

Iniciação à Programação no 1.º Ciclo do Ensino Básico: Análise da implementação da medida a alunos do 3.º ano

António Soares Marques

Tese para obtenção do Grau de Doutor em
Educação
(3.º ciclo de estudos)

Orientadora: Prof. Doutora Sandra Carina Guimarães

março de 2021

Folha em branco

Iniciação à Programação no 1.º Ciclo do Ensino Básico: Análise da implementação da medida a alunos do 3.º ano

António Soares Marques

Tese para obtenção do Grau de Doutor em
Educação
(3.º ciclo de estudos)

Orientadora: Prof. Doutora Sandra Carina Guimarães

Júri:
Prof. Doutor Mário Marques Freire
Prof. Doutor José Luís Pires Ramos
Prof.^a Doutora Maria de Fátima de Jesus Simões
Prof.^a Doutora Ana Cristina Dias Pinheiro
Prof. Doutor Carlos Alberto Correia Carreto
Prof.^a Doutora Sandra Carina Machado Guimarães

22 de fevereiro de 2021

Folha em branco

Dedicatória

À minha Esposa e Filho,

A quem dedico esta Tese, por todo o apoio e carinho - suporte fundamental ao longo deste percurso.

Folha em branco

Agradecimentos

Aos meus pais,

Um especial agradecimento aos meus pais pelos princípios que me inculcaram.

Aos meus colegas do grupo 550,

Que me ajudaram em atividades cruciais, tais como a construção dos instrumentos a utilizar, e a fazer opções nos momentos de incerteza, mas, principalmente, pela disponibilidade e prontidão com que colaboraram nesta investigação.

À minha orientadora, Prof^a. Doutora Sandra Carina Guimarães,

Pelo apoio, motivação, orientação, disponibilidade e, sobretudo, por todos os sábios conselhos que me deu ao longo destes 3 anos.

Aos docentes deste Doutorado,

Pela sua partilha e ensinamentos.

Folha em branco

Prefácio

Depois de ter concluído o Bacharelato em Engenharia Informática encetei, de imediato, a minha atividade profissional na área que mais me plenificava, a área da programação. Não obstante, o arbítrio de evoluir, que na prática é uma faceta pessoal, fez-me sem demora coadunar a vida profissional com a vida académica, assim, a orientação seguida culminou na conclusão da Licenciatura de Engenharia Informática. Porém, e quase sem disso me aperceber, com frequência estabeleci relação com saberes de índole vária e com a formação, inicialmente como formador interno na empresa onde trabalhava, posteriormente, e por indicação da mesma, como formador noutras entidades.

Estava num ponto em que o gosto pelo ensino era crescente, no entanto sentia que era imprescindível opulentar a minha formação para conseguir dar resposta aos desafios do processo ensino/aprendizagem. Assim, ingressei na Universidade da Beira Interior, onde concluí a Licenciatura em Ensino de Informática e direcionei a minha atividade profissional apenas para a área do Ensino, quer no ensino público, quer no ensino profissional.

Nas últimas décadas, as políticas de educação especial em Portugal conduziram à inclusão nas nossas escolas, quase na totalidade, de crianças e jovens com deficiência e incapacidade, e esse ponderoso levou-me a fazer a parte curricular do Mestrado em Ciências da Educação, na especialidade de Educação Especial e Domínio Cognitivo e Motor, através da Escola Superior de Educação Almeida Garrett.

No ano letivo 2015/16, a Direção-Geral da Educação (DGE) promoveu o projeto-piloto “Iniciação à Programação no 1.º Ciclo do Ensino Básico”, projeto desafiante para os professores do grupo 550 – Informática, que consubstanciou a elaboração de novos materiais e a formulação de novas estratégias para dar resposta aos objetivos definidos. Assim brotou a ideia deste estudo que, apesar da clara consciência das dificuldades com que me iria deparar, desde logo me estimulou, não só por me parecer iminente a carência de encontrar materiais que fossem ao encontro dos objetivos propostos, mas também por considerar extremamente importante existirem instrumentos que pudessem avaliar a eficácia do projeto.

Em resultado deste desiderato, espero que esta tese, os planos de aula, o instrumento de avaliação e os trabalhos científicos que dela resultaram possam contribuir para a explanação das virtualidades da utilização do pensamento computacional em contexto educativo, mais concretamente nos alunos do 3.º ano.

Folha em branco

Resumo

Do lazer ao trabalho, os meios tecnológicos estão enraizados no nosso dia a dia e são inúmeras as ferramentas que podemos encontrar. O ensino das Tecnologias de Informação e Comunicação, no nosso país, tem acompanhado esta tendência, quer em termos de conteúdos, quer na dinamização de projetos, como os clubes de robótica ou o ensino da programação. Esta investigação foca-se num desses projetos, a “Iniciação à Programação no 1.º Ciclo do Ensino Básico”, apresentando uma proposta concreta a desenvolver no âmbito do mesmo e avaliando o seu impacto em termos de pensamento computacional e de autoeficácia, domínios que têm surgido na literatura como um forte preditor de motivação, persistência e desempenho ao longo do tempo.

Nos estudos de validação dos instrumentos que aqui se apresentam participaram 329 alunos, do 3.º e do 4.º ano de escolaridade, pertencentes a três Agrupamentos de Escolas da região centro. Das análises efetuadas, e apesar da diversidade e heterogeneidade presentes em cada grupo/sala de aula, salienta-se o impacto positivo do projeto no pensamento computacional havendo, como seria de esperar, conceitos em que os alunos revelam maior dificuldade em adquirir e consolidar e, portanto, deixa-se o repto de se desenvolver novas estratégias/metodologias no ensino da programação.

Foi ainda desenvolvido um programa de intervenção como proposta curricular que qualquer professor responsável por este projeto poderá implementar e desenvolver. Se, por um lado, damos resposta à necessidade que todos os professores envolvidos sentem na planificação e organização das suas aulas, por outro lado, ao avaliarmos o impacto do mesmo, estamos a validar a proposta aqui apresentada e a promover as competências dos alunos a que se propõe o projeto do Ministério da Educação.

O programa de intervenção, composto por 21 sessões, com aulas orientadas, através de um ensino e aprendizagem centrada no aluno, recorre à ferramenta Scratch para que, de uma forma lúdica, sejam explorados diversos conceitos básicos de computação. Através de um modelo quasi-experimental, com avaliação de um pré-teste e de um pós-teste, em alunos do 3.º ano, os resultados evidenciam um incremento significativo no domínio do pensamento computacional.

O trabalho que aqui se apresenta poderá dar um contributo importante para esta área específica das ciências da computação e, sobretudo, o programa proposto poderá ser uma

ferramenta importante à qual, qualquer professor da área possa recorrer para promover a aprendizagem dos alunos no domínio da programação.

Este trabalho teve a anuência da Associação Nacional de Professores de Informática e hoje há já uma equipa de trabalho a aperfeiçoar o programa de intervenção aqui proposto com a utilização da versão atual do Scratch. Estes recursos estão disponíveis em www.programacao1ciclo.pt, que se encontra em constante atualização.

Palavras-chave

Pensamento Computacional; Programação 1.º Ciclo; Programa de Intervenção

Abstract

From leisure to work, technological capabilities are ingrained into our daily lives surrounded by countless tools we can resort to. The teaching of Information and Communication Technologies in Portugal has been keeping pace with this trend, both concerning contents and stimulating projects like robotics clubs or programming teaching. This research is focused in one of these projects, the Initiation to Programming in Primary Education, presenting a concrete proposal to be developed in the aforementioned context and evaluating its impact concerning computational thought and self-efficiency, domains that have emerged in the literature as a strong predictor of motivation, persistence and performance over time.

329 students of the third and fourth grades of primary school, belonging to three School Groupings of the Centre region, participated in the validation studies of the instruments here presented. Of the analyses made, and despite the diversity and heterogeneity present in each group/classroom, a positive effect, stemming from the project, in computational thinking was revealed, expectedly there being concepts in which students revealed a greater difficulty in assimilating and consolidating, and thus leaving the challenge of developing new strategies/methodologies in teaching programming.

It was also developed an intervention programme as a curricular proposal that any teacher responsible for this project will be able to implement and develop. If, on the one hand, we answer to the need felt by every teacher involved in planning and organising their classes, on the other hand, whilst evaluating its impact, we are validating the proposal presented here and promoting the students' skills as they are set in the Ministry of Education's project.

The intervention programme, composing of 21 sessions, with classroom orientation, through a student-centred teaching and learning, makes use of the Scratch programme so that, in a ludic fashion, various basic concepts of computing can be explored. Through a quasi-experimental model, evaluated through a pre-test and a posterior test on third grade students, results evidenced a significant increment in the mastery of computational thinking.

The work here presented could give an important contribution to this specific field of computing sciences and, above all, the proposed programme could constitute an

important tool with which any teacher on this field can resort to, in order to promote the students' learning in the programming field.

This work had the consent of the National Association of IT Teachers and already today there is a work team perfecting the intervention programme here proposed with the current version of Scratch. These resources are available in www.programacao1ciclo.pt, which is constantly updated.

