

# PENSAMENTO COMPUTACIONAL NA ESCOLA E PRÁTICAS DE AVALIAÇÃO DAS APRENDIZAGENS. UMA REVISÃO SISTEMÁTICA DA LITERATURA.

*José Luís Ramos*

*Centro de Investigação em Educação e Psicologia da Universidade de Évora*

*Rui Gonçalo Espadeiro*

*Centro de Competência TIC da Universidade de Évora*

**Resumo:** Nos últimos anos assistimos à multiplicação de iniciativas e programas de introdução do pensamento computacional na escola em diversos países, através do recurso a diferentes tecnologias, linguagens e ambientes computacionais. Com recurso a uma revisão sistemática da literatura relativa às práticas de avaliação das aprendizagens, os autores sublinham, nesta comunicação, a importância de criar e desenvolver estratégias de avaliação das aprendizagens dos estudantes, apropriados a este tipo de ambientes computacionais, no quadro do conteúdo das propostas de trabalho educativo neste domínio.

O conhecimento das práticas de avaliação predominantes nas intervenções neste domínio, no nosso país bem como a identificação de instrumentos de avaliação adequados à avaliação do pensamento computacional pode ser um contributo importante para o desenho das propostas de avaliação por parte dos professores do ensino básico e secundário para que estes possam acompanhar as iniciativas de promoção do pensamento computacional na escola, em diferentes formatos e percursos, contribuindo para uma avaliação clara e compreensiva dos seus impactos na aprendizagem.

**Palavras-chave:** *pensamento computacional, avaliação das aprendizagens, ambientes computacionais .*

**Abstract:** In recent years we have seen the multiplication of initiatives for the introduction of computational thinking in school programs in many countries, by the use of different technologies, languages and computing environments. In this communication the authors wish to underline the importance in creating and developing of student evaluation's learning strategies, appropriate to this computing environments, in the framework of the content of the proposal. The knowledge of the prevailing teacher's assessment practices in interventions in this field in our country and the identification of appropriate assessment tools to evaluate the computational thinking may be an important contribution to the design of assessment strategies by teachers of primary and secondary education so that they can keep up with initiatives to promote computational thinking in school, in different shapes and paths, contributing to a clear and comprehensive assessment of its impact on learning.

**Keywords:** *computational thinking, learning assessment, computational environments.*

## **Introdução**

O panorama curricular no que diz respeito ao ensino e à aprendizagem das TIC está a ser profundamente alterado, no seu conteúdo e na sua forma. Entre os muitos “ventos da mudança” destacamos a emergência de movimentos na sociedade que reclamam da escola e das instituições educativas, não uma mudança de cosmética, na forma de uma nova abordagem ou uma nova tecnologia ou aplicação, mas pretendem algo mais fundo e duradouro: uma mudança de paradigma de aprendizagem.

Estes movimentos podem ser observados a partir da informação disponível sobre iniciativas de vários governos e da sociedade civil, em relatórios acerca da introdução do pensamento computacional nas escolas, livros e artigos científicos, iniciativas de disseminação e de ensino informal de linguagens de programação, desafios e concursos, artigos de opinião nos media nacionais e internacionais, publicações, redes sociais, etc.

O debate centra-se sobretudo no desfasamento entre o que é ensinado na escola, neste domínio (quando existe) e o que parecem ser as necessidades sociais, económicas e culturais das sociedades industrializadas e desenvolvidas, em que o papel da tecnologias ganha uma maior relevância. O perfil da mão de obra está igualmente a mudar. Este desfasamento faz-se sentir de forma mais visível na falta de profissionais nestas áreas, não só a longo mas também a curto e a médio prazo (EC, 2014). A falta de candidatos com apetência para as áreas da computação, ciências, tecnologias, engenharia e matemática (CTEM) durante as trajetórias de educação e formação no ensino secundário e posteriormente no ensino universitário fazem sentir-se com intensidade e constituem hoje motivo de preocupação. E, para além de outros aspectos certamente a levar em consideração, o ensino das TIC parece não estar a responder de forma apropriada a esta situação. Uma profunda mudança nos perfis profissionais implicaria uma profunda alteração ao currículo.

Trata-se por isso de uma mudança que interroga o que é ensinado, o que se aprende, como se aprende e como se avalia o que é aprendido, nos espaços curriculares onde são ensinadas as disciplinas de TIC, programação e informática nos ensinos básico e secundário.

Estes movimentos são protagonizados quer pelas autoridades de educação (centrais, regionais ou locais), incluindo os governos, em vários países quer por instituições de natureza muito diversa: académicas, empresariais, científicas, sociais e culturais. Estes movimentos propõem mudanças de grande envergadura, através da substituição, total ou parcial, conforme os casos, dos conteúdos curriculares tradicionalmente incluídos nestas áreas bem como um alargamento da presença da computação a novos espaços curriculares, ao longo de todo o tempo de escolaridade de crianças e jovens.

Nos casos mais disruptivos é colocado o ênfase na ciência da computação tornando central, do ponto de vista dos conteúdos, o ensino do código ou programação.

Em maior ou menor escala, com maior ou menor grau de mudança, este movimento vai estimulando a introdução do ensino do código e da programação nos currículos do ensino básico e secundário, nos sistemas de ensino, um pouco por todo o mundo, com destaque para o contexto europeu onde o nosso país se integra.

Na Europa, o Reino Unido tomou a dianteira, removendo a disciplina de TIC e fixando uma nova disciplina no currículo, no que foi seguido por iniciativas em França e Espanha mas também na Alemanha, Bélgica e Holanda. Mais recentemente no nosso país foi anunciada uma iniciativa piloto para a introdução da programação no 1º ciclo (Neven, 2015).

No essencial, é proposto um alargamento e um aprofundamento do conhecimento e das capacidades dos jovens no domínio das tecnologias reforçando a componente da computação, e em particular a programação, até agora inexistente ou com muito pouca expressão curricular e sublinhando a importância, para o desenvolvimento das crianças e jovens, de aprender a “desenhar, criar e combinar e não apenas navegar, conversar e interagir” sustenta Resnick (2009).

Tratam-se de um movimento pouco definido na forma como concretiza as propostas no terreno da escola, sendo possível identificar perspectivas que sustentam o ensino da programação, por si só, como uma área suficiente para o desenvolvimento do pensamento do computacional, enquanto outras perspectivas têm preferência por uma maior abertura e sustentam que a computação inclui a programação, mas também as tecnologias de informação e neste sentido ampliam as suas possibilidades de intervenção educativa.

O que parece ser comum às diferentes abordagens é a adopção de uma grande diversidade de tecnologias e suportes lógicos, permitindo a criação e o desenvolvimento de inúmeras perspectivas ao desenvolvimento do pensamento

computacional, quer seja através de linguagens de programação específicas para crianças e jovens quer seja através de jogos ou de robótica, análise de dados, etc.

Em Portugal, um número significativo de escolas do ensino básico e secundário tem vindo a desenvolver iniciativas e projectos com o objectivo de introduzir o pensamento computacional. As mais recentes iniciativas por parte do MEC são o projeto-piloto “*Iniciação à Programação no 1.º Ciclo do Ensino Básico*” e o inventário e apoio aos *Clubes de Programação e Robótica* existentes no país.

Este texto tem como objectivo analisar e discutir práticas de avaliação no domínio da introdução do pensamento computacional na escola que possam ajudar a compreender a forma como tem sido abordada a avaliação das aprendizagens em contextos educativos permeados pelos ambientes computacionais.

Através de uma revisão da literatura seletiva e não exaustiva foram identificadas as práticas de avaliação predominantes neste campo bem como identificados alguns instrumentos e técnicas de avaliação destinados a avaliar as aprendizagens dos alunos, neste caso, especificamente destinadas ao ambiente computacional Scratch, um dos mais populares no âmbito do movimento acima referido.

### **Avaliação das aprendizagens em ambientes computacionais**

O contexto brevemente descrito de multiplicação de intervenções educativas centradas no desenvolvimento do pensamento computacional, através da introdução da computação nas escolas, para além de outras questões (por exemplo, a formação dos professores), levanta a necessidade prática da avaliação das aprendizagens nestes contextos. E é neste aspecto que concentramos a nossa atenção, considerando este tema é um elemento crítico em processos desta natureza.

Como avaliar as aprendizagens decorrentes da exploração pedagógica de ambientes computacionais? Sem atenção à avaliação, o pensamento computacional não terá muita probabilidade de seguir o caminho de sucesso em qualquer currículo (...) 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).

Recorde-se que as propostas de pensamento computacional são motivo de acesso debate (Wing, 2006, 2008, 2008b, 2009, 2011, 2014; Yadav, 2011, 2014) e é um construto bastante amplo na medida em que reflete, no seu conteúdo, a diversidade de conceitos, teorias, modelos e áreas de prática de um campo extenso como é o

campo das ciências da computação. Se acrescentarmos as inúmeras ferramentas computacionais disponíveis, destinadas especificamente ao uso por crianças e jovens e que podem, cada uma à sua maneira, ser usadas para criar diferentes contextos de aprendizagem e estimular diferentes formas de aquisição e desenvolvimento dos conceitos e princípios do pensamento computacional (*Scratch, Kodu, Blockly, Alice, Squeak, Angrybirds, Minecraft* entre outros), compreenderemos a complexidade do empreendimento e as dificuldades que se colocam aos que procuram envolver-se neste tipo de trabalho.

Werner (2012) reforça a importância da necessidade de desenvolver conceitos, teorias, modelos e dispositivos de avaliação apropriados, ao observar que “um factor que limita a adopção do pensamento computacional no ensino secundário é a falta de avaliações” (ACM/CSTA, 2010; Werner, 2012).

A emergência de programas e iniciativas destinadas ao desenvolvimento do pensamento computacional tem envolvido, no caso das melhores práticas, a definição do rationale que suporta e justifica a “entrada na escola” e no currículo desta área de estudo, através da explicitação do conteúdo da(s) proposta(s) incluindo as metas educativas a alcançar por nível de escolaridade através da introdução desses ambientes e ferramentas computacionais bem como orientações curriculares para as escolas, professores e famílias, acerca do suporte técnico e de informação disponível, materiais de apoio e avaliação das aprendizagens.

Esta definição e explicitação do conteúdo das propostas de introdução ao pensamento computacional na escola, do seu rationale e dos seus referenciais de avaliação é um trabalho essencial que permitirá alcançar a sustentabilidade das iniciativas implementadas neste domínio. Para isso é igualmente essencial o envolvimento ativo da sociedade, através das instituições, envolvimento dos atores dos campos económico, cultural, académico e científico, da comunidade educativa e das famílias numa construção colectiva e partilhada de um novo paradigma que permita pensar o futuro.

Este quadro de grande amplitude e complexidade no que diz respeito às possibilidades de trabalho educativo no âmbito do pensamento computacional, implica por isso a adopção de referenciais curriculares e de avaliação que acompanhem as iniciativas e possam apoiar as intervenções a realizar, incluindo a avaliação dos seus impactos.

Neste caso específico, o referencial deve refletir e justificar a introdução do pensamento computacional na escola e no currículo a partir das necessidades sociais, culturais, científicas, tecnológicas e económicas identificadas e que justificam uma resposta da escola, desta natureza e desta amplitude, com base numa visão para o que se pretende no futuro. É o processo de construção da visão que dará lugar à construção dos referenciais curriculares e de avaliação.

A partir destes instrumentos de regulação do currículo, é possível iniciar e fazer caminho “Uma vez definido de forma adequada o conceito, os instrumentos apropriados de avaliação das aprendizagens podem então ser desenvolvidos.” (National Science Foundation Research Report, 2010).

No caso do nosso país, não dispomos de referencial para além daquele que diz respeito e apenas às metas curriculares TIC do 7º e 8º . Será por isso este referencial que será usado, neste texto, para refletir sobre as práticas de avaliação em ambientes computacionais.

Vejamos com mais detalhe os aspectos inscritos no ponto P8 – Produção das metas curriculares TIC do 7º e 8º ano:

Exploração de ambientes computacionais

1. Criar um produto original de forma colaborativa e com uma temática definida, com recurso a ferramentas e ambientes computacionais apropriados à idade e ao estágio de desenvolvimento cognitivo dos alunos, instalados localmente ou disponíveis na Internet, que desenvolvam um modo de pensamento computacional, centrado na descrição e resolução de problemas e na organização lógica das ideias.

1. Identificar um problema a resolver ou conceber um projeto desenvolvendo perspetivas interdisciplinares e contribuindo para a aplicação do conhecimento e pensamento computacional em outras áreas disciplinares (línguas, ciências, história, matemática, etc.);

2. Analisar o problema e decompô-lo em partes;

3. Explorar componentes estruturais de programação (variáveis, estruturas de decisão e de repetição, ou outros que respondam às necessidades do projeto) disponíveis no ambiente de programação;

4. Implementar uma sequência lógica de resolução do problema, com base nos fundamentos associados à lógica da programação e utilizando componentes estruturais da programação;

5. Efetuar a integração de conteúdos (texto, imagem, som e vídeo) com base nos objetivos estabelecidos no projeto, estimulando a criatividade dos alunos na criação dos produtos (jogos, animações, histórias interativas, simulações, etc.)
6. Respeitar os direitos de autor e a propriedade intelectual da informação utilizada;
7. Analisar e refletir sobre a solução encontrada e a sua aplicabilidade e se necessário, reformular a sequência lógica de resolução do problema, de forma colaborativa;
8. Partilhar o produto produzido na Internet.

### **Técnicas e instrumentos de avaliação das aprendizagens em ambientes computacionais**

Nesta seção faremos uma referencia breve a alguns exemplos de técnicas e Instrumentos de avaliação das aprendizagens utilizados pelos professores na sua prática de ensino em ambientes computacionais, com especial enfoque em técnicas e instrumentos de avaliação dos artefactos ou projectos, portfólios e grelhas de avaliação.

A avaliação do artefactos, que correspondem aos projectos, programas/código escrito e objetos desenvolvidos coloca um desafio importante aos professores no que diz respeito à identificação dos conceitos computacionais envolvidos na sua construção. A análise no seu conjunto e passo a passo do programa e a identificação e interpretação das rotinas criadas e a sua sequência constitui o fio da avaliação do trabalho realizado.

No caso particular do Scratch envolve por isso a leitura dos blocos e a sua interpretação quer na forma como o aluno estruturou e escreveu as instruções e as sequências quer em relação a cada conceito computacional ( uma rotina pode conter mais do que um conceito) traduzindo-se o resultado na categorização dos projectos, dos mais simples aos mais complexos , como mostra o exemplo da Tabela 1 (Ramos, 2014).

Tabela 1 Tipos de projectos e conceitos computacionais: um exemplo prático

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

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 projectos criados através do Scratch.

Um exemplo é o “Dr. Scratch “ (Figura 1) que procura não apenas calcular blocos e identificar comandos mas analisar as sequências de programação inscritas em cada projecto e associar essas sequências e a sua concepção aos princípios do pensamento computacional (Moreno, 2014). Está disponível em <http://drscratch.programamos.es> .

Através da aplicação Dr. Scratch é possível verificar conceitos computacionais envolvidos como sejam a abstracção, paralelização, lógica, sincronização, controlo , interatividade do utilizador e a representação de dados. O programa também destaca questões que possam merecer uma melhor atenção ou revisão do criador: código repetido, ou código inútil, a designação dos sprites, por exemplo.

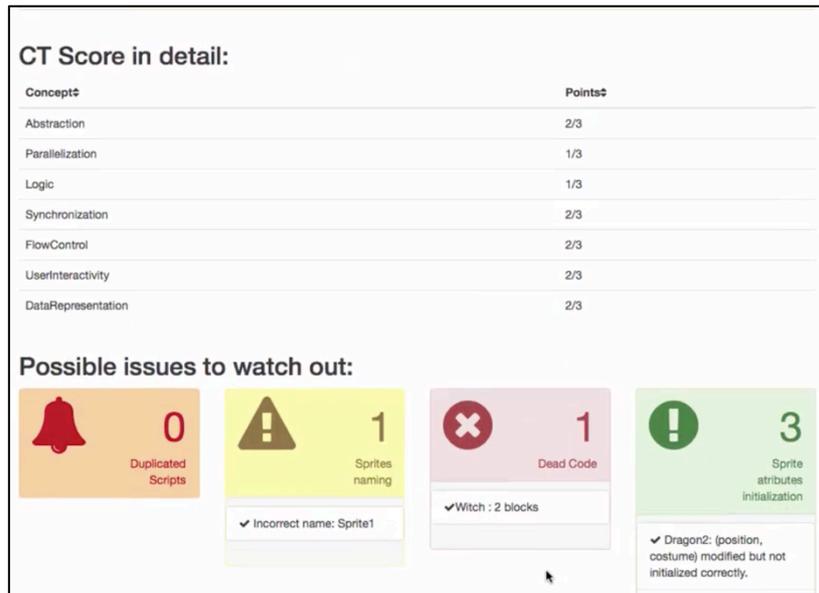


Figura 1 Princípios do pensamento computacional na aplicação Dr. Scratch

A aplicação 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, que vai de um nível mais simples e básico a um nível mais avançado, como mostra a Figura 2 (Moreno, 2014).

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 a sincronização de ações e eventos.

Dr. Scratch			
CT concept	Basic	Developing	Proficiency
<b>Parallelization</b>	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
<b>Synchronization</b>	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
<b>Data representation</b>	Modifiers of properties of sprites	Operations on vars	Operations on lists
<b>Conditional logic</b>	if	if_else	logic operations
<b>Interactivity (UI)</b>	Green_flag	key_pressed, sprite_clicked, ask_and_wait, mouse blocks	when_%s_is_>_%s, video, audio
<b>Algorithmic notions of flow control</b>	Sequence of blocks	Repeat, Forever	repeat_until
<b>Abstraction and problem decomposition</b>	> 1 scripts	> 1 scripts and > 1 sprites	def_block

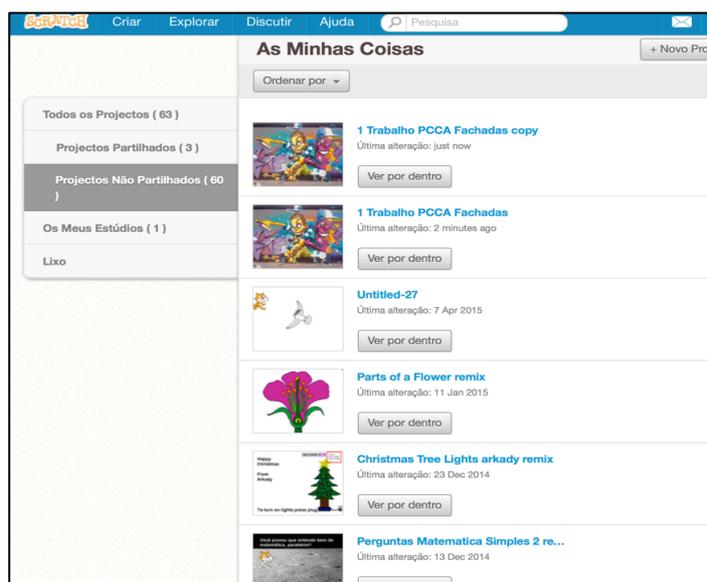
Figura 2 Pensamento computacional e níveis de desempenho em Dr. Scratch

A avaliação dos artefactos digitais criados de forma manual ou automática e adicionando dispositivos – instrumentos e técnicas - 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 uma abordagem consensual na comunidade educativa e em particular a associada ao Scratch.

#### a. *Portfólios*

Este instrumento é de grande interesse para avaliar o progresso dos estudantes do ponto de vista da aquisição e aplicação dos conceitos computacionais desenvolvidos uma vez que permite observar e acompanhar o progresso dos alunos na concepção e desenvolvimento de projectos desde patamares mais iniciais a patamares de maior complexidade (Figura 3)

No caso concreto, para além de portfólio individual e considerando a disponibilidade de armazenamento deste tipo de projectos no site da comunidade Scratch, estes podem ser igualmente partilhados com a comunidade podendo vir a ser reutilizados e reconstruídos por outros utilizadores.



### Figura 3 Portfólio de projectos Scratch

Este instrumento, no caso de professores que leccionam o 7º e 8º ano de escolaridade, teria como referencia o ponto 8 das metas curriculares que solicita a partilha dos produtos desenvolvidos pelos estudantes na Internet.

#### b. Grelhas de avaliação

As grelhas de avaliação são também muito utilizadas pelos professores para captar com mais pormenor e profundidade as aprendizagens e os processos desenvolvidos pelos alunos no decorrer dos seus projectos . Um exemplo é apresentado na Figura 4 (Brenann, 2014).

Este tipo de instrumentos poderá servir para recolher evidências relativas ao ponto 7 das metas curriculares , (nos casos em que os professores lecionam estes níveis de escolaridade): analisar e refletir sobre a solução encontrada e a sua aplicabilidade e se necessário, reformular a sequência lógica de resolução do problema, de forma colaborativa.

<b>Experimentar e interagir</b>	<b>Baixo</b>	<b>Médio</b>	<b>Alto - Elevado</b>
Descreve como construístes o teu projecto, passo a passo	Fornece uma descrição elementar da construção do projeto, mas não detalha aspectos específicos do mesmo.	Faz uma descrição genérica do projeto, de forma ordenada.	Fornece detalhes sobre as diferentes componentes dum projeto específico e descreve o modo como foram desenvolvidos, de forma ordenada.
Que outras coisas foste experimentando ao longo da elaboração do projecto?	Não apresenta exemplos específicos do que experimentou.	Deixa transparecer de forma genérica que experimentou outras coisas no projecto.	Fornece exemplos específicos de outras coisas que foi experimentando no projecto.
Que revisões fizeste e porque é que as fizeste?	Afirma não ter feito revisões, ou afirma ter feito algumas mas não exemplifica.	Descreve uma revisão específica que fez ao projecto.	Descreve aspectos específicos de coisas que acrescentou ao projecto e justifica.

Descreve as diferentes abordagens que experimentaste no teu projecto, ou quando tentaste fazer algo novo.	Não revela evidências de ter experimentado algo novo.	Fornece um exemplo de algo novo que experimentou no projecto.	Descreve com detalhe coisas novas que experimentou no projecto.
<b>Testar e corrigir</b>	<b>Baixo</b>	<b>Médio</b>	<b>Alto - Elevado</b>
Descreve o que aconteceu com o teu projecto de diferente em relação ao pretendido.	Não descreve o que resultou diferente em relação ao pretendido.	Descreve o que correu mal no projecto, mas não o que pretendia fazer.	Dá um exemplo detalhado do que aconteceu e o que pretendia, quando executa o programa.
Descreve de que forma fazes a leitura do código para encontrar a causa do problema.	Não descreve um problema.	Descreve como faz a leitura mas não apresenta um exemplo específico de encontrar um problema no código.	Descreve como faz a leitura e apresenta um exemplo específico de encontrar um problema no código
Descreve como fizeste alterações e testaste para verificar os resultados.	Não descreve que problemas teve ou a solução	Fornece um exemplo genérico sobre as alterações feitas e os testes feitos para verificar o funcionamento.	Fornece um exemplo específico sobre as alterações feitas e os testes feitos para verificar o funcionamento.
Descreve como considerarias outras formas de resolver o problema.	Não apresenta uma forma para encontrar uma solução para o problema.	Apresenta uma forma genérica para encontrar uma solução para o problema.	Apresenta um exemplo específico de como encontrar uma solução para o problema.

Figura 4 Grelha de avaliação de dimensões do pensamento computacional através do Scratch

Um exemplo adicional de grelha de avaliação (Brenann, 2014) é aqui apresentado na Figura 5, relativa aos processos de aprendizagem desenvolvidos neste tipo de ambientes computacionais, neste caso, registando as ações de reutilização e recombinação de projectos e o respeito pela autoria .

Para os professores que leccionam o 7º e 8º ano, este tipo de grelhas está referenciado ao ponto 6 Ponto das metas curriculares da disciplina TIC: respeitar os direitos de autor e a propriedade intelectual da informação utilizada .

Também no que diz respeito a algumas das capacidades das crianças e jovens e enquanto informação complementar são também utilizadas pelos professores grelhas de avaliação de forma recolher evidências sobre a forma como os projectos correspondem e em que nível de operações mentais, de maior ou menor complexidade, como seja a abstração e a modularização. Estes aspectos estão referenciados aos pontos 1,2,3 e 4 das metas curriculares.

<b>Reutilizar e recombina</b>	<b>Baixo</b>	<b>Médio</b>	<b>Alto</b>
Descreve se encontrou inspiração em outros projetos e na leitura do código disponível.	Descreve como desenvolveu as ideias ou em que projectos se inspirou.	Fornece uma descrição geral de um projeto que o inspirou .	Dá um exemplo específico do projeto que o/a inspirou.
Descreve como seleccionou uma parte de outro projeto e a adaptou	Não descreve como adaptou as ideias, scripts ou recursos de outros	Identifica scripts, ideias ou recursos que adaptou	Fornece exemplos específicos de scripts, ideias ou recursos é / ele

ao seu projeto	projetos.	de outros projetos.	adaptou de outros projetos e como.
Como refere/cita as pessoas cujo trabalho inspirou o seu próprio	Não identifica as fontes e os autores em que se inspirou no projeto.	Identifica as fontes e os autores em que se inspirou no projeto.	Documenta no projecto as fontes e os autores que inspiraram o projeto.
<b>Abstrair e modularizar</b>	<b>Baixo</b>	<b>Médio</b>	<b>Alto</b>
Como foi decidido que <i>sprites</i> eram necessários para o projecto e onde eram utilizados	Não descreve que <i>sprites</i> foram seleccionados	Fornece uma descrição geral da decisão de escolher certos <i>sprites</i>	Dá uma explicação detalhada acerca de como seleccionou os <i>sprites</i> em função do objectivo do projecto
Como foi decidido que <i>scripts</i> eram necessários para o projecto e onde eram utilizados	Não descreve que <i>scripts</i> foram criados.	Fornece uma descrição geral da decisão de criar certos <i>scripts</i> .	Dá uma explicação detalhada acerca de como criou os <i>scripts</i> em função do objectivo do projecto
Como foram organizados os <i>scripts</i> de forma a terem significado para o estudante e para os outros	Não descreve como os <i>scripts</i> foram organizados	Fornece uma descrição geral da forma como foram organizados <i>scripts</i> .	Dá uma explicação detalhada acerca de como organizou os <i>scripts</i> e porquê.

Figura 5 Grelha de avaliação de dimensões do pensamento computacional

## Metodologia

Com o objectivo de identificar as práticas de avaliação da aprendizagem predominantes no domínio da introdução do pensamento computacional na escola, procurámos indagar acerca da forma como a investigação educacional tem reflectido, na produção científica nacional, as actividades, projetos e iniciativas que as escolas e os professores estão a realizar nesses contextos e que processos de avaliação são utilizados quando os alunos são expostos a este tipo de ambientes computacionais.

Para alcançar este objectivo foi utilizada uma revisão sistemática da literatura relativa à produção académica nacional publicada e disponível aos investigadores.

As questões a que procurámos responder foram:

1. Que ferramentas e ambientes computacionais são utilizados para o desenvolvimento do pensamento computacional dos alunos ?
2. Em que níveis de escolaridade estão a ser desenvolvidos os projectos de introdução do pensamento computacional?
3. Que estratégias são adoptadas para a avaliação da aprendizagem?
4. Quais as práticas de avaliação são predominantes nas actividades desenvolvidas por professores e alunos?

Os critérios de seleção dos documentos a serem objecto de análise sistemática, foram previamente definidos e são os seguintes:

1. Foi definido como fonte de informação principal os documentos de natureza científica que correspondam e descrevam actividades de investigação e intervenção educativa no ensino básico, secundário e superior, em Portugal, dedicados à temática do pensamento computacional.
2. Foram selecionados os seguintes tipos de documentos: dissertação de Mestrado, documento de conferência, artigo, parte ou capítulo de livro e tese de Doutoramento.
3. Foram considerados como fazendo parte do corpus documental a utilizar neste estudo, os documentos académicos depositados no repositório científico nacional (RCAAP).
4. Foram considerados documentos do repositório com data de publicação nos últimos 10 anos, ou seja, entre 2005 até 2015.
5. Foram usadas as seguintes palavras-chave: “Pensamento computacional”, “Scratch Aprendizagem” e “Robótica ensino aprendizagem escola”.

Os critérios de inclusão e exclusão dos trabalhos para a realização da análise foram os seguintes:

1. Inclusão: apenas foram incluídos trabalhos que resultaram da pesquisa de palavras-chave previamente definidas e já descritas e que correspondessem a descrições de investigações e intervenções em contexto educativos relativos ao tema em estudo;
2. Exclusão: foram excluídos trabalhos referenciados mais do que uma vez, por exemplo, tese de mestrado e artigo, relativos à investigação já referenciada num outro tipo de documento; trabalhos que foram recuperados através de mais de uma pesquisa, apenas foram considerados uma única vez. Por exemplo, os mesmos trabalhos recuperados com a palavra-chave “pensamento computacional” e “Scratch Aprendizagem” foram apenas uma única vez referenciados .

A análise teve como objectivo a identificação e estratégias das práticas de avaliação no que diz respeito ao pensamento computacional, os tipos de avaliação predominantes, os ciclos de ensino onde são realizadas as investigações e intervenções e as técnicas e instrumentos de avaliação que serviram de base às avaliações realizadas das actividades descritas nos vários contextos.

## Resultados

Para responder às questões de investigação formuladas foram analisados, documento a documento, um total de 128 documentos como resultado das três consultas realizadas ao repositório científico nacional com recurso às palavras-chave referidas.

Após a análise, documento a documento e aplicados os critérios de inclusão e exclusão, foi obtido um total de 32 estudos publicados entre os anos de 2005 e 2015.

A Tabela 1 apresenta os resultados obtidos no que diz respeito às ferramentas e ambientes computacionais que são descritos na investigação e utilizados pelos professores, por ciclo de ensino, destinados ao desenvolvimento do pensamento computacional dos alunos.

Tabela 1 Ferramentas e ambientes de suporte ao pensamento computacional na escola, por ciclo de ensino

	Pré-escolar	Ensino Básico	Ensino Secundário	Ensino Especial	Ensino Superior	Total
Scratch	1	6	3		1	11
Lego Mindstorms NXT		2	17	1		19
Physical e-Toys			1			1
i-Brick			1			1
Robots “artesanais”		1				1
Raspberry Pi		1				1
Total	1	9	21	1	1	32

No que diz respeito às ferramentas e ambientes computacionais um total de 87,5% dos estudos refere-se a intervenções no domínio da robótica educativa, com recurso aos robots Lego Mindstorms NXT. De notar também o recurso ao ambiente Scratch e com menor utilização, aos outros robots e placas como é o caso do Raspberry Pi. A maioria destes estudos correspondem a documentos de tipo “teses de mestrado”. Por ciclo de ensino a análise coloca em evidência mostra que o ensino secundário recorre com mais frequência à robótica educativa e o outros níveis de ensino ao ambiente computacional Scratch. A Tabela 2 apresenta os resultados relativos às referencias a estratégias e práticas de avaliação, incluindo instrumentos e técnicas, reportados nos estudos analisados.

Tabela 2 Estratégias e práticas de avaliação em ambientes computacionais

		N.º estudos	%
Estratégias de avaliação	Avaliação formativa	28	87,5%
	Observação	24	75%
	Autoavaliação	6	18,8%
	Avaliação do trabalho em grupo	1	3,1%
	Avaliação por pares	1	3,1%

Instrumentos e técnicas de avaliação	Avaliação do artefacto/projecto	28	87,5%
	Grelhas de Observação	17	53,1%
	Questionários	14	43,8%
	Grelha de avaliação	7	21,9%
	Entrevistas	6	18,8%
	Reflexões dos alunos	4	12,5%
	Análise documental	3	9,4%
	Teste de avaliação	3	9,4%
	Apresentação oral	3	9,4%
	Relatório	3	9,4%
	Ficha de avaliação	2	6,2%
	Portfólio	2	6,3%
	Avaliação de tarefas	1	3,1%
	Quizzes	1	3,1%
	Registos do professor	1	3,1%
	Prova de Planeamento	1	3,1%
	Escala Colectiva de Desenvolvimento Lógico (ECDL).	1	3,1%
Total de estudos	32	100%	

Relativamente às estratégias de avaliação reportadas nos estudos em análise, uma larga maioria destes refere-se a adopção de estratégias de avaliação formativa (87,5%) por parte dos professores.

Da mesma forma e concomitantemente, as estratégias baseadas na observação representam igualmente uma opção importante dos professores, no seu quadro avaliativo (75%). Com menor intensidade, os estudos reportam estratégias de autoavaliação (18,8%), avaliação por pares (3,1%) e de trabalho em grupo (3,1%).

Em diversos estudos surge também com significado a referencia ao facto dos critérios de avaliação das aprendizagens serem “negociados” e acordados” ou propostos pelos alunos, o que revela a importância conferida à participação ativa dos alunos nos processos de avaliação.

Já no que diz respeito aos instrumentos e às técnicas referidas nos estudos, a maioria destes reporta a avaliação do artefacto ou projecto como o modo mais frequente de avaliar a aprendizagem dos alunos em ambientes computacionais.

Esta avaliação centrada no projecto ou produto da aprendizagem, refletirá de forma clara quais as aprendizagens realizadas e como estas se refletem no produto final. A avaliação do artefacto ou produto final revelará as aprendizagens dos alunos quer ao

nível da aquisição dos conceitos quer ao nível da aquisição e aplicação do código de programação adoptado no projecto.

A análise revela também que os estudos fazem referência ao facto de apesar da importância do produto final na avaliação das aprendizagens neste tipo de ambientes computacionais, ser necessário complementar essa avaliação, utilizando outras técnicas e instrumentos, como é o caso das grelhas de observação (53,1%), questionários (43,8%) grelhas de avaliação (21,9% ) e entrevistas ( 18,8%) entre outras como a reflexão dos alunos, os relatórios, apresentações orais, portfólios, etc. Ou seja, técnicas e instrumentos que possam proporcionar evidências da aprendizagem ao longo de todo o processo de concepção e de desenvolvimento do artefacto ou projecto.

De notar também a presença de instrumentos já desenvolvidos e usados em outros contextos, como é o caso da Escala Colectiva de Desenvolvimento Lógico (ECDL) .

### **Conclusões**

A revisão sistemática da literatura realizada revela as repostas às questões assumidas no início deste estudo e que são as seguintes:

1. As ferramentas e ambientes computacionais mais utilizados na escola para o desenvolvimento do pensamento computacional dos alunos são o Lego Mindstorms e o ambiente computacional Scratch. Os Legos são reportados com maior incidência no ensino secundário enquanto a linguagem Scratch é mais frequente nos outros níveis de ensino.
2. A avaliação formativa é a estratégia de avaliação das aprendizagens claramente mais adoptada pelos professores e reportada nos estudos analisados . A autoavaliação, a avaliação por pares e avaliação do trabalho em grupo são igualmente reportadas, mas com muito menos incidência.
3. As práticas de avaliação da aprendizagem predominantes recorrem à avaliação dos artefactos e projectos finais como aspecto central, mas recorrem a outras técnicas e instrumentos centrados na avaliação dos processos desenvolvidos pelos alunos no decorrer das actividades propostas pelos professores, de acordo com os estudos analisados. São reportados diversos instrumentos como fornecedores de evidência das aprendizagens realizadas como as grelhas de observação, questionários, grelhas de avaliação, entrevistas e reflexões dos alunos, entre outros.

Nesta análise foi possível concluir que, no universo dos estudos analisados, foram encontradas as seguintes características das estratégias e das práticas avaliação das aprendizagens em ambientes computacionais:

1. Uma avaliação predominantemente formativa na generalidade dos estudos analisados;
2. Uma avaliação que toma como central a avaliação dos artefactos, resultados/produtos finais (em forma de programas desenvolvidos pelos alunos ou outros resultados/produtos dos projectos de computação);
3. Uma avaliação que atribui uma importância significativa à avaliação dos processos desenvolvidos pelo alunos;
4. Uma avaliação que considerando a natureza prática e de processos, combina métodos e técnicas que permitem aos professores analisar, complementar e compreender de uma forma mais aprofundada, os processos envolvidos na construção desse projecto, resultado ou produto.

A preocupação em desenvolver estratégias de avaliação de natureza formativa e centradas quer nos projectos quer nos processos (procurando compreender e avaliar no contexto as aprendizagens dos alunos realizadas ao longo das intervenções reflete a adopção de “formas de avaliação formativa diversificadas centradas nos percursos” (Marques ,2009, p. 113) e parece assim constituir o padrão relativo às estratégias de avaliação adoptadas pelos professores e investigadores envolvidos neste tipo de projectos.

A nosso ver, uma explicação deste padrão pode ser interpretada à luz da influência do ambiente computacional Scratch, herdeiro da Linguagem Logo, do Media Lab e da sua extensa comunidade educativa e das influências da psicologia cognitivista ( de Jean Piaget a Seymour Papert) salientando a importância da participação ativa dos alunos nos processos de aprendizagem, as interações sociais, a importância do erro como fonte de aprendizagem, etc.

As estratégias identificadas inspiram-se certamente nesta filosofia, tonada explícita Brennan, K. ( 2014, p.8). (...) A nossa abordagem à avaliação é orientada para o processo, com foco na criação de oportunidades para os alunos poderem falar sobre a sua própria criação (e de outros) e das suas práticas criativas. Há muitas formas de recolher dados orientados aos processos e várias estratégias para realizar esta tarefa (...).

Uma nota final para sublinhar a escassez de dados de investigação relativos ao nosso país, específicos sobre este tipo de actividades destinadas à introdução do pensamento computacional na generalidade das nossas escolas e necessidade de estimular/motivar os professores a relatar as suas experiências e práticas pedagógicas neste domínio e a partilhar o conhecimento obtido através destas actividades com a comunidade educativa.

### **Referências**

- 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/>
- 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.
- 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.
- Brennan, K., Balch, C. & Chung, M. (2014) Creative computing. An Introductory Computing Curriculum Using Scratch MIT . Harvard Graduate School of Education.
- 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.
- EC European Commission (2014) e-skills for jobs in europe: measuring e-skills for jobs in europe: measuring progress and moving ahead progress and moving ahead. Final report . Consultado em 2 abril de 2015.
- Grover, S. & Pea, R. ( 2013) Computational Thinking in K–12: A Review of the State of the Field Shuchi Grover<sup>1</sup> and Roy Pea<sup>1</sup>. Educational Researcher, Vol. 42 No. 1, pp. 38–43 .
- 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.
- Neven, F. (2014) Programming as creativity. Conference report. [Available:](#)
- Philips, P. (2008) Computational Thinking: A Problem-Solving Tool for Every Classroom. Disponível em:  
<http://education.sdsc.edu/resources/CompThinking.pdf>
- Ramos, J.L. & Espadeiro, R.G. (2014) Title: Exploring computational thinking in initial teacher training: a preliminary study and reflection on practice. Universidade de Évora . Paper presented to “Scratch – Connecting Worlds. Citilab Barcelona 25-27 2013”. Disponível em : <http://www.scratch2013bcn.org/node/190> .
- 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.
- 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. (2008a). Computational Thinking. CACM Viewpoint, 33-35. Obtido de:  
<http://www.cs.cmu.edu/~wing/> .
- Wing, J. (January de 2008). Five Deep Questions in Computing. Communications Of The ACM, pp. Vol. 51, No. 1.
- 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 Mellon 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.](http://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>