

## **Scratch: uma opção válida para desenvolver o pensamento computacional e a competência de resolução de problemas**

**Rui Miguel Sousa**

Universidade do Minho  
rui\_rms@hotmail.com

**José Alberto Lencastre**

Universidade do Minho  
jlencastre@ie.uminho.pt

**Resumo** - O pensamento computacional é considerado uma aptidão fundamental no século XXI pois permite aumentar a capacidade analítica das crianças nas diversas áreas do conhecimento (Wing, 2006, 2007; Resnick, 2012). O presente artigo apresenta as principais conclusões obtidas num estudo desenvolvido do âmbito do Mestrado em Ensino de Informática, e que teve como objetivos identificar o impacto de diferentes estratégias de ensino-aprendizagem utilizadas na construção do pensamento computacional e no desenvolvimento de competências de resolução de problemas, com recurso ao *Scratch*. Inicialmente trabalharam-se, com alunos do 8º ano de escolaridade, os conceitos básicos da ferramenta, através de tarefas orientadas com vista à resolução de problemas (Jonassen, 2004). Posteriormente os alunos desenvolveram jogos multimédia aliando a dimensão interdisciplinar, dado que as personagens e os cenários foram desenhados na disciplina de Educação Visual, à dimensão comunitária, através da parceria com a Escola de Prevenção Rodoviária, sobre a qual incidiram as temáticas dos jogos. A principal conclusão retirada a partir desta experiência pedagógica é que o *Scratch* é uma opção válida para desenvolver o pensamento computacional e a competência de resolução de problemas com este público alvo.

**Palavras-chave:** Pensamento Computacional; Scratch; Resolução de Problemas

### **Introdução**

São cada vez maiores os desafios que têm sido colocados à nossa sociedade dada a vertiginosa evolução tecnológica que se tem verificado. Desde a construção do primeiro computador (1946), o **ENIAC** (*Electrical Numerical Integrator and Computer*), desenvolvido para fins militares, até à atualidade, em que o uso do computador está presente no nosso dia a dia, novos reptos têm emergido. A sua massificação levou a que, nas mais diversas profissões, se tornasse imperativo ter, pelo menos, conhecimentos na ótica do utilizador de determinados programas. O maior desafio que hoje se impõem aos utilizadores é o de deixarem de ser meros consumidores de conteúdos (programas, jogos) e passarem a criadores (Resnick, 2012; Resnick, et al., 2009). É neste contexto que surge a aptidão (*skill*) que é vista como fundamental no século XXI – o pensamento computacional (Wing, 2007).

O presente artigo é parte de um estudo desenvolvido no âmbito do plano de Intervenção Pedagógica Supervisionada, inserido no Estágio Profissional do 2º ano no Mestrado em Ensino de Informática da Universidade do Minho. Por permitir desenvolver a capacidade analítica em diversas áreas do conhecimento, e por estar previsto nas novas metas curriculares já a serem implementadas na escola onde decorreu o estudo, o desenvolvimento do pensamento computacional e da competência de resolução de problemas em alunos de três turmas do 8º ano do ensino básico tornaram-se nos objetivos principais.

### **Contextualização**

O pensamento computacional é a capacidade de desencadear o processo de formulação de problemas do mundo real e de os solucionar (Cuny, Snyder, & Wing, 2010; Wing, 2007). Ao ser promovido o seu desenvolvimento, os indivíduos ficam um passo à frente da literacia tecnológica (Resnick, 2012; Phillips, 2009), deixando de ser meros utilizadores. Passam a ter, não só a capacidade de desenvolver os seus próprios sistemas, como reforçam competências adjacentes, sendo elas: o pensamento abstrato (utilização de diferentes níveis de abstração para perceber os problemas e, passo a passo, solucioná-los), o pensamento algorítmico (expressão de soluções em diferentes passos de forma a encontrar a forma mais eficaz e eficiente de resolver um problema), o pensamento lógico (formulação e exclusão de hipóteses) e o pensamento dimensionável (decomposição de um grande problema em pequenas partes ou composição de pequenas partes para formular uma solução mais complexa) (Phillips, 2009; Resnick, 2007-2008). Tais capacidades, associadas por defeito às ciências da computação, transpõem-se para outras áreas de conhecimento e consequentemente para o dia a dia.

Karen Brennan e Mitchel Resnick, investigadores do MIT - Massachusetts Institute of Technology -, têm levado a cabo nos últimos anos diversos estudos relacionados com o pensamento computacional no que diz respeito à criação de produtos multimédia. Os resultados desses estudos deram origem ao quadro de referências para estudar e avaliar o desenvolvimento do pensamento computacional, e que engloba três dimensões: (i) conceitos computacionais; (ii) práticas computacionais; e (iii) perspetivas computacionais.

Os (i) conceitos computacionais são sete e englobam sequências (identificar uma série de etapas de uma tarefa), ciclos (executar a mesma sequência várias vezes), execução em paralelo (fazer as ações decorrerem ao mesmo tempo), eventos (fazer um acontecimento provocar outro), condições (tomar decisões com base em condições), operadores (expressar operações matemáticas e lógicas), e dados (armazenar, recuperar e atualizar valores) (Brennan & Resnick, 2012). As (ii) práticas computacionais estão focadas no processo de pensar e aprender indo para além *do que se aprende para o como se aprende* (Brennan & Resnick, 2012). Assim, foram definidos quatro conjuntos de práticas: ação iterativa e incremental (desenvolver, verificar se funciona e, em seguida, continuar a desenvolver), teste e depuração (verificar se tudo funciona e encontrar e corrigir erros, se for o caso), reutilização e reformulação (fazer algo utilizando o que já foi feito, ou o que outros fizeram), abstração e modulação (construir algo grande unindo conjuntos de partes mais pequenas) (MIT, 2011). As (iii) perspetivas computacionais são três: expressão (a

computação é um meio de criação), a conexão (criar *com* e *para* outros), questionar (a tecnologia e com a tecnologia) (MIT, 2011). Para estes autores, a avaliação do desenvolvimento do pensamento computacional em jovens, pode ser feita através da análise dos projetos tendo por base as três dimensões por estes definidas.

Sendo o pensamento computacional um processo de formulação de problemas é inevitável mencionar o método de resolução de problemas. Este método é visto também como uma aptidão (skill) importante para os alunos em qualquer contexto (Jonassen, 2004). O autor refere que a aprendizagem advém dos problemas que precisam de ser resolvidos e que enquanto os alunos os resolvem estão a aprender e a compreender. Jonassen caracteriza um problema como sendo uma entidade desconhecida num determinado contexto, tradução livre de: "(...) a problem is an unknown entity in some context." (Jonassen, 2004, Pág. 3).

No entanto, encontrar um problema e resolvê-lo possui um valor social, cultural e intelectual, isto é se um aluno não sentir a necessidade de determinar o desconhecido passa a não existir um problema concreto (Jonassen, 2004).

O método de resolução de problemas ao ser utilizado, auxilia os alunos a *aprenderem a aprender* (Papert, 1993) obrigando o aluno a procurar a resposta ao seu problema em vez de receber uma resposta dada pelo professor, desenvolvendo assim o domínio dos procedimentos (Echeverría & Pozo, 1998). Este método revela possuir um poder motivador pois o aluno passa a ser o principal agente do processo de aprendizagem.

Neste sentido, o Scratch, um ambiente visual de programação inspirado na linguagem Logo, tem vindo a ser uma das ferramentas mais utilizadas na promoção do pensamento computacional. Foi criado no Media Lab do Massachusetts Institute of Technology (MIT) tendo em conta os "interesses e necessidades dos jovens (idades entre 8 e 16 anos)" (Resnick, Kafai, & Maeda, 2003, p.10), e permite-lhes criarem projetos interativos, como jogos, histórias, simulações e vídeos (Maloney, Resnick, Rusk, Silverman, & Eastmond, 2010). A aplicação possui elementos visuais atrativos para os jovens e a sua execução é imediata possibilitando feedback em tempo real. É gratuita e pode ser descarregada a partir da internet para qualquer computador, desde que este obedeça aos requisitos mínimos. A sua sintaxe é muito simples pois consiste em encaixar diferentes blocos de comandos num género de puzzle ou construção de legos. Nesses blocos existem conectores que sugerem as possibilidades de junção com outros, desenhados de forma a que todas as possibilidades de encaixe façam sentido do ponto de vista sintático da linguagem, prevenindo assim, possíveis erros. (Resnick, 2012).

No decorrer do estudo, todos os elementos do *Scratch* foram abordados tendo como estratégia a atribuição de tarefas orientadas com vista à resolução de problemas. Com esta abordagem construcionista (Papert, 1993) procurou-se dar as bases para que os alunos pudessem aprender a aprender, porque "o tipo de conhecimento que as crianças necessitam é aquele que as ajudará a adquirir mais conhecimento" (Papert, 1993, p.139).

## Problema e Objetivos

O pensamento computacional manifesta-se já na idade infantil, uma vez que as crianças raciocinam de forma computacional (Nunes, 2011). Contudo esta capacidade não é explorada no ensino básico pelo que acaba por se perder (Nunes, 2011).

Assim, sendo a primeira vez que o desenvolvimento do pensamento computacional e a implementação do método de resolução de problemas surgem no sistema educativo português, ao estarem previstos nas novas metas curriculares da disciplina de Tecnologias da Informação e Comunicação (TIC) do 8º ano do 3º ciclo do ensino básico, tornou-se relevante medir o impacto da estratégia utilizada para o desenvolvimento das já referidas competências. O estudo teve então como objetivos identificar o impacto da estratégia:

- na construção do pensamento computacional;
- no desenvolvimento da competência de resolução de problemas.

A amostra do estudo incidiu sobre três turmas do 8º ano de escolaridades (69 alunos) de uma escola na cidade do Porto (Tabela 1). Os alunos desta escola são reflexo de uma grande diversidade populacional quer a nível socioeconómico quer a nível cultural. A especificidade da população escolar era oriunda de contextos familiares em evidente desvantagem social, económica e cultural (Projeto de Educativo, 2010-2013).

|                | Nº de Alunos | Distribuição por Género  | Observações  |
|----------------|--------------|--|--|
| <b>Turma A</b> | 28           | <ul style="list-style-type: none"><li>• Masculino: 16</li><li>• Feminino: 12</li></ul> | <ul style="list-style-type: none"><li>• 7 alunos com retenções,</li><li>• 1 aluna com deficiência auditiva e adequação curricular</li><li>• 4 alunos com currículo específico individual</li></ul> |
| <b>Turma B</b> | 22           | <ul style="list-style-type: none"><li>• Masculino: 13</li><li>• Feminino: 12</li></ul> |  |
| <b>Turma C</b> | 19           | <ul style="list-style-type: none"><li>• Masculino: 14</li><li>• Feminino: 6</li></ul>  | <ul style="list-style-type: none"><li>• 11 alunos com retenções</li><li>• 2 alunos com adequações curriculares</li></ul>   |

**Tabela 1** – Caracterização da amostra do estudo

## Metodologia

O conjunto de elementos que constituíram as orientações do projeto de intervenção em que decorreu o estudo em causa, a ação docente, a dimensão investigativa e a sua avaliação (mediante a recolha e análise de dados), aproxima-se de uma clara articulação da metodologia de Investigação-Ação com a de Estudo de Caso. Esta aproximação pode ser enquadrada recorrendo a uma breve revisão de literatura a respeito da já referida metodologia.

A Investigação-Ação é descrita por Watts (1985) como um processo em que os participantes analisam as suas próprias práticas usando técnicas de investigação. Coutinho (2011) acrescenta ainda que é um processo interativo e sempre focado num problema. Segundo Sousa (2005), esta estratégia metodológica de estudo é geralmente utilizada pelo professor sobre a sua ação pedagógica com os alunos, observando os seus comportamentos e atitudes.

Por este “projeto de ação” ter decorrido num determinado contexto, considera-se que dentro da metodologia da Investigação-Ação, no que se refere à componente investigativa, configurou-se um Estudo de Caso. Como a própria expressão indica, um estudo de caso pressupõem o examinar de um determinado caso, isto é, uma identidade num contexto real (Yin, 1989). Assim, no Estudo de Caso, “examina-se o caso (...) em detalhe, em profundidade, no seu contexto natural, reconhecendo-se a sua complexidade e recorrendo-se para isso todos os métodos que se revelem apropriados” (Coutinho, 2011, p. 293)

Neste sentido, o estudo que ora se descreve tratou-se de um plano de investigação que envolveu o estudo intensivo e detalhado de uma ação pedagógica desenvolvida para a promoção de competências relevantes num caso que englobou três turmas com características muito próprias.