Keywords

Computational Thinking;Primary School Programming;Intervention Programme

Índice

INTRODUÇÃO	1
ENQUADRAMENTO TEÓRICO	3
A Escola e as Tecnologias de Informação e Comunicação.....	5
O Pensamento Computacional	18
Autoeficácia	27
Ensino da Programação.....	30
METODOLOGIA	39
Estudo 1 – Teste de Pensamento Computacional	41
1.1. Teste de pensamento computacional desenvolvido por Marcos González.....	43
1.2. Procedimentos da adaptação à versão portuguesa	45
1.3. Validação do instrumento	51
1.3.1. Critérios de definição e estratificação dos participantes	51
1.3.2. Caracterização dos participantes	56
1.3.3. Instrumento de recolha de dados	58
1.3.4. Procedimentos de Recolha	59
1.3.5. Dificuldades encontradas	64
1.3.6. Resultados da validação do teste de pensamento computacional.....	65
Estudo 2 – Teste de perceção de autoeficácia para a área da programação	69
2.1. Conceção do Teste de perceção de autoeficácia para a área da programação ..	71
2.2. Validação do instrumento	73
2.2.1. Participantes, instrumentos e procedimentos.....	74
2.2.2. Resultados da validação do teste de perceção de autoeficácia para a área da programação	75
Estudo 3 – Avaliação do impacto da “Iniciativa IP1” nos participantes em estudo	77
3.1. Questões de investigação	79
3.2. Participantes, instrumentos e procedimentos	80

3.3.	Resultados	82
3.3.1.	Resultados do teste de pensamento computacional	82
3.3.2.	Resultados do teste de perceção de autoeficácia para a área da programação.....	87
Estudo 4 – Programa de Intervenção		89
4.1.	Aplicação Scratch	91
4.2.	Sessões do Programa	93
4.3.	Procedimentos.....	99
4.4.	Condições de implementação do programa	103
4.5.	Dificuldades encontradas	105
4.6.	Avaliação do Programa de Intervenção.....	106
4.6.1.	Objetivos.....	106
4.6.2.	Caracterização dos participantes	107
4.6.3.	Resultados do Pensamento Computacional.....	109
4.6.4.	Resultados da perceção de autoeficácia para a área da programação	117
CONCLUSÕES E LIMITAÇÕES DO ESTUDO.....		Erro! Marcador não definido.
1.1.	Conclusão da Investigação	121
1.2.	Limitações e sugestões para investigações futuras	123
REFERÊNCIAS.....		125
ANEXOS.....		139
Anexo A: Autorização de Marcos Román González para a utilização e adaptação do seu teste de pensamento computacional		
Anexo B: Autorização do MIME		
Anexo C: Autorização dos Encarregados de Educação		
Anexo D: Poster apresentado no Regional Helix 2018 Conference		
Anexo E: Artigo Scratch 3 – Beginners Programming Course in 3rd Year of Primary School		
Anexo F: Programa de Intervenção		
Anexo G: Protocolo de Investigação (Questionário, Teste de Autoeficácia para a área de programação e Teste e Pensamento Computacional)		

Lista de Figuras

Figura 1. Esquema de componentes do pensamento computacional (Zapata-ros, 2015, p. 13)	22
Figura 2. Exemplo de pergunta do Teste de Pensamento Computacional.....	44
Figura 3. Exemplo de pergunta que sofreu alterações após a reflexão falada.....	50
Figura 4. Exemplo de pergunta do Teste de autoeficácia para a área da programação .	72
Figura 5. Ambiente de Trabalho do Scratch	92

Folha em branco

Lista de Gráficos

Gráfico 1 – Distribuição de participantes por idade.....	57
Gráfico 2 – Distribuição de participantes por anos de frequência na iniciativa	80
Gráfico 3 – Gráficos de normalidade do Pré-teste e do Pós-teste.....	110
Gráfico 4 – Gráfico de diferenças no grupo de controlo.....	113
Gráfico 5 – Gráfico de diferenças no grupo de teste.....	113

Folha em branco

Lista de Tabelas

Tabela 1. Principais projetos, programas e iniciativas educativas tecnológicas em Portugal (1985-2013)	13
Tabela 2. Escolas com a IP1 no ano letivo 2016/2017	53
Tabela 3. Distribuição dos participantes por Agrupamento.....	56
Tabela 4. Índice do género da amostra.....	57
Tabela 5. Medidas de consistência interna do Teste do Pensamento Computacional ...	66
Tabela 6. Medidas de consistência interna do Teste de Autoeficácia.....	75
Tabela 7. Distribuição da amostra por ano letivo	81
Tabela 8. Índice de idades da amostra	81
Tabela 9. Resultados do Teste do Pensamento Computacional por género.....	83
Tabela 10. Resultados do Teste do Pensamento Computacional em relação à frequência na iniciativa.....	83
Tabela 11. Resultados do Teste do Pensamento Computacional em relação ao ano de escolaridade	84
Tabela 12. Resultados do Teste do Pensamento Computacional em relação à idade	84
Tabela 13. Resultados do Teste do Pensamento Computacional em relação aos 4 grupos de questões.....	85
Tabela 14. Autoeficácia perante a iniciativa.....	87
Tabela 15. Autoeficácia perante a iniciativa por ano de escolaridade	87
Tabela 16. Autoeficácia perante a iniciativa por sexo	88
Tabela 17. Resumo das sessões	94
Tabela 18. Resumo de Conteúdos abordados	96
Tabela 19. Interação com os elementos do Scratch	98
Tabela 20. Índice de idades dos participantes.....	107
Tabela 21. Índice de idades por grupos (GT e GC)	108
Tabela 22. Índice do género da amostra do grupo experimental.....	108
Tabela 23. Distribuição do género pelo GT e GC.....	108
Tabela 24. Médias pré-teste e pós-teste	109
Tabela 25. Resultados do teste de Shapiro-Wilk	109
Tabela 26. Tabela de dados com comparação de médias do grupo de teste	111
Tabela 27. Tabela de dados com comparação de médias do grupo de controlo.....	112
Tabela 28. Tabela de dados com as diferenças do pré-teste e do pós-teste	112
Tabela 29. Tabela de dados com as diferenças do pré-teste e do pós-teste.....	114
Tabela 30. Tabela de dados por género no pré-teste.....	115

Tabela 31. Tabela de correlação no pré-teste	116
Tabela 32. Tabela de correlação no pós-teste	116
Tabela 33. Tabela de resultados de autoeficácia	117
Tabela 34. Tabela da análise das amostras pareadas de autoeficácia.....	118

Lista de Acrónimos

AEC	Autoeficácia no uso do computador
ANPRI	Associação Nacional de Professores de Informática
DGE	Direção-Geral da Educação
DGEstE	Direção-Geral dos Estabelecimentos Escolares
DOCX	Documento criado através do Microsoft Word 2007 ou superior
GC	Grupo de Controlo
GE	Grupo Experimental
IP1	Introdução à Programação no 1.º ciclo
MEC	Ministério da Educação e Ciência
MIME	Monitorização de Inquéritos em Meio Escolar
MOOC	Curso Online Aberto e Massivo, do inglês Massive Open Online Course
OCDE	Organização para a Cooperação e Desenvolvimento Económico
PDF	Portable Document Format
TIC	Tecnologias de Informação e Comunicação

Folha em branco

INTRODUÇÃO

Nas últimas décadas ocorreram diversas transformações sociais que estão profundamente ligadas à evolução tecnológica e que, direta ou indiretamente, impõem notórias alterações ao quotidiano de qualquer pessoa. Drago (2001) evidencia o forte relacionamento entre a possibilidade de desenvolvimento de uma sociedade com os conhecimentos/domínios das novas tecnologias de informação e comunicação, em vez das suas riquezas naturais ou da sua capacidade de produção industrial.

De acordo com Ponte (2000), em termos de Educação “o problema é levar a escola a contribuir para uma nova forma de humanidade, onde a tecnologia está fortemente presente e faz parte do quotidiano, sem que isso signifique submissão à tecnologia” (p. 89). Tal como menciona Figueiredo (2001), “a escola do futuro exige, acima de tudo, permanente intencionalidade estratégica e capacidade para imaginar um futuro tornado possível pelos progressos tecnológicos” (p.80).

A área do pensamento computacional, apesar de ser uma área transversal, tem sido explorada intencionalmente pelas disciplinas lecionadas pelo grupo de informática (550), nos diferentes níveis de ensino, quer através das áreas curriculares, quer através de projetos que envolvem a comunidade educativa no geral, como, por exemplo, através dos clubes de robótica.

O estudo que aqui se apresenta procura ser um contributo para as práticas pedagógicas dos professores do 1.º ciclo da área das Tecnologias de Informação e Comunicação (TIC) enriquecendo, desta forma, a própria área das ciências da educação.

Em 2015, a Direção-Geral da Educação (DGE) promove o projeto-piloto “Iniciação à Programação no 1.º Ciclo do Ensino Básico”, tendo, à partida, diversos apoios de entidades/instituições relacionadas com a área das ciências da computação, tais como a Associação Nacional de Professores de Informática (ANPRI), o Centro de Competência TIC da Universidade de Évora (CCTIC UE), o Centro de Competência TIC da Escola Superior de Educação do Instituto Politécnico de Setúbal (CCTIC ESE Setúbal) e a MICROSOFT.

A implementação do projeto iniciou-se em fevereiro de 2015 através de convite a todas as Escolas e Agrupamentos, destacando-se que foi da responsabilidade dos Diretores a inscrição, bem como a verificação do cumprimento das orientações gerais. O interesse

foi imediato e, logo no primeiro ano, estiveram envolvidas, de forma voluntária, 625 escolas, 1483 turmas e 27810 alunos.

Atualmente, muitas são as Escolas e Agrupamentos que optaram por manter esta oferta formativa, e é neste seguimento que surge o programa de intervenção que é construído e avaliado no âmbito deste trabalho. O impacto da iniciativa no desenvolvimento do pensamento computacional e autoeficácia dos alunos em estudo são também objeto de atenção.

O trabalho que aqui se apresenta está organizado em três secções. A primeira, dedicada ao enquadramento teórico, onde são apresentados alguns conceitos fundamentais desta área de estudo, tais como o pensamento computacional e a autoeficácia; é também explorada a interligação entre as Tecnologias de Informação e Comunicação e a Educação, bem como o ensino da programação.

Segue-se a metodologia, onde são descritos os estudos realizados quer em termos de validação de instrumentos, quer nos impactos que a iniciativa está a ter no pensamento computacional e na perceção de autoeficácia na área da programação. Também é descrito o programa de intervenção, constituído por 21 sessões, construído no âmbito deste trabalho, bem como a sua implementação e avaliação.

Por fim, são apresentadas as conclusões do trabalho e as implicações para os processos ensino e de aprendizagem. São também discutidas as limitações dos estudos sendo propostas algumas sugestões para investigações futuras.

ENQUADRAMENTO TEÓRICO

Folha em branco

A Escola e as Tecnologias de Informação e Comunicação

O processo de ensino-aprendizagem é de facto, cada vez mais, um desafio que todos os professores têm que encarar no que concerne a todos os metamorfismos que a nossa sociedade apresenta face a uma evolução constante. Ponte (1986) reconhece que uma “nova sociedade precisa de um novo tipo de escola para cumprir novos objetivos de ensino.”(p.55) e o Ministério da Educação tem feito esforços para dotar as escolas e os professores de meios que viabilizem a construção do paradigma atual da aprendizagem centrada no aluno e no desenvolvimento das suas competências.

Um grande problema vivido na nossa sociedade é “a contradição da facilidade”. Deparamo-nos facilmente com a dificuldade em lidar com todas as novidades/potencialidades que as novas tecnologias criam para facilitar o nosso dia a dia, criando, na maioria de nós, um sentimento de impotência perante a arduidade de gerir/assimilar toda a informação disponibilizada apenas com um clique e que deveria simplificar a comunicação, aproximando as pessoas. E este facto nem sempre acontece devido à falta de definição de prioridades que as nossas crianças fazem no seu dia a dia e da sua estreita visão acerca das ciências da computação.

Nesta linha de pensamento, no nosso país, ao longo do tempo, têm surgido várias medidas/iniciativas e em vários domínios do saber. A área em que se insere este trabalho, o pensamento computacional, é uma das áreas mais recentes de investimento e que muito interesse tem suscitado no ensino. A medida “Escola, Professores e Computadores Portáteis”, por exemplo, foi lançada no final do ano letivo de 2005/2006 pelo Ministério da Educação, com o objetivo de “promover a melhoria das condições de trabalho na Escola e apoiar o uso individual e profissional das TIC por parte dos Professores”(Ramos et al., 2009, p. 200) que foi alcançado de forma muito significativa “ao nível das competências dos alunos no uso das tecnologias, do interesse e motivação dos alunos nas disciplinas onde utilizaram a tecnologia e da participação dos alunos nos processos de aprendizagem” (Ramos et al., 2009, p. 201).

A destacar também o projeto-piloto “Iniciação à Programação no 1.º Ciclo do Ensino Básico”, no ano letivo 2015/2016, promovido pela DGE, juntamente com outras entidades relacionadas com a área da educação e da informática. Desde o primeiro momento muitas foram as escolas que abraçaram o desafio e são hoje muitos os alunos do 3.º e 4.º anos envolvidos no mesmo. Como referem os documentos oficiais do projeto, a principal finalidade é que “os alunos não só aprendam a programar mas, ao mesmo tempo, aprendam programando”(Direção Geral de Educação, 2015, p. 3), promovendo

ao mesmo tempo um ensino em termos de ciências da computação, de forma a alargar os seus horizontes, e a forma como veem o computador. São claros os objetivos, mas não é proposto pela DGE um programa específico, ou documento de apoio no desenvolvimento do currículo na área.

As próprias orientações Curriculares para a Educação Pré-Escolar já salientam a relação e a importância das TIC na Educação sendo esta transversal a todas as áreas. A este respeito, por exemplo, I. Silva et al. (2016) referem que “a importância dos meios tecnológicos e informáticos no conhecimento do mundo, próximo e distante, e no contacto com outros valores e culturas faz com que a sua utilização no jardim de infância seja considerada como um recurso de aprendizagem ”(p. 93).

Para Ramos (1999),

a integração do computador e das tecnologias da informação e comunicação deve ser entendida como um conjunto de processos complexos de utilização planeada, sistemática, adequada e ‘natural’ nos diferentes planos: no do currículo e da aprendizagem, em estreita relação com as finalidades, objetivos, conteúdos, experiências e actividades curriculares (p. 90).

No entanto, tal como refere Cuban (1993), a implementação das inovações tecnológicas na Escola é diferente do que acontece noutras áreas, uma vez que a Escola está muito ligada às suas crenças culturais no que concerne à forma de ensinar e de aprender. A mesma autora destaca ainda, como condicionante a esta implementação, a própria organização escolar.

Segundo Akker et al. (1992) devem ser consideradas quatro categorias nos fatores que influenciam a implementação de inovações na prática educacional: “ national (and/or state and/or district) context; characteristics of the school (organization); external support; characteristics of the innovation itself.” (p. 66)

Em termos de contexto nacional, Ramos et al. (2009) focam-se nas políticas educativas, tendo em vista as suas decisões nos assuntos educacionais que influenciam o tempo, recursos e instalações disponibilizados para a inovação. Na organização da Escola, salientam o papel dos líderes, que atualmente são as equipas que compõem as Direções das Escolas, as experiências com outras inovações, a forma como a partilha de saberes e experiência é feita entre os professores, bem como as facilidades em termos de logística

(tempo, dinheiro, materiais). No que respeita aos apoios externos, o foco está na formação contínua dos professores, bem como na interligação com especialistas e professores de outras escolas, de forma a proporcionar um enriquecimento pessoal e profissional. Por fim, na última categoria, que se refere às características das inovações, destaca-se a importância da interligação entre a necessidade e a inovação, bem como a clareza dos objetivos e significado prático de uma proposta de mudança.

Relativamente à introdução do Pensamento Computacional nas Escolas, os autores Ramos e Espadeiro (2014) mencionam que:

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. (p. 20)

Todas estas temáticas não são novas nem apareceram de um momento para o outro, uma vez que a utilização do computador no processo de ensino-aprendizagem já está enraizada e já ninguém coloca em causa que a educação necessita de dar resposta a novos desafios. Ao analisar a utilização dos telemóveis, segundo estudos do INOV - INESC Inovação do Instituto Superior Técnico, relativos ao projeto FAQtos, da responsabilidade de Catarré e Correia (2019), que analisam os resultados dos inquéritos realizados a estudantes do ensino secundário, de 2010/11 a 2017/18, conclui-se que é caso raro o aluno com idade superior a 14 anos que não possua pelo menos um telemóvel; ou que a internet, com maior ou menor facilidade/frequência é utilizada por quase toda a comunidade escolar. No entanto, estas alterações de comportamentos levam à criação de novas necessidades, surgindo um novo consumismo, uma nova linguagem, um novo mundo do trabalho com a alteração de paradigmas de emprego e, neste sentido, será importante, como já referem as próprias políticas educativas portuguesas, trabalhar desde o pré-escolar, promovendo o pensamento computacional, integrando no próprio currículo e aprendizagem mais académica do aluno, ou seja, logo no primeiro ciclo.

A utilização do computador na educação nunca irá deixar de ser uma ótima ferramenta para melhorar o ensino-aprendizagem, pois, quando usado da forma correta, é um promotor da motivação do aluno, cria situações de multidisciplinidade, além de desenvolver um espírito de experimentação; no entanto, o que não significa, como referem Teodoro e Freitas (1992), ter apenas um computador na sala:

não se trata simplesmente de substituir o quadro preto ou o livro pelo ecrã do computador. A introdução das tecnologias da informação na educação pode estar associada à mudança do modo como se aprende, à mudança das formas de interação entre quem aprende e quem ensina, à mudança como se reflete sobre a natureza do conhecimento. (p.10)

Há já alguns anos que se considera que as novas tecnologias têm potencialidades que, necessariamente, devem ser exploradas. O que nem sempre acontece é compreender que as potencialidades e as formas como se exploram têm de ser atualizadas.

Tal como destacam Lye e Koh (2014), por todo o mundo, os currículos têm sido atualizados nesta área, de forma a incluir alunos em idades cada vez mais jovens, utilizando uma enorme heterogeneidade de intervenções educacionais. Por exemplo, o programa de robótica “TangibleK” juntou ferramentas de programação de computadores e robótica destinadas a crianças do jardim de infância com o objetivo de desenvolver o pensamento computacional, robótica, programação e resolução de problemas (Bers et al., 2014).

No que se refere ao ensino da informática em Portugal, este sofreu, tal como era expectável, um desenvolvimento que tentou acompanhar a rápida evolução desta área.

É óbvio que, de início, a informática, mais concretamente as Tecnologias de Informação e Comunicação, em contexto educativo, eram bastante diminutas; no entanto, ao longo dos últimos trinta anos, têm assumido, à semelhança do que acontece na comunidade em geral, um papel cada vez mais marcante e, hoje em dia, já conquistou o seu próprio espaço dentro do contexto educativo e nas matrizes curriculares do sistema educativo português.

Um dos marcos desta evolução foi a reorganização curricular do Ensino Básico em 2003, com a criação/implementação da disciplina de TIC como disciplina obrigatória para o 9.º ano e para a componente geral do 10.º ano de escolaridade, do curso científico-

humanístico e dos cursos tecnológicos. Esta disciplina, e segundo João (2003), foi criada porque nessa altura o

sistema educativo português assume como objectivo estratégico a necessidade de assegurar a todos os jovens o acesso às tecnologias da informação e comunicação como condição indispensável para a melhoria da qualidade e da eficácia da educação e formação à luz das exigências da sociedade do conhecimento. Esta medida integra-se na prossecução dos objectivos e metas para o futuro do espaço de educação e formação europeu, definidos pelo Conselho Europeu aquando da aprovação do “Programa de Trabalho Pormenorizado sobre o seguimento dos objectivos dos sistemas de educação e formação na Europa” e por forma a contribuir para o sucesso da Estratégia de Lisboa. (p. 3)

Foi apresentada como sendo uma área curricular que teria uns moldes de funcionamento diferentes do habitual e, segundo a autora referida anteriormente, aconselhava-se que esta fosse uma disciplina prática com uma forte componente experimental, solicitando-se, também, uma inovação nas metodologias e atividades utilizadas em sala de aula, privilegiando-se a concretização de projetos, a pesquisa, a resolução de problemas e a simulação da realidade das empresas, bem como a interligação com as outras áreas curriculares.

Através destas indicações, a disciplina ganhou o seu espaço dentro da comunidade educativa e nos planos anuais de atividades, fomentando nos alunos a autonomia para a realização de trabalhos para as várias disciplinas e tornando-se um apoio capital para desafios futuros em termos académicos ou profissionais.

Porém, o Decreto-Lei n.º 74/2004, de 26 de março, sofreu sucessivos ajustamentos, sendo retificado pela declaração de Retificação n.º 44/2004, de 25 de maio, e regulamentado pela Portaria n.º 550-D/2004, de 21 de maio. Foi ainda alterado pelo Decreto-Lei n.º 24/2006, de 6 de fevereiro, que por sua vez foi ainda retificado pela declaração de Retificação n.º 23/2006, de 7 abril, e regulamentado pela Portaria n.º 259/2006, de 14 de março.

Algum tempo depois, em 2007, concretizou-se uma nova alteração, após o Ministério da Educação ter analisado as recomendações feitas pelo Grupo de Avaliação e

Acompanhamento da Implementação da Reforma do Ensino Secundário, onde se refere a necessidade de fortificar a vertente prática e experimental nos cursos científico-humanísticos com o aumento da carga horária em um tempo semanal (noventa minutos), nas disciplinas de Biologia e Geologia e Física e Química A, no 10.º e 11.º anos (GAAIRES, 2007). Relacionado com este fortalecimento da carga horária destas disciplinas, houve a necessidade da cessação da obrigatoriedade da disciplina de TIC no 10.º ano, no entanto, houve também a preocupação de indicar que esta disciplina devia ser antecipada para o 3.º ciclo, após um período de estudo e de adaptação, de forma a garantir o desenvolvimento das aprendizagens das TIC ao longo do 3.º ciclo do ensino básico, ou seja, as indicações seriam para que à saída do 9.º ano, escolaridade obrigatória nessa altura, todos os alunos ficassem aptos a utilizar as TIC no seu dia a dia e na sua vida escolar.

Em 2011, no seguimento de uma discussão pública, surge uma nova revisão da estrutura curricular, que foi homologada pelo Decreto-Lei n.º 139/2012, de 5 de julho de 2012. Neste decreto pode ler-se que,

a disciplina de Tecnologias de Informação e Comunicação inicia-se no 7.º ano de escolaridade, garantindo aos alunos mais jovens uma utilização segura e adequada dos recursos digitais e proporcionando condições para um acesso universal à informação, funcionando sequencialmente nos 7.º e 8.º anos, semestral ou anualmente, em articulação com uma disciplina criada pela escola, designada por oferta de escola. (p. 3479)

O mesmo decreto concede também às escolas flexibilidade para definirem a duração dos tempos letivos, selecionarem as metodologias, entre outros aspetos, o que levou a que o ensino das TIC deixasse de ser igual em todas as escolas do ensino público. Surgiram então vários cenários de organização dos tempos, sendo os mais comuns o funcionamento da disciplina de forma anual, nos dois anos, com a atribuição de um tempo letivo (45 minutos ou 50 minutos), ou então a distribuição semestral, onde a disciplina é lecionada em 2 tempos (90 ou 100 minutos, dependendo da escola) durante um semestre. Houve ainda escolas que implementaram a disciplina de forma anual, mas a funcionar quinzenalmente com 2 tempos.

Para esta disciplina, apesar de, muitas vezes, haver mais de trinta alunos para salas com o máximo de catorze computadores, nunca foi previsto haver desdobramento de turmas;

no entanto, esta organização ocorreu em situações esporádicas, e por decisões assumidas pelas Direções das escolas/agrupamentos, muitas vezes justificadas pela existência de disponibilidade/horário de professores do quadro de escola ou de zona pedagógica que estavam colocados nas mesmas.

Houve também novas indicações acerca do funcionamento geral da disciplina, deixando esta de respeitar conteúdos para atingir metas curriculares que funcionaram através de quatro domínios específicos, mais concretamente: informação, produção e comunicação e colaboração, além do domínio da segurança, que seria abordado de forma transversal nos outros domínios atrás referidos. Estas metas curriculares foram formalizadas através de dados científicos, bem como recomendações produzidas no âmbito da Agenda Digital Europeia através da Digital Agenda Scoreboard e da Organização para a Cooperação e Desenvolvimento Económico (OCDE) com base nas E-Skills for The 21st Century: Fostering Competitiveness, Growth and Jobs, que sublinham a importância de, desde cedo, os alunos utilizarem as TIC como ferramentas de trabalho (Horta et al., 2012).

Atualmente, vigora o Decreto-Lei n.º 55/2018, de 6 de julho, que veio definir um novo currículo para o ensino básico e secundário e estabelecer novas regras que dão mais autonomia às escolas para tomarem decisões que visam ajudar os alunos a alcançar as competências previstas no perfil dos mesmos, à saída da escolaridade obrigatória, que se encontra desde a publicação da Lei n.º 85/2009, de 27 de agosto (estabeleceu o alargamento da escolaridade obrigatória), no 12.º ano, podendo esta obrigatoriedade ser cessada após o aluno ter completado os dezoito anos de idade.

Este Decreto-Lei traz muitas novidades para o funcionamento das escolas, para a disciplina de TIC e para o sistema educativo em geral. Uma das principais alterações prende-se com a extinção das metas curriculares e com o aparecimento das aprendizagens essenciais que, tal como se pode ler no Decreto-Lei, são os conhecimentos, as capacidades e as atitudes que os alunos têm, obrigatoriamente, de adquirir e desenvolver em cada ano de escolaridade, podendo ser aprofundadas, reforçadas e enriquecidas por decisão da cada escola/agrupamento.

Mais concretamente, em termos das TIC, este Decreto-Lei implementa novas medidas no primeiro, segundo e terceiro ciclos que, no entanto, só vão ser implementadas em todos os alunos no ano letivo de 2021/2022, uma vez que se prevê uma implementação progressiva do Decreto-Lei.

No que se refere ao primeiro ciclo, estão contempladas, na matriz, as componentes de Cidadania e Desenvolvimento e de TIC como sendo de integração curricular transversal potenciada pela dimensão globalizante, ou seja, o professor titular de turma tem por obrigação explorar estas áreas no decorrer de todo o ciclo.

No segundo ciclo, quinto e sexto ano, a matriz curricular introduz a disciplina de TIC na componente de Educação Artística e Tecnológica; no entanto, não define em concreto a sua carga curricular, deixando que cada escola/agrupamento o faça da melhor maneira, estipulando apenas que no conjunto das quatro disciplinas desta componente (Educação Visual, Educação Tecnológica, Educação Musical e TIC) os alunos deverão ter, quer no quinto ano, quer no sexto ano, uma carga letiva semanal de 325 minutos, isto porque, tal como menciona no Decreto-Lei n.º 55/2018, de 6 de julho,

tomando por referência a matriz curricular-base e as opções relativas à autonomia e flexibilidade curricular, as escolas organizam o trabalho de integração e articulação curricular com vista ao desenvolvimento do Perfil dos Alunos à Saída da Escolaridade Obrigatória. As escolas organizam os tempos letivos na unidade que considerem mais adequada. (p. 2940)

Além disso, o n.º 2 do artigo 19.º refere ainda que a organização do funcionamento das disciplinas pode ocorrer de um modo trimestral, semestral ou outro. Estas indicações fazem com que, a nível nacional, exista uma grande variedade na forma como a disciplina está a ser implementada e, analisando o questionário sobre a implementação e funcionamento da disciplina de TIC, disponibilizado pela ANPRI no seu site¹, verifica-se que, no mínimo, a atribuição feita é de 25 minutos. Esta disciplina está a ser lecionada de forma anual ou semestral, mais concretamente através das seguintes modalidades: a) 90/100 minutos semestral; b) 90/100 minutos de 15 em 15 dias anual; c) 45/50 minutos anual; d) 45/50 minutos semestral.

Relativamente ao terceiro ciclo, os princípios orientadores são iguais aos do segundo ciclo, onde a disciplina de TIC é implementada nos três anos (sétimo, oitavo e nono ano), também na componente de Educação Artística e Tecnológica que, neste caso, é composta por três disciplinas (Educação Visual, Complemento à Educação Artística e TIC) e com

¹ <https://forms.gle/NAibTWEFhDwj52Bz9>

uma carga letiva semanal de 175 minutos em cada ano. A atribuição feita pelas escolas, por norma, também é semelhante ao que acontece no segundo ciclo.

Em termos de ensino secundário, a disciplina continua apenas a constar na matriz dos cursos profissionais, que tiveram origem nas escolas públicas através da Portaria 550-C/2004, que regulamentava a criação, organização e gestão do currículo, bem como a avaliação e certificação das aprendizagens dos cursos profissionais de nível secundário, nos termos estabelecidos no Decreto-Lei n.º 74/2004, de 26 de março, mais concretamente na componente Sociocultural, com a atribuição de 100 horas letivas ao longo dos 3 anos, que, normalmente, são lecionadas no 10.º ano através de quatro tempos semanais ou no 10.º e 11.º ano, com a atribuição de dois tempos semanais em cada ano. No entanto, desde a implementação do Decreto-Lei n.º 55/2018, a escola pode optar pelo desenvolvimento da disciplina de TIC ou por uma Oferta de Escola, de frequência obrigatória, podendo gerir a carga horária em função da necessidade de reforço das aprendizagens.

Na sequência do referido, a evolução das TIC no ensino público esteve sempre relacionado, direta ou indiretamente, com a implementação de projetos de reestruturação tecnológica que foram implementados ao longo do tempo, dotando assim as escolas públicas de recursos tecnológicos que possibilitassem o ensino das TIC e a implementação de outros projetos relacionados, como por exemplo os clubes de robótica.

Tabela 1. Principais projetos, programas e iniciativas educativas tecnológicas em Portugal (1985-2013)

Designação	Data	Entidade responsável
Projecto MINERVA	1985-1994	Ministério da Educação (GEP e DEPGEF)
Programa Nónio-Século XXI 1996	1996-2002	Ministério da Educação
uARTE- Internet na Escolas	1997-2002	Ministério da Ciência e Tecnologia
Programa Internet@EB1	2002-2005	Ministério da Ciência e Tecnologia; Escolas Superiores de Educação; FCCN
Projecto CBTIC@EB1	2005-2006	Ministério da Educação (CRIE)
Iniciativa Escolas, Professores e C. Portáteis	2006/2007	Ministério da Educação
Plano Tecnológico da Educação	2007-2011	Ministério da Educação (GEPE)
Internet Segura	Desde 2007	UMIC; Ministério da Educação (ERTE/PTE-DGIDC); FCCN; Microsoft
Iniciativa e-Escolinha	2008-2011	MOPTC
Aprender e Inovar com TIC	2010-2013	Ministério da Educação (ERTE/PTE-DGIDC)

Fonte: (S. Pereira & Pereira, 2011, p. 160)

A tabela anterior, apresentada por S. Pereira e Pereira (2011) exhibe o projeto Minerva como sendo o primeiro desta evolução. Este programa, da responsabilidade do Ministério da Educação, pretendia introduzir as TIC no ensino não superior e durou quase dez anos.

Também Ponte (1994) realçou que, com base neste projeto, os

computadores começaram por ser vistos numa forma dual: ou para o ensino da informática ou para serem usados como tecnologia. (...) trata-se do estabelecimento de novos objectivos e do desenvolvimento de novas capacidades, o que se faz através duma variedade de novos recursos tecnológicos. Os computadores são máquinas. Mas as tecnologias de informação, mais do que isso, são uma perspectiva cultural. Elas constituem uma importante ferramenta intelectual que permite estender as capacidades de pensamento e de acção dos seres humanos nos mais diversos domínios. (p. 54)

O mesmo autor destaca ainda que este projeto teve impacto em centenas de milhares de alunos e que envolveu numerosas instituições.

O Plano Tecnológico da Educação e a iniciativa do e-escolinhas, que aparece no seu seguimento como uma medida de enorme impacto na sociedade portuguesa é também salientado por S. Pereira e Pereira (2011) em particular na generalização do uso de computadores pelos alunos do 1.º ciclo do ensino básico e em permitir às escolas beneficiar de um conjunto de equipamentos informáticos, infraestruturas tecnológicas e diversos serviços para uma melhoria da aprendizagem e ensino.

Atualmente, estão em curso várias iniciativas/projetos que têm a aprovação/apoio do Ministério da Educação com o objetivo de fomentar as TIC nas escolas, que não só a própria disciplina. São bons exemplos disso os seguintes projetos, que fazem parte de um vasto elenco disponível para a comunidade educativa em geral:

- Portugal INCoDe.2030 – Esta iniciativa, que começou em 2017, pretende estimular e garantir o desenvolvimento de competências digitais como instrumento para a preparação de uma sociedade orientada para o futuro e para as novas oportunidades que surgem face à acelerada adoção das TIC (<http://www.incode2030.gov.pt>).

- eTwinning - Este projeto apresenta-se como sendo o maior da comunidade escolar da Europa e tem como principal objetivo a criação de redes de trabalho colaborativo entre as escolas europeias, com recurso às TIC, de forma a desenvolver o espírito de cidadania europeia (www.etwinning.net).
- Laboratórios de Aprendizagem (LA) – Iniciativa desenvolvida pela Direção-Geral da Educação, em parceria com a European Schoolnet, e consiste na disseminação de metodologias para a integração curricular das TIC que foram validadas em projetos piloto de âmbito europeu.
- APPS FOR GOOD - Programa educativo que está a ser implementado em Portugal desde 2014 e que pretende desafiar grupos de alunos, apoiados por professores, a assumir o papel de “Problem Solvers” e “Digital Makers”, ao longo de todo o processo de criação de uma App que contribua para a resolução de um problema da comunidade escolar (<http://erte.dge.mec.pt/apps-good>).
- UAC - Using Arduíno in the Classroom é um projeto que se iniciou em 2018 e pretende dinamizar o desenvolvimento de Kits de Aprendizagem Arduíno, destinados a contextos de aprendizagem das ciências, tecnologias, engenharias, artes e matemática, de âmbito disciplinar ou interdisciplinar (<http://www.ccems.pt/projeto/uac>).
- EduScratch – Projeto que tem como meta principal promover a utilização do Scratch em ambiente escolar (<http://eduscratch.dge.mec.pt>).
- CPR – Este projeto refere-se à rede nacional de Clubes de Programação e Robótica, iniciada em 2014, que se baseia no facto dos vários tipos de linguagem de programação estarem a ganhar uma importância crescente no mundo atual, sendo fundamentais, não só na área das CTEM (Ciências, Tecnologia, Engenharia e Matemática), mas também noutras áreas, por auxiliarem no desenvolvimento de capacidades transversais, tais como o pensamento analítico, a resolução de problemas, o trabalho colaborativo e a criatividade (<http://erte.dge.mec.pt/clubes-de-programacao-e-robotica>).
- Conta-nos uma história! – Pretende incentivar os estabelecimentos de Educação Pré-Escolar e do 1.º Ciclo do Ensino Básico a desenvolverem projetos, relacionados com o ato de contar, que coloquem os alunos em situações educativas desafiantes, recorrendo à utilização das TIC, nomeadamente

tecnologias de gravação digital de áudio e vídeo (<http://www.erte.dge.mec.pt/concurso-conta-nos-uma-historia>).

- MILAGE Aprender + - Apoiado pela DGE desde 2016, este projeto, pretende estender o ambiente de aprendizagem da sala de aula tradicional para uma sala de aula virtual, num sistema de aprendizagem misto (blended-learning) que combina aulas presenciais com aulas online, com o objetivo de manter os alunos motivados para aprender matemática e outras disciplinas através da exploração motivadora de ferramentas diversas, suportadas pelas tecnologias de informação e comunicação (<http://milage.ualg.pt>).
- Sitestar – É uma iniciativa nacional que já se encontra na sexta edição e que possibilita/convida a comunidade educativa a desenvolver sites em português e sob o domínio .pt, durante o ano letivo (<https://www.sitestar.pt>).
- ArdRobótic - É uma iniciativa promovida pela ANPRI nas áreas da programação, robótica, placas eletrónicas, internet das coisas, drones e outros artefactos e pretende promover a criação de ambientes de aprendizagem que integrem as tecnologias com metodologias ativas, além de estabelecer articulações com os conteúdos das áreas curriculares e/ou transversais (<http://www.anpri.pt>).
- 3Digital – Também promovida pela ANPRI, esta iniciativa na área da modelação 3D, estimula a participação dos alunos na conceção e desenvolvimento de projetos nesta área, estimulando assim a utilização das TIC (<http://www.anpri.pt/3digital>).

Obviamente que a implementação destes projetos e a integração no plano anual de atividades das escolas/agrupamentos ou a criação de clubes depende fortemente das convicções/opções tomadas pelo grupo de professores de informática presente nas escolas, bem como dos possíveis apoios das Direções dos estabelecimentos educativos.

Em termos de pensamento computacional, atualmente, existem várias abordagens e uma grande variedade de tecnologia ao dispor dos alunos e professores, permitindo a criação/desenvolvimento de várias aplicações, mas apenas destinada a um leque muito reduzido de alunos. É o que acontece através do ensino de linguagens de programação a alunos do ensino secundário ou através de atividades/clubes destinados aos mais novos em momentos pontuais.

Todavia, tal como destacado por Ramos e Espadeiro (2016), tem havido evoluções consideráveis nos últimos anos, uma vez que,

em Portugal, um número significativo de escolas do ensino básico e secundário tem vindo a desenvolver iniciativas e projetos com o objetivo 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. (p.4)

Sobre a iniciativa em causa, vários estudos já efetuados podem corroborar conclusões, tais como “uma apreciação geral muito positiva relativamente ao grau de consecução dos objetivos da IP1 por parte dos respondentes: diretores, responsáveis ou coordenadores da Iniciativa nas escolas” (Ramos & Espadeiro, 2016, p. 9), que vêm dar sustentabilidade à sua prossecução da iniciativa.

Nos nossos dias, podemos também ver que existem ensejos para as nossas escolas progredirem nesta conjuntura, já que podem optar por ter a área da programação como oferta complementar no primeiro ciclo e, em 2021, os alunos dos 2.º e 3.º ciclos terão Tecnologias de Informação e Comunicação (TIC) em todos os anos letivos. Não obstante, já estão disponíveis as candidaturas ao projeto GEN10S Portugal, promovido pela SIC Esperança, em parceria com a associação espanhola Ayuda en Acción e a Google.org, sendo um dos objetivos do mesmo "assegurar que os alunos tenham uma aprendizagem contínua na área das tecnologias e programação" (Lusa, 2017), o que roborará a importância da aposta na formação em termos de pensamento computacional.

Por fim, outro facto que tem acompanhado/potenciado o ensino das TIC é a utilização de ambientes de aprendizagem apoiados pela tecnologia. Entre os ambientes que permitem o ensino, a aprendizagem fora do tempo e do espaço escolar e o desenvolvimento da autonomia, está a plataforma Moodle, a mais adotada em Portugal (Pedro et al., 2008). Em geral, neste momento, todas as escolas dispõem de uma página ou portal para a sua divulgação, havendo mesmo as que, recentemente, estão a trabalhar com o auxílio da Plataforma Office365.

Esta atualização tecnológica, que influenciou a rapidez de resposta, tornou-se fundamental nos desafios criados pela pandemia originada pelo Covid-19, que obrigou ao “fecho” das escolas/aulas presenciais e levou à implementação de um ensino à distância que, à partida, não podia ser unidirecional, do professor para o aluno,

ignorando que a criança tem de ter voz ativa no processo, como aconteceu através da telescola. Nunca o recurso aos meios eletrónicos e digitais foi tão necessário, levando as escolas a implementarem/explorarem ferramentas/plataformas. Destaca-se que o recurso ao correio eletrónico, através de e-mail institucional ou dos diferentes serviços gratuitos (Gmail, Hotmail, Yahoo, Aol...), o recurso à videoconferência (Zoom, Teams, Meet, Webex, TeamSpeak, Jitsi, OpenMeetings...) ou o contacto através de outras aplicações de trocas de mensagens com recurso à internet (Whatsapp, Skype, Facebook Messenger, Viber...) foram uma constante fazendo parte do dia a dia dos professores e alunos ao longo do 3.º período, onde os conhecimentos das TIC foram fundamentais.

Tal como refere Oliveira (2018) na conclusão de um estudo feito em Portugal, os ambientes apoiados pela tecnologia e estratégias que envolvam alunos, professores e a utilização do Moodle ou outra ferramenta digital podem ser bastante enriquecedores para a aprendizagem e para o sucesso escolar, nomeadamente no desenvolvimento de competências autorregulatórias.

Este aspeto foi confirmado pela maior facilidade dos alunos em reportarem ou em mencionarem mais as crenças de autoeficácia, estratégias de monitorização e os motivos conducentes ao sucesso e ainda pela sobrevalorização da forma como planificam, desempenham e monitorizam, e refletem e avaliam o seu trabalho. (p. 207)

O Pensamento Computacional

O termo pensamento computacional esteve, durante muitos anos, apenas destinado aos cientistas da computação e para a maioria das pessoas, a programação é vista como uma ação complexa, que se baseia em conhecimentos tecnológicos complexos e, por isso, apenas acessível a alguns. Contudo, vários autores defendem uma ideia contrária. Papert (1980) refere que as linguagens de programação devem ter diferentes níveis, do simples ao complexo, tornando-se acessível a todos.

Os próprios conceitos de computação e informática são frequentemente confundidos/relacionados. Segundo Computing at School Working Group (2012), a ciência da computação pretende entender e explorar o mundo que nos rodeia, natural e artificial, em termos computacionais, dando especial atenção ao estudo, projeto, implementação de sistemas e compreensão dos princípios subjacentes a esses projetos.

Por sua vez, a informática está centrada na utilização de diferentes sistemas para resolver problemas do mundo real, incluindo questões como a instalação e uso de hardware e software.

Em 2006, o pensamento computacional começou a despertar o interesse da comunidade científica e educativa, após Wing (2006) ter defendido o seu uso e adoção por todos os cidadãos, incluindo jovens e crianças. De acordo com a mesma autora, o pensamento computacional vai desde a estruturação do raciocínio até à resolução de problemas; procura encontrar a solução de problemas, projetar sistemas e compreender comportamentos, baseando-se nos conceitos fundamentais da ciência da computação. Defende ainda que o pensamento computacional inclui uma gama de ferramentas mentais que estão, sobretudo, presentes na ciência da computação. Em termos de educação, considera o pensamento computacional como um conceito crucial que envolve a resolução de problemas, conceção de sistemas e compreensão do comportamento humano, baseados nos princípios das ciências da computação.

Para Phillips (2008), a essência do pensamento computacional é o pensar em dados e ideias, usar e combinar esses recursos para resolver problemas. Mas, ao contrário do que normalmente se pensa, o pensamento computacional não se deve reduzir à ideia de tecnologia, equipamentos, redes, etc. O mesmo autor chega mesmo a identificar alguns aspetos que fundamentam essa expansibilidade do pensamento computacional, referindo que o mesmo: a) não é apenas mais detalhes técnicos para a utilização de um determinado software; b) não é pensar como um computador; c) não é programação (necessariamente); d) nem sempre requer um computador; e) não é mais uma “coisa” para acrescentar ao currículo.

Encontrar uma definição exata e concreta do pensamento computacional, para toda a comunidade científica, revelou-se uma tarefa difícil, tal como mostram os resultados obtidos nos workshops “The Scope and Nature of Computational Thinking” realizados nos Estados Unidos da América e patrocinados pela National Academy of Sciences, onde a presença de investigadores de diferentes áreas levou a concluir que não era possível chegar a um consenso sobre o conteúdo preciso de pensamento computacional e, muito menos, da sua estrutura. Nessa mesma altura, admite-se que essa falta de desacordo explícito poderia ser entendida como refletindo uma intuição, compartilhada entre os participantes do workshop, de que o pensamento computacional seria um modo de pensamento com o seu próprio carácter distintivo (National Research Council, 2010).

Vários autores, como Isbell et al. (2009) , Barr e Stephenson (2011) ou Lee et al. (2011) basearam os seus trabalhos no pensamento computacional, apresentando em comum um forte relacionamento com a abstração de um processo que permite a sua autonomia e referindo ainda que essa abstração pode ser adquirida na recolha e análise de dados para a seguir ser representada através de um computador em formato de algoritmo.

No entanto, mais uma vez, não houve uma tendência clara e nem todos seguiram a mesma linha de pensamento. Basawapatna et al. (2011) apresentaram uma definição de pensamento computacional através de padrões de funcionalidades que foram identificados por alunos com a utilização de linguagens de programação. Uma particularidade da definição destes autores é o facto de identificarem competências que podem ser transferidas para outros domínios.

Não demorou muito tempo para que o pensamento computacional estivesse presente nas escolas. Em 2012 a introdução do pensamento computacional na escola já era uma proposta suportada pela comunidade científica e educativa e um tópico relevante na discussão sobre as competências que os jovens deviam adquirir ao longo da sua escolaridade, tendo em vista os cenários de futuro no que diz respeito ao desenvolvimento social e económico (Charlton, P., & Luckin, 2012).

Alguns países tiveram uma posição mais astuta e futurista e fizeram a introdução do pensamento computacional no ensino, de diferentes formas, mas com o mesmo propósito.

Assim, e segundo o site da Forbes (Forbes, 2012), na Estónia foi criado, em 2012, um programa intitulado de ProgeTiger ,destinado a alunos com idades que variavam entre os 7 e os 19 anos, com a principal finalidade de ensinar programação. Este programa piloto foi implementado em 550 escolas desse país e a sua coordenadora destacou o facto de para a faixa etária mais baixa o programa se ter focado mais no ensino da lógica da programação, potenciando a integração da robótica. Nota-se, portanto, que houve a preocupação de possibilitar aos alunos mais novos uma maior exploração dos conceitos gerais do pensamento computacional.

A National Science Foundation deu também o seu contributo e adotou uma perspetiva que se baseou em sete premissas da computação: a) a computação é uma atividade humana criativa; b) a abstração restringe informações e detalhes para se concentrar em conceitos importantes para a compreensão e resolução de problemas; c) os dados e informações têm como objetivo facilitar a criação de conhecimento; d) os algoritmos são

instrumentos para desenvolver e revelar soluções para problemas computacionais; e) a programação é um processo criativo que cria artefactos computacionais; f) os dispositivos digitais, os sistemas e as redes que os interligam permitem estimular abordagens computacionais para resolver problemas; g) a computação permite a inovação noutras áreas, incluindo as ciências, as ciências sociais, as humanidades, as artes, a medicina, a engenharia e os negócios (Grover & Pea, 2013).

Por sua vez, Selby e Woollard (2013) definem o pensamento computacional como um processo cognitivo ou pensamento que engloba o raciocínio lógico através do qual os problemas são solucionados; no entanto, também referem que o mesmo não se limita à solução de problemas, acrescentando que:

It is a cognitive or thought process that reflects

- the ability to think in abstractions,
- the ability to think in terms of decomposition,
- the ability to think algorithmically,
- the ability to think in terms of evaluations, and
- the ability to think in generalizations. (p. 5)

Em sùmula, esta definição esclarece que o pensamento computacional, mais uma vez, está focado na solução de problemas, integrando processos de pensamento que utilizam interação, decomposição, algoritmia, avaliação e generalizações.

O cenário de inclusão na educação aparece novamente bem patente por Wing (2014), que destaca a utilização de diferentes estratégias e abordagens, incluindo diferentes tecnologias e aplicações. Porém, apesar das divergências, parece ser consensual que o pensamento computacional deve ser introduzido no processo de ensino-aprendizagem.

Diversos autores têm procurado identificar os componentes do pensamento computacional. Por exemplo, Zapata-ros (2015), apresentou uma pesquisa onde faz a interligação desde os conceitos e modalidades do pensamento computacional até às teorias da aprendizagem, definindo 14 componentes, com fortes ligações entre eles.

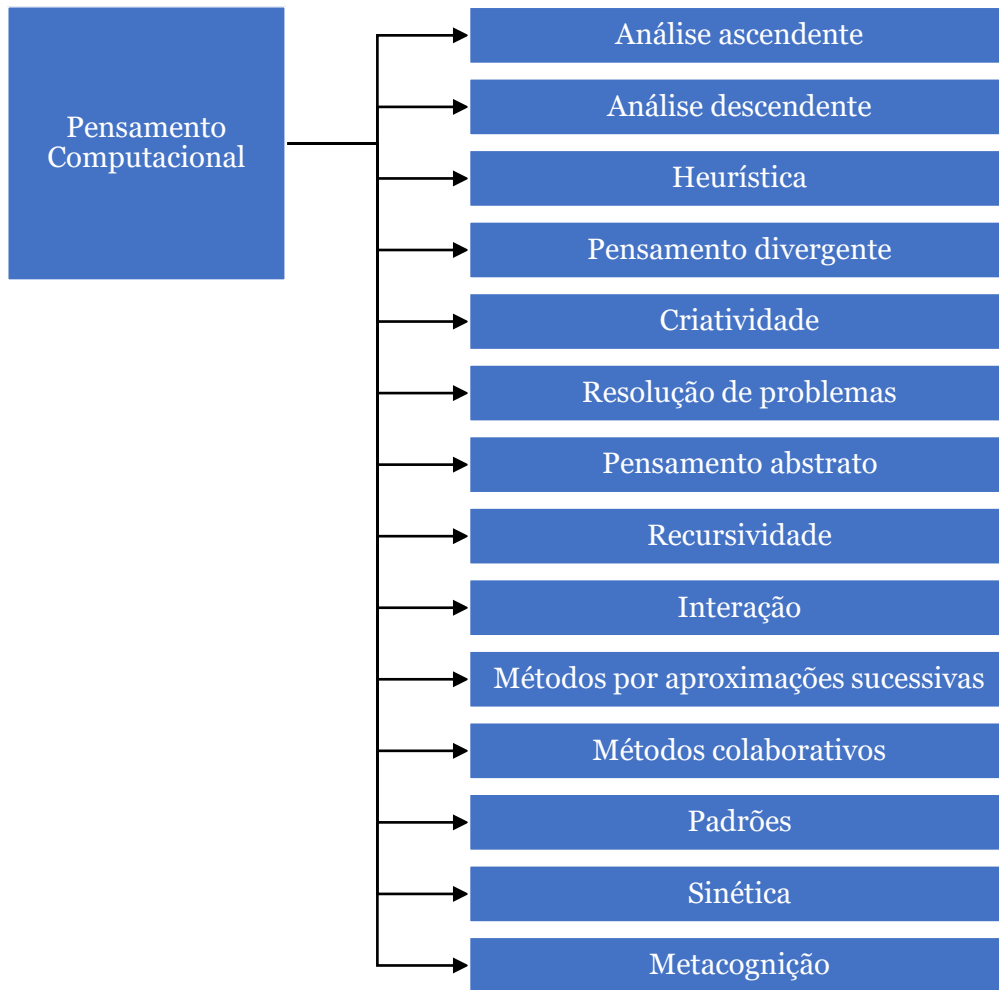


Figura 1. Esquema de componentes do pensamento computacional (Zapata-ros, 2015, p. 13)

Em termos gerais, o autor apresentou os seguintes componentes:

- **Análise descendente** – onde a obtenção de um método geral pode originar a implementação de um conjunto de métodos de menor dimensão e de diferentes tipos;
- **Análise ascendente** – na presença de um problema complexo, primeiro solucionar os problemas mais concretos e, depois, os mais abstratos;
- **Heurística** – com foco nas técnicas baseadas na experiência, é com as regras metodológicas que se evitam dificuldades para resolver problemas e se formulam hipóteses para a sua resolução.

- **Pensamento divergente** – através das diversas formas de pensar/idealizar que não correspondem ao padrão, conseguindo-se evitar a inércia resultante de ideias comuns ou normalmente aceites que limitam as soluções apresentadas.
- **Criatividade** – O autor refere-se a este componente como sendo a psicologia da descoberta e a invenção que é capaz de produzir uma novidade aceitável utilizando duas formas de pensar opostas: convergentes e divergentes.
- **Resolução de problemas** – baseia-se na visão que pode ser desenvolvida nos alunos, e implementada em sala de aula, de como encontrar soluções para os diversos problemas através da utilização de um computador, tornando-os não apenas simples utilizadores, mas autores e construtores das ferramentas para solucionar o problema.
- **Pensamento abstrato** – apresenta-se como a capacidade de trabalhar com modelos abstratos. Obtido o modelo, são estudadas as suas propriedades e estabelecidas as regras/conclusões que permitem prever os comportamentos dos objetos. Este pensamento está fortemente ligado ao pensamento matemático e áreas envolventes.
- **Recursividade** – por vezes, um determinado problema não pode ser resolvido, ou pelo seu tamanho ou por outro fator não contornável, mas pode ser encaminhado para outro problema com as mesmas características ou natureza, de menor dimensão ou com fatores mais favoráveis.
- **Interação** – este componente refere-se a uma maneira de pensar que deve ser explorada no pensamento computacional, mais concretamente em termos de procedimentos complexos que incluam construção de algoritmos repetitivos.
- **Métodos por aproximações sucessivas** – refere-se à exploração da tentativa e erro e ao confronto das ideias formadas com a realidade. Este autor recorre a Karl Popper (1934), entre outros, para enquadrar este componente e destacar a importância de existir ajuda pedagógica por parte do professor através duma forte orientação no processo de tentativa e erro, para que os alunos não desmotivem ou se desenquadrem das aprendizagens pretendidas.
- **Métodos colaborativos** – baseiam-se na exploração duma cultura comum, referências e experiências que fazem o trabalho colaborativo fluir sem se apoiar apenas em técnicas para trabalhar em conjunto.

- **Padrões** – os padrões são uma ferramenta para a análise de programação com dupla singularidade: evitam o trabalho repetitivo que envolve, por exemplo, a repetição de partes de código quando aplicados a diferentes contextos/ situações, e por outro lado, exigem a capacidade de distinguir onde eles podem ser úteis. Em suma, o padrão consiste em diretrizes comuns que permitem resumir e comunicar a experiência acumulada e aplicá-la na resolução de problemas.
- **Sinética** – apresenta-se com um ponto de convergência entre a criatividade e as técnicas de trabalho em grupo. Segundo o autor, a sinética está presente na fase inicial das ideias, como, por exemplo, através do "brainstorming".
- **Metacognição** – baseia-se no conhecimento dos processos de pensamento e de aprendizagem de cada um de nós, em aprender novas técnicas e, principalmente, treinar/desenvolver as próprias capacidades cognitivas.

Em geral, pode-se verificar que este autor considera a codificação como uma competência complexa, subdividindo-a num conjunto de habilidades que envolvem os catorze componentes do seu esquema, destacando sempre que, para uma aquisição completa do pensamento computacional, nenhum componente pode ser esquecido e deixando sempre em aberto a possibilidade de estes componentes serem ainda complementados.

Parece óbvio que a identificação dos componentes do pensamento computacional está amplamente relacionada com o método de pensar a forma de resolução de problemas com o recurso ou não de tecnologia.

Facto é que, muitas vezes, as tecnologias digitais estão presentes na forma como se resolvem alguns tipos de problemas e até mesmo na estimulação deste tipo de pensamento. Além disso, frequentemente encontramos documentados os benefícios do uso das diversas tecnologias na resolução de problemas e na estimulação do raciocínio.

Segundo Kurshan (2016), o pensamento computacional pode ser definido como a:

capacidade criativa, crítica e estratégica humana de saber utilizar os fundamentos da informática nas mais diversas áreas do conhecimento, com a finalidade de identificar e resolver problemas, de maneira individual ou

colaborativa, através de passos claros, de tal forma que uma pessoa ou uma máquina possam executá-los eficazmente. (p. 2)

Com o aumento do número de linguagens de programação gratuitas, cada vez mais “user friendly”, mais próximas da forma humana de representar a linguagem, mais fáceis de compreender e começar a utilizar, como o Scratch (Resnick et al., 2009) ou o Alice (Graczyńska, 2010), o interesse pela introdução da programação nos níveis pré-universitários também aumentou.

Neste sentido, nos últimos anos, as escolas do ensino básico e secundário, sobretudo em países anglo-saxónicos, têm vindo a desenvolver iniciativas com o objetivo de introduzir o pensamento computacional, seguindo as recomendações de várias organizações, tais como a “Google For Education”.

Valente (2016) identificou alguns exemplos onde o pensamento computacional pode ser explorado na educação, nomeadamente: a) atividades sem recurso a tecnologias; b) programação em Scratch; c) robótica pedagógica; d) produção de narrativas digitais; e) criação de jogos e f) uso de simulações. O mesmo autor considera que o foco não deve ser a programação nem a inclusão de atividades ou de disciplinas específicas, destacando que o pensamento computacional pode ser trabalhado em conjunção com as disciplinas já existente no currículo, dando como exemplo o ensino em Itália. No entanto, alerta para o facto de

o grande desafio dos trabalhos nessa área é o fato de ainda não existir consenso entre os pesquisadores, tanto da Ciência da Computação quanto de pesquisadores da área de tecnologias na educação, sobre em que consiste o pensamento computacional. Sem essa definição é bastante difícil estipular como esse tema pode ser abordado na educação, como formar educadores para essa atividade e como avaliar o aluno. (p. 892)

Já foram elaborados alguns estudos que relacionam o pensamento computacional com outras áreas, até porque a introdução do pensamento computacional na escola e no currículo tem sido um desafio aceite por todos os agentes educativos e em todos os ciclos do sistema educativo.

Vários autores chegam mesmo a considerar que, tal como a matemática é uma das ferramentas mais utilizadas na resolução de problemas complexos, a computação deveria ser encarada como outra ciência fundamental, uma vez que está na base de tudo. Todavia, não a consideram como uma ferramenta, antes como uma ciência para a solução de problemas em inúmeras áreas. (Sobral et al., 2015)

A formação dos professores, principalmente a contínua, é certamente um dos pontos de maior impacto para o aparecimento do pensamento computacional nas diversas áreas.

A introdução do pensamento computacional na formação dos futuros professores tem sido fundamentada pelo facto de o poderem integrar, mais facilmente, no currículo quando reconhecem e identificam as suas potencialidades na resolução de problemas e conseguem identificar estratégias pedagógicas para o incluir na sua própria área curricular, ou seja, na disciplina que se está a lecionar (Zhou, 2011).

Serafini (2011), apresentou um estudo, realizado na Suíça, com crianças entre os 8 e os 13 anos, onde, com a utilização de um robô (objeto de aprendizagem), se faz uma introdução ao ensino da programação e do pensamento computacional, acreditando-se que esta é uma competência essencial e transversal a todas as áreas do saber.

Nunes (2011) refere, por exemplo, os advogados, que, através do pensamento computacional e após a leitura dos respetivos documentos, podem extrair factos e regras, que permitem tirar conclusões lógicas que fundamentam um parecer irrefutável. De notar ainda que procedimentos desta profissão, como, por exemplo, organizar uma eleição, podem ser sistematizados na forma algorítmica, de modo a refletir processos cognitivos de resolução de problemas.

Em 2012, o pensamento computacional foi classificado como uma aptidão elementar no século XXI, uma vez que resulta no aumento da capacidade analítica das crianças nas diversas áreas do conhecimento (Brennan & Resnick, 2012).

Em Portugal, Carmo (2013) realizou um estudo com o tema “A robótica Educativa no Desenvolvimento do Raciocínio matemático” e referiu que,

os resultados do estudo destacam que programar um sistema robotizado, controlando-o a partir de um computador com softwares especificados, permite ao aluno assumir um papel ativo na construção dos seus próprios conhecimentos

matemáticos, reconhecendo a importância de refletir sobre as decisões tomadas, aprendendo com os erros, tentando evitar repeti-los. (p. 33)

Também Shailaja e Sridaran (2015), nos seus estudos, destacaram a necessidade de ensinar o pensamento computacional o mais cedo possível e concluíram que os alunos com um pensamento computacional mais desenvolvido têm um bom desempenho e podem analisar, avaliar e resolver melhor os problemas na escola, no trabalho e na vida em geral.

No Brasil, recentemente, Garofalo (2018) referiu que a robótica, considerando como principal objetivo desenvolver o raciocínio e a lógica na construção de algoritmos e programas, está a crescer nas escolas brasileiras, uma vez que estimula a criatividade nos diferentes segmentos da construção do conhecimento.

Na sequência do referido é claro o quanto a estimulação do pensamento computacional tem um impacto positivo em diversas áreas, podendo trazer diversos benefícios para o processo ensino-aprendizagem.

Autoeficácia

O constructo autoeficácia foi introduzido na psicologia cognitiva por Albert Bandura, em 1986, através da sua publicação *Social Foundations of Thought and Action: A Social Cognitive Theory* e refere-se às percepções ou expectativas dos indivíduos sobre as suas capacidades para aprender ou desempenhar ações antes de serem realizadas, ou seja, são as apreciações das pessoas sobre as suas próprias capacidades para alcançarem os objetivos desejados.

O indivíduo, na perspetiva social cognitiva, é capaz de ser pró-ativo, autorregular os seus pensamentos, auto-organizar-se, autorefletir e autoavaliar, em vez de ser modelado pelos contextos sociais (Pajares, 2006).

Pajares (2006) refere ainda uma forte ligação entre a motivação e a autoeficácia, visto que, a não ser que os indivíduos acreditem que são capazes de desenvolver ações que produzam os efeitos desejados, a motivação será baixa para executar a ação e não serão persistentes face às dificuldades encontradas.

Por sua vez, os indivíduos com percepções de autoeficácia elevadas encaram os obstáculos como oportunidades para o seu desenvolvimento e não como um impedimento, sendo, por isso, persistentes e esforçados (Bandura, 2001).

Neste seguimento, a autoeficácia afeta não só a aquisição de novos comportamentos como também a inibição ou desinibição de comportamentos já existentes (Amaral, 1993). De realçar que, em diversas meta-análises, a autoeficácia, mesmo em diferentes ambientes e populações, tem surgido como um forte preditor de motivação, persistência e desempenho ao longo do tempo (Chemers et al., 2001).

Costa (2002) confirma este constructo ao afirmar que a “percepção de auto-eficácia interfere na autoregulação da motivação na medida em que serve de base para a pessoa estabelecer seus alvos e desafios, perseverar em seus esforços, reajustar seus padrões internos àquilo que conseguiram realizar.” (p. 50)

Neste sentido, tal com Valle et al. (2003) referiram, a análise das crenças em termos de autoeficácia dos alunos é fundamental para compreender as suas ações, uma vez que revelam não o que o sujeito é capaz de realizar, mas aquilo que acredita que pode realizar.

Poderá também concluir-se, tal como Martínez e Soria (2006) sublinharam, que as crenças de eficácia são construídas através de juízos acerca das capacidades que cada individuo possui. No entanto, com as mesmas capacidades, indivíