muitos projetos) e estimulam o desenvolvimento de dimensões-chave como a criação de sequências, os eventos e o pensamento paralelo. Os projetos mais avançados e complexos recorrem com maior predominância aos comandos das categorias de sensores, operações e variáveis, envolvendo e combinando com ciclos e condições.

As práticas computacionais identificadas por Brennan & Resnick, mostram que os processos de criação de projetos com recurso ao Scratch,

são desenvolvidos de forma incremental e interativa, com sucessivas tentativas e correções de erros.

Apesar de uma boa parte dos estudantes ter acabado por desenvolver novos projetos, numa primeira fase, quase todos os projetos passaram pela "desconstrução" doutros e aproveitamento de alguns elementos (reutilização e combinação de código, por exemplo) para criar um projeto que já tinha algo de diferente em relação ao original.

Também os processos de abstração e modularização foram observados ao longo da criação de projetos, sobretudo naqueles de maior envergadura considerando que os estudantes têm um "background" fora das áreas da computação e não têm nas suas rotinas de pensamento, os dispositivos de trabalho cognitivo próprios das áreas das ciências da computação, enfrentando por isso maiores dificuldades (Yadav, 2011, Fesakis, 2009).

Relativamente às perspetivas computacionais, desenvolvidas durante os processos de criação de projetos, foram visíveis evidências nos testemunhos escritos dos estudantes que mostram que os projetos constituíram um meio de expressão, conexão com os outros e de interrogação do mundo e da realidade, com maior ênfase nos conteúdos disciplinares dos cursos, na forma como foram concebidos e desenvolvidos.

A segunda questão orientadora procurava esclarecer o papel do papel que os futuros professores atribuem ao pensamento computacional na Escola e à sua relevância nos seus próprios processos formativos.

A introdução de atividades educativas na escola e no currículo (em contextos formais, não formais ou informais) baseadas no pensamento computacional, nos testemunhos dos futuros professores, pode constituir um fator motivacional e uma mais-valia para crianças e jovens, através da adoção de recursos, ferramentas e ambientes computacionais que explorem o potencial de estratégias e que utilizem o pensamento computacional como estímulo à curiosidade, à experimentação, à colaboração e interação social, à resolução de problemas e à aprendizagem de uma linguagem e de uma gramática fundamental no futuro das crianças e dos jovens.

Apesar do entusiasmo, os estudantes mantêm uma perspetiva muito centrada nos conteúdos e nos processos da sua disciplina específica, procurando formas e estratégias de complemento entre os saberes e saberes-fazeres próprios da sua área e as possibilidades conferidas pelo pensamento computacional, nessa área específica. A conceção do conteúdo e do próprio ato de ensino são determinantes para uma maior ou menor abertura, relativamente à integração do pensamento computacional nas áreas específicas. Uma perceção do conteúdo disciplinar como um conjunto de saberes mais ou menos estáticos e a perceção do papel do professor como um transmissor desses conteúdos aos seus alunos, pode não facilitar uma atitude mais aberta para integrar, no futuro, abordagens bem mais centradas nas capacidades e processos de pensamento dos alunos.

Assim, a adoção de tecnologias e as propostas baseadas no pensamento computacional, neste caso através da criação de projetos Scracth, acabam por ser desenvolvidas e concretizadas por estes estudantes nos conteúdos e saberes das suas áreas (Matemática, Artes Visuais, Espanhol, Educação Física, por exemplo). Foram raros os projetos que se destinariam a mais do que uma disciplina.

No futuro, estas propostas deverão integrar-se na cultura pedagógica e didática específica da disciplina a que cada estudante está vinculado a contribuir para a inovação educativa no quadro de cada disciplina.

Ficará por aprofundar esta perspetiva de combinação entre os conteúdos e as ferramentas computacionais que, nesta fase, pode ter constituído um desafio demasiado grande para quem está ainda em fase de formação científica e pedagógica.

Os resultados deste estudo parecem reforçar as ideias de que a programação *deve ser para todos e não apenas para os envolvidos nos cursos de computação* (Wing, 2006). Também as propostas realizadas com

recurso ao Scrath no contexto da formação inicial de professores podem ter influência positiva nas perceções dos estudantes quanto ao potencial deste ambiente no trabalho com os futuros alunos (Yadav, 2011, Fesakis, 2008). Para além disso, as propostas educativas centradas nos processos de pensamento (em vez de propostas centradas exclusivamente nos aspetos técnicos da programação), poderão ajudar os estudantes, futuros professores; a desenvolver capacidades de adaptação aos diferentes contextos de ensino e aprendizagem que podem ter que enfrentar no futuro (Kim, 2012).

A possibilidade de analisar os processos de pensamento computacional no decurso da utilização do Scracth, com um maior grau de profundidade, requer agora a conceção e validação de um instrumento de avaliação formativa que permita, a partir da base de indicadores levantada no campo empírico, um acompanhamento e monitorização do processo.

O instrumento deverá incluir, para além das dimensões do pensamento computacional, as dimensões relativas à pedagogia e aos conteúdos, científicos e técnicos ou artísticos envolvidos nos cursos de proveniência dos estudantes, futuros professores.

A introdução do pensamento computacional na escola e no currículo constitui um desafio a todos os agentes educativos e em todos os planos do sistema educativo. A formação dos professores, inicial e contínua, é certamente um dos planos de maior relevo.

A introdução do pensamento computacional na formação dos futuros professores tem sido justificada pelo facto dos professores poderem mais facilmente integrar o pensamento computacional no currículo quando reconhecem e identificam os benefícios do pensamento computacional na resolução de problemas e o interpretam num contexto significativo (relacionado com a experiências pessoais) e conseguem identificar as estratégias de trabalho educativo para integrar o pensamento computacional na sua própria disciplina (Zhou, 2011).

A experiência educativa a que se refere o texto, foi encarada por estes estudantes como uma experiência positiva, útil no quadro da sua própria cultura digital e correspondeu a uma primeira fase de aprendizagem, nestes domínios. A necessidade de se adaptarem e de se prepararem para novos cenários de aprendizagem permitiu uma perceção desta experiência como uma mais-valia no seu processo formativo.

No entanto, a introdução do pensamento computacional nos processos formativos, no quadro da formação inicial, tem exigências e que decorrem de diversos fatores, entre eles, certamente o fato dos estudantes não pertencerem a domínios das ciências da computação. Esta dificuldade tem constituído um obstáculo importante, e em particular a dificuldade de encontrar exemplos de ensino de pensamento computacional em cada uma das áreas curriculares envolvidas (Zhou, 2011).

A aprendizagem de uma nova linguagem e de um novo código correspondem a um esforço muito exigente para os estudantes, futuros professores.

O desenho das iniciativas de introdução do pensamento computacional no âmbito da formação inicial de professores, em vários formatos e modalidades, deverão permitir o equilíbrio entre o conteúdo, a pedagogia e a tecnologia.

A experiência analisada deixa à vista a importância de apresentar propostas de trabalho com recurso à tecnologia mas que se possam integrar no campo dos conteúdos e das disciplinas e adotem as abordagens pedagógicas adequadas, estabelecendo uma relação equilibrada e de mais valia para os processos de ensino e aprendizagem.

Importa destacar a importância de introduzir o pensamento computacional nos programas de formação inicial de professores do ensino básico e secundário através da exploração de ferramentas e ambientes computacionais adequados a crianças e jovens em que uma parte do

programa diga respeito ao universo da cultura científica, técnica ou pedagógica de uma determinada área disciplinar.

É igualmente importante reservar um tempo necessário para permitir que os alunos possam progredir de uma fase mais elementar, criando projetos simples, para uma fase mais avançada, criando projetos inovadores, ou seja, possam explorar e consolidar as dimensões-chave do pensamento computacional, de maior poder computacional e com mais profundidade. Ficou igualmente à vista a necessidade de criar um espaço para que os estudantes possam realmente conhecer e explorar as funcionalidades disponíveis num plano mais transversal e multidisciplinar, dando tempo para que as suas próprias conceções de conteúdo, de ensino e de aprendizagem possam evoluir, de uma perspetiva mais "mecânica" para uma perspetiva mais ágil, incorporando conceitos e práticas mais consistentes com abordagens à aprendizagem conetivistas e construtivistas, que permitam explorar mais a fundo, o verdadeiro potencial das tecnologias e dos computadores.

Não havendo lugar a um programa adequadamente desenhado, nem lugar a um tempo e um espaço próprios, a perspetiva de "transmissão dos conteúdos" dificilmente dará lugar à perspetiva de "construção do conhecimento", exigindo professores facilitadores da aprendizagem, professores que estimulem a criatividade e a curiosidade dos seus alunos e que fazem da prática pedagógica um espaço de inspiração e exploração das potencialidades do pensamento computacional. Em última análise, um professores com os olhos no futuro dos seus alunos.

Este trabalho levanta questões de maior profundidade ao questionar a necessidade de reinventar um novo paradigma com repercussões quer no contexto da formação dos futuros professores quer no ensino da informática no ensino básico e secundário. Este paradigma implica uma orientação mais acentuada e explícita para o desenvolvimento do pensamento computacional na Escola, no currículo e na aprendizagem, mais conforme aos problemas e necessidades das sociedades contemporâneas, formando uma nova geração de professores que desenvolvem uma perspetiva mais atualizada e aberta da informática no ensino e na educação.

5. REFERÊNCIAS BIBLIOGRÁFICAS

ACM/CSTA (2010) Running On Empty The Failure to Teach K–12 Computer Science in the Digital Age. Disponível em http://www.acm.org/Runningonempty/

Bardin, L. (1977) *Análise de conteúdo*. Edições 70.

Barr, B. V. (March, de 2011). Bringing Computational Thinking to K-12 : What is Involved and What is the Role of the Computer Science Education Community ? ACM, *Inroads*, pp. Vol.2, N1, pp. 48-54.

Bell, T. (2011). Ensinando Ciência da Computação sem o uso do computador. Computer Science Unplugged ORG.

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. Paper presented at annual American Educational Research Association meeting, Vancouver, BC, Canada.

Charlton, P. (2012). Time to reload? Computational Thinking and Computer Science in Schools. What researches says? Briefing 2. . London Knowledge Lab - Institute of Education, University of London.

CSTA - Computer Science Teachers Association & Machinery, A. f. (2012). Computer Science K–8: Building a Strong Foundation. *Jornal of Computer Science Teachers Association.*

Cuban, L. (1993). Computers Meet Classroom: Classroom Wins. *Teachers College Record,* Volume 95 Number 2, p. 185-210.

Denning, P. (June de 2009). The profession of IT. Beyond computational thinking. *Points of View*, pp. 28-30.

Fesakis, G., & Serafeim, K. (2009). Influence of the familiarization with "scratch" on future teachers' opinions and attitudes about programming and ICT in education. *ACM SIGCSE Bulletin*, 41(3), 258. doi:10.1145/1595496.1562957

Franklin, D. Conrad, P., Boe, B. Nilsen, K., Hill, C., Len, M., Dreschler, G., Aldana, G., Almeida-Tanaka, P., Kiefer, B., Laird, C., Lopez, F., Pham, C., Suarez, J. and Waite, R. (2013). Assessment of computer science learning in a scratch-based outreach program. In *Proceeding of the 44th ACM technical symposium on Computer science education (SIGCSE '13*). ACM, New York, NY, USA, 371-376. http://doi.acm.org/10.1145/2445196.2445304.

Grover, S. & Pea, R. (2013) Computational Thinking in K–12: A Review of the State of the Field Shuchi Grover1 and Roy Pea1. *Educational Researcher*, Vol. 42 No. 1, pp. 38–43 .

Guzdial, M. (2008). Education: Paving the way to computational thinking. *Communications of the ACM*, pp. Vol. 51,n 8.

He, W. (2012). A Computational Thinking-oriented Online Lesson Planning System, 7(1), 27–30. *Ieee Technology And Engineering Education (ITEE)*, *V*OL. 7, (1) p.27-30.

Hutchison, D., & Mitchell, J. C. (2008). *Informatics Education – Supporting Computational Thinkin*g. Third International Conference on Informatics in Secondary Schools – Evolution and Perspectives, ISSEP 2008 Torun, Poland, July 1-4, 2008 Proceedings

Kim, H. (2012). Enhancing teachers ' ICT capacity for the 21st century learning environment cases of teacher education in Korea. *Australasian Journal of Educational Technology*. pp. 28(6) 965–982.

Lye, S.Y & Koh, J.H.L. (2014) Review on teaching and learning of computational thinking through programming: What is next for K-12*?*. *Computers in Human Behavior*. Volume 41, December 2014, Pages 51–61.

Marques, M.T.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. Universidade de Lisboa.

Moreno, J., Robles, G. e Chusig, C. (2014) Dr scratch, Automatic analysis of Scratch projects to assess the development of CT. MIT, Scratch Conference, Boston.

National Research Council (2010). Report of a Workshop on the Scope and Nature of Computational Thinking. *The National Academies Press.*

Philips, P. (2008) Computational Thinking: A Problem-Solving Tool for Every Classroom. Disponível em: http://education.sdsc.edu/ resources/CompThinking.pdf

Resnick, M., et. al. (2009). Scratch: Programming for all. *Communications of the ACM*, 60-68.

Royal Academy of Engineering. (2012). Shut down or restart? The way forward for computing in UK schools. Royal Society.

Science Business. (2007). The Science of Thinking. Europe's next Policy Change. Brussels: Microsoft Corporate.

Selby, C, Dorling, M. Woolalard, J. (2014) Evidence of Assessing Computational Thinking . Original do autor, 1-11. Disponível em : http://eprints.soton.ac.uk/id/eprint/372409 . Acesso 30 dezembro de 2104.

United Nations Educational, Scientific and Cultural Organization (2005). *Toward knowledge societies*. UNESCO World Report. Conde-sur-Noireau, France: Imprimerie Corlet.

Werner, L., Denner, J., Campe, S. (2012). The Fairy Performance Assessment: Measuring computational thinking in middle school, Proceedings of Special Interest Group in Computer Science Education, Feb. 29 – Mar. 3, Raleigh, N. Carolina, USA

Wing, J. (2006). Computational thinking. *Communications of the Association for Computing Machinery*, pp. 152-155.

Wing, J. (January de 2008). Five Deep Questions in Computing. *Communications Of The ACM*, pp. Vol. 51, No. 1.

Wing, J. (2008a). Computational Thinking. *CACM Viewpoin*t, 33-35. Obtido de: http://www.cs.cmu.edu/~wing/ .

Wing, J. M. (2008b). Computational thinking and thinking about computing. *Philosophical transactions*. Series A, Mathematical, physical, and engineering sciences, pp. 366(1881), 3717–25.

Wing, J. M. (2011). Computational Thinking—What and Why? *The Link* - Carnegie Melon School of Computer Science. Disponível em : http://link.cs.cmu.edu/article.php?a=600.

Wing, J., (2014) "Computational Thinking Benefits Society". Social Issues in Computing. New York: Academic Press. Artigo disponível e consultado em: Socialissues.cs.toronto.edu., 2014.

Yadav, A. L. (2011). Introducing Computational Thinking in Education Courses. *SIGCSE11*. Dallas, USA: ACM.

Yadav, A., Mayfield, C., Zhou, N., Hambrusch, S., and Korb, J. T. (2014). Computational thinking in elementary and secondary teacher education. *ACM Trans. Comput. Educ.* 14, 1, Article 5 (March 2014), 16 pages. DOI:http://dx.doi.org/10.1145/2576872

Zhou, N. W. (2011). Computational Thinking In Pre-Service Teacher Education : Integrating CT. *Technology And Teacher Education Annual*, 1 (pp. 62-63). Associationn for the Advancement of Computing in Education.

The future teachers and the teachers of the future. Challenges from the introduction of computational thinking into the school, curriculum and learning

Abstract:

The text describes and analyses the introduction of computational thinking in the context of initial teacher education in a university context through the use of Scracth programming language. The approach is based in theorethical and practical framework in order to study and evaluate the development of computational thinking. We designed a study aimed to investigate the adequacy of the dimensions present in the referred framework, in the context of initial teacher training, as well as evaluating the attributed relevance to the development of computational thinking by future teachers as part of their training process. A total sample of 44 students participated in the research selected from undergraduate and master's courses. A qualitative research methodology was adopted, using Scratch programming language and multimedia projects, reports and focus-group interviews as main data collection techniques. Results shows the importance of providing experiences and learning opportunities to the initial teacher training students, appropriate to the development of computational thinking so that students can, as future professionals, to prepare for an increasingly demanding and complex society and to take fully advantage of the educational potential of computers and particularly the computacional environments for children.

Keywords: computational thinking; initial teacher education; and teaching computing**.**

Texto:

- Submetido: outubro de 2014.
- Aprovado: novembro de 2014.

Para citar este artigo:

Ramos, J. L., & Espadeiro, R. G. (2014). Os futuros professores e os professores do futuro. Os desafios da introdução ao pensamento computacional na escola, no currículo e na aprendizagem. *Educação, Formação & Tecnologias*, 7 (2), 4-25 [Online], disponível a partir de http://eft.educom.pt.

Notas biográficas dos autores

[i] **José Luís Pires Ramos** é Doutor em Ciências da Educação pela Universidade de Évora e Mestre em Tecnologia Educativa pela Universidade de Salamanca. É atualmente Professor Associado na Universidade de Évora e investigador do Centro de Investigação em Educação e Psicologia.

[ii] **Rui Gonçalo Espadeiro** é licenciado em Ensino de Matemática pela Universidade de Évora e pós-graduado em Educação Matemática pela mesma universidade. Atualmente está destacado no Centro de Competência TIC da Universidade de Évora.