De forma a compreender o impacto das estratégias utilizadas bem como a consecução dos objetivos previsto, foram utilizadas diferentes fontes, sendo elas: (i) análise documental; (ii) observação naturalista; (iii) reflexões individuais; (iv) questionário inicial; (v) demonstração; (vi) grelhas de avaliação de tarefas; (vii) grelha de avaliação do projeto; (viii) teste de avaliação; (ix) relatórios de atividade da plataforma moodle; e (x) questionário final.

### **Análise dos Dados**

O estudo organizou-se em três fases. A primeira fase envolveu a análise dos documentos reguladores e do contexto em que iria decorrer. Durante o decorrer esta última, percebemos que seria necessário adoptar uma estratégia que motivasse os alunos, que os desafiasse e os envolvesse na concepção de algo significativo (Papert, 1993), dada a sua clara desmotivação para a escola. Aplicamos, então, um questionário relativamente aos seus interesses e à utilização que davam ao computador, para que pudéssemos partir dos seus interesses e conhecimentos prévios, e assim, tornar as suas aprendizagens mais significativas (Coll, et al., 2001). Verificamos que a maioria dos alunos utilizavam-no para jogar e que revelavam interesse em aprender a utilizar novos programas de computador. Promovemos, por isso, uma demonstração do Scratch, para perceber a receptividade das turmas face ao mesmo. Os alunos consideraram-no interessante e demonstraram vontade em utiliza-lo para desenvolver jogos. Relativamente à sua temática, aliaram-se duas vertentes: a vontade de atribuir uma dimensão holística e interdisciplinar ao projeto de intervenção e a existência de uma Escola de Prevenção Rodoviária no agrupamento. Assim, foi proposto aos alunos que desenvolvessem jogos neste âmbito, com o objetivo de apoiar as atividades da referida escola, desenhando os cenários e as personagens na disciplina de Educação Visual (Figura 1).

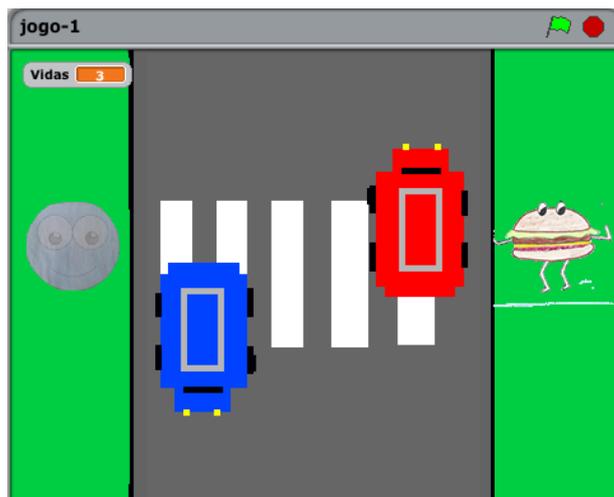


Figura 1 – Exemplo de projeto desenvolvido pelos alunos

Para a segunda fase do estudo, subdividiram-se, pelas diferentes aulas, os conteúdos necessários para o desenvolvimento do jogo. Desenharam-se, conseqüentemente, quatro tarefas problema, de acordo com as bases teóricas da resolução de problemas (Echeverría & Pozo, 1998; Jonassen, 2004) com o objetivo dos alunos aplicarem os conhecimentos previamente disponíveis (adquiridos na parte inicial da aula) para lhes dar resposta (Echeverría & Pozo, 1998). Com esta abordagem construcionista (Papert, 1993) procuramos dar as bases para que os alunos pudessem prender a aprender, porque “o tipo de conhecimento que as crianças necessitam é aquele que as ajudará a adquirir mais conhecimento” (Papert, 1993, p.139). Para uma avaliação efetiva das tarefas problema foram construídas grelhas de avaliação com base nos conceitos computacionais de Brennan & Resnick (2012). Os jogos finais foram desenvolvidos e avaliados segundo grelhas igualmente criadas para o efeito, baseadas no quadro de referências de Brennan & Resnick (2012). Na terceira fase foi avaliado o estudo à luz dos objetivos propostos.

Relativamente ao primeiro objetivo de investigação, **identificar o impacto da estratégia na construção do pensamento computacional**, ao longo das aulas foram promovidas diversas ações que visavam promover esta aptidão. Nas **tarefas problema** encontravam-se presentes diferentes **competências adjacentes ao pensamento computacional**. Ao resolverem as tarefas, os alunos, eram obrigados a utilizar diferentes níveis de abstração para desmontar os problemas e solucioná-los (**pensamento abstrato**) (Phillips, 2009; Wing, 2006). Para além disso, tiveram que resolvê-los procurando as soluções mais eficazes e eficientes, ao optarem por certos comandos e não por outros (**pensamento algorítmico**) (Phillips, 2009; Wing, 2006). Durante estes processos esteve intrínseca a formulação e exclusão de hipóteses (**pensamento lógico**) (Phillips, 2009; Wing, 2006). Enquanto um problema era resolvido existia, no *Scratch*, a possibilidade de irem verificando (testando) se uma determinada opção resultava ou não, de forma a irem excluindo hipóteses. Os alunos depararam-se também com a necessidade de decompor tarefas propostas em diferentes partes, de forma a chegarem ao todo que era a solução (**pensamento dimensionável**) (Phillips, 2009; Wing, 2006).

De acordo com as **dimensões** que definem o **pensamento computacional**, definidas por **Brennan & Resnick** (2012), em todas as tarefas, estiveram presentes, de forma repartida, elementos das três dimensões: (i) **conceitos computacionais**; (ii) **práticas computacionais**; e (iii) **perspetivas computacionais** (Tabela 2).

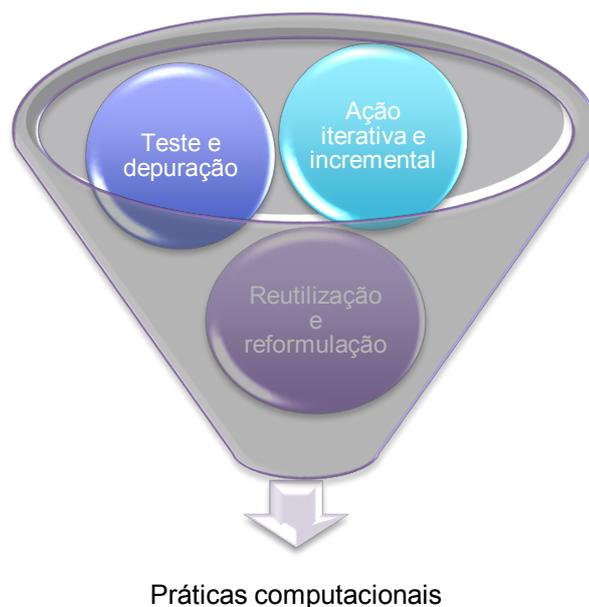
| Conceitos computacionais | Tarefa 1 | Tarefa 2 | Tarefa 3 | Tarefa 4 |
|--------------------------|----------|----------|----------|----------|
| 1. Sequências            | ✓        | ✓        | ✓        | ✓        |
| 2. Ciclos                |          |          | ✓        | ✓        |
| 3. Execução em paralelo  | ✓        | ✓        |          | ✓        |
| 4. Eventos               |          |          | ✓        | ✓        |
| 5. Condições             |          |          | ✓        | ✓        |
| 6. Operadores            |          |          | ✓        | ✓        |
| 7. Dados                 |          |          |          | ✓        |

**Tabela 2** – Resumo dos conceitos computacionais presentes nas diferentes tarefas

Para chegarem a uma representação do problema, os alunos eram obrigados a identificar as diferentes etapas de uma tarefa (sequências). Por outro lado, alguns pontos de algumas tarefas exigiam a execução da mesma sequência várias vezes (ciclos). Também em outros pontos era necessário fazer com que determinadas ações decorressem ao mesmo tempo (execução em paralelo). Os eventos (fazer um acontecimento provocar outro), estavam também permanentemente presentes. De igual forma, as condições (tomar decisões com base em condições), e operadores (expressar operações matemáticas e lógicas), na medida em que, recorrendo aos respetivos sinais aritméticos estabeleciam-se comparações. O armazenamento, recuperação e atualização de valores (dados) encontrava-se presente sempre que havia necessidade de guardarem valores e de os atualizarem consoante a concretização de determinadas ações, como perder um ponto e uma “vida”.

Nas tarefas estavam analogamente vinculadas (ii) práticas computacionais (Figura 2) pois os alunos iam desenvolvendo e testando para verem se funcionava (ação iterativa e incremental). Nestes casos era-lhes dada a hipótese de corrigirem erros, se verificassem (teste e depuração). Ao existir um crescente avanço no grau de complexidade das tarefas, os alunos tinham a hipótese de se servir de conhecimentos e saberes já utilizados previamente ou apreendidos com outros colegas (reutilização e reformulação).

À semelhança do pensamento abstrato e dimensionável já referidos, a abstração e modulação (construir algo grande unindo conjuntos de partes mais pequenas) era uma prática corrente ao longo das tarefas.



**Figura 2** – Práticas computacionais presentes nas tarefas

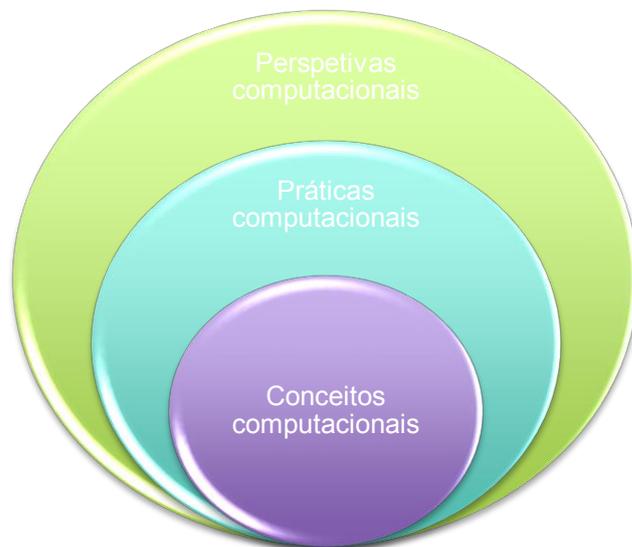
As três (iii) perspetivas computacionais (Figura 3) foram transversais ao longo de todo o processo. Os alunos ao desenvolverem as tarefas, e apesar de existirem algumas linhas orientadoras, criavam algo de novo (expressão) pois incluíam elementos pessoais às suas criações. Também durante o desenvolvimento das tarefas os alunos trabalhavam em conjunto (conexão) e quando solicitavam ajuda, era essencialmente para questionarem a tecnologia (questionar), como por exemplo, “...é possível ter múltiplos palcos?”.



**Figura 3** – Perspectivas computacionais presentes nas tarefas

Após a fase de dotar os alunos das competências necessárias para desenvolverem o trabalho de projeto, deu-se início ao desenvolvimento do mesmo. Ao longo das aulas e sessões de apoio extra,

os alunos puderam evidenciar conceitos, desenvolver práticas e perspetivas computacionais (Figura 4).



**Figura 4** - Esquema resumo das dimensões do pensamento computacional presentes no trabalho de projeto

No que diz respeito à segunda questão de investigação (b), **identificar o impacto da estratégia no desenvolvimento da competência de resolução de problemas**, todas as tarefas bem como o trabalho de projeto eram situações abertas e sugestivas (apesar de existirem requisitos obrigatórios, podiam ser adicionados elementos adicionais) que exigiam dos alunos uma atitude ativa para alcançarem uma solução (Echeverría & Pozo, 1998). As tarefas eram constituídas por vários passos a seguir, não podendo ser solucionadas de forma imediata (Echeverría & Pozo, 1998). Para a resolução de cada ponto das tarefas e o desenvolvimento do projeto, foi necessário obedecer a quatro etapas (Pólya, 2003): primeiro era necessário compreender o problema, de seguida elaborar um plano e executá-lo, e por fim verificar os resultados. As tarefas já se encontravam subdivididas em pontos mas no trabalho de projeto eram os alunos que definiam estes pontos autonomamente. Por outro lado, os problemas apresentados nas tarefas eram definidos quanto à sua estrutura, complexidade, dinamismo e domínio específico (Jonassen, 2004). Em cada tarefa estavam presentes um número limitado de conceitos (estrutura) e um determinado número de questões (complexidade). Relativamente ao dinamismo, os alunos procuravam soluções e iam adaptando a sua compreensão do problema de forma a chegar a uma solução. O domínio específico englobou a variância da estrutura e complexidade (elementos já definidos nos enunciados), mas relativamente ao dinamismo era um processo mental realizado por cada aluno de forma particular. Esta aprendizagem levou a que os alunos adquirissem conhecimentos e competências associadas ao trabalho de grupo (comunicação, relação interpessoal, cooperação e respeito mútuo) (Leite & Esteves, 2005), uma vez que, como já referido, as tarefas e o trabalho de projeto foram desenvolvidos em pares.

Através dos resultados obtidos com a avaliação das tarefas e do projeto, recorrendo às grelhas referidas anteriormente (Gráfico 9), é possível evidenciar o cumprimento dos objetivos da investigação.

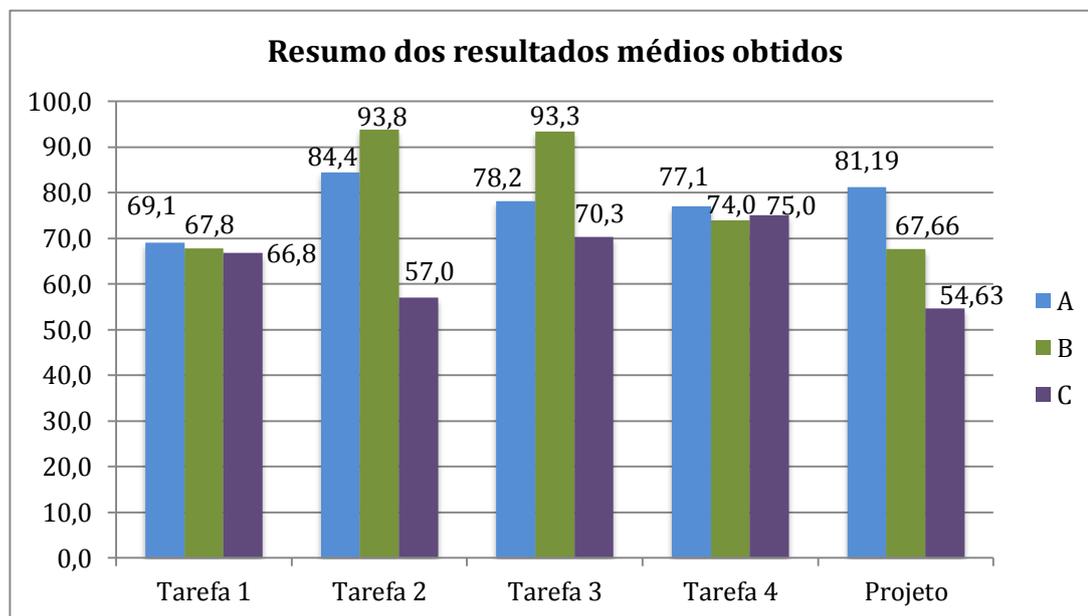


Gráfico 6 – Resumo dos resultados médios obtidos

Em todas as tarefas e no trabalho de projeto, as três turmas obtiveram resultados satisfatórios ou bastante satisfatórios. A turma A, obteve o seu resultado médio mais alto na tarefa dois (84,4%), bastante aproximado dos 81,1% do trabalho de projeto, e o resultado médio mais baixo na tarefa um (69,1%). A turma B, obteve o seu resultado médio mais alto na tarefa dois (93,8%) e três (93,3%), e o resultado médio mais baixo no trabalho de projeto (67,6%). A turma C, obteve o seu resultado médio mais alto na tarefa quatro (75%) e o resultado médio mais baixo na tarefa dois (57%). No trabalho de projeto chegou aos 54,6%.

Apesar dos resultados diferirem de turma para turma, devido às suas características, é possível afirmar que as estratégias utilizadas foram promotoras da construção/desenvolvimento do pensamento computacional, do desenvolvimento da competência de resolução de problemas e de competências relativas ao uso do software *Scratch*.

## Conclusão

O pensamento computacional é a capacidade de formular um problema e de encontrar uma solução (Cuny, Snyder, & Wing, 2010). Diretamente associado a este conceito encontramos o quadro de referências de Brennan e Resnick (2012) que identifica os já referidos conceitos, práticas e perspetivas computacionais. Assim, ao estarem presentes em todas as tarefas e no trabalho de projeto, a busca para a resolução de problemas onde se encontravam presentes os conceitos, práticas e perspetivas computacionais, é possível afirmar que este pensamento foi promovido e que a estratégia utilizada funcionou, pois os alunos atingiram resultados positivos.

Jonassen (2004) refere que os alunos aprendem a resolver problemas. Os problemas podem ser definidos relativamente a quatro parâmetros quanto à sua estrutura, complexidade, dinamismo e domínio específico (Jonassen, 2004). Estes parâmetros eram contemplados nos enunciados das tarefas. Pólya (2003) define quatro etapas para a resolução de problemas: (i) compreender o problema; (ii) elaborar um plano; (iii) executar um plano; (iv) verificar resultados. Assim, de forma a que os alunos compreendessem o problema era-lhes fornecido o enunciado, e feita uma demonstração do resultado final da tarefa. Desta forma, era-lhes possível elaborar um plano mentalmente, para ser executado com vista ao cumprimento dos objetivos. No final, havia espaço para verificar os resultados obtidos.

Com o estudo desenvolvido, foi possível concluir que o *Scratch* é uma opção válida para desenvolver o pensamento computacional e a competência de resolução de problemas em alunos do 8º ano.

## Referências

- Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *American Educational Research Association meeting* (pp. 1-25). Vancouver: AERA.
- Coll, C., Martín, E., Mauri, T., Miras, M., Onrubia, J., Solé, I., et al. (2001). *O Construtivismo na sala de aula*. Porto: Edições ASA.
- Coutinho, C. (2011). *Metodologia de Investigação em Ciências Sociais e Humanas: Teoria e Prática*. Coimbra: Almedina.
- Cuny, J., Snyder, L., & Wing, J. (2010). *Demystifying Computational Thinking for Non Computer Scientists*. work in progress.
- Echeverría, M., & Pozo, J. (1998). Aprender a resolver problemas e resolver problemas para aprender. In J. Pozo, *A solução de problemas: aprender a resolver, resolver para aprender*. Porto Alegre: Artmed.
- Jonassen, D. H. (2004). *Learning to solve problems - an instructional design guide*. São Francisco: Pfeiffer.
- Leite, L., & Esteves, E. (2005). Ensino orientado para a aprendizagem baseada na resolução de problemas na Licenciatura em Ensino de Física e Química. In B. Silva, & L. Almeida, *Actas do Congresso Galaico-Português de Psico-Pedagogia* (pp. 1751-1768). Braga: Universidade do Minho.
- Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. (Novembro de 2010). The Scratch Programming Language and Environment. *ACM Transactions on Computing Education*, 10 (16), pp. 1-16.
- MIT. (2011 de Setembro). *Computação Criativa - uma introdução ao pensamento computacional baseada no conceito de design*. (EduScratch, Trans.)
- Nunes, D. (9 de setembro de 2011). *Ciência da Computação na Educação Básica*. Obtido em 15 de outubro de 2013, de *Jornal da Ciência*: <http://www.jornaldaciencia.org.br/Detail.php?id=79207>

- Papert, S. (1993). *The Children Machine*. New York: BasicBooks.
- Phillips, P. (2009).  
[http://www.csta.acm.org/ProfessionalDevelopment/sub/CSIT09Presentations/Phillips\\_Computational.pdf](http://www.csta.acm.org/ProfessionalDevelopment/sub/CSIT09Presentations/Phillips_Computational.pdf) (Acessível em 17 de Junho de 2013)
- Pólya, G. (2003). *Como resolver problemas*. Lisboa: Gradiva.
- Projeto de Educativo. (2010-2013).
- Resnick, M. (July-August de 2012). Point of View - Reviving Papert's Dream. *Educational Technology*, 52, pp. 42-46.
- Resnick, M. (dezembro/janeiro de 2007-2008). Sowing the Seeds for a more Creative Society. *Learning & Leading with Technology*.
- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., et al. (Novembro de 2009). Scratch: Programming for All. *Communication of ACM*, 52, pp. 60-67.
- Sousa, A. (2005). *Investigação em Educação*. Lisboa: Livros Horizonte.
- Watts, H. (1985). When teachers are researcher, teaching improves. *Journal of Staff Development*, 6 (2), pp. 118-127.
- Wing, J. M. (Março de 2006). Computacional Thinking. *Communications of the ACM*, 49, pp. 33-35.
- Wing, J. M. (2007). [http://www.cs.cmu.edu/afs/cs/usr/wing/www/Computational\\_Thinking.pdf](http://www.cs.cmu.edu/afs/cs/usr/wing/www/Computational_Thinking.pdf) (Acessível em 20 de Junho de 2013)
- Wing, J. M. (2010). *School of computer Science*. <http://link.cs.cmu.edu/article.php?a=600> (Acessível em 15 de Junho de 2013)
- Yin, R. (1989). *Case Study Research: Design and Methods*. USA: Sage Publications Inc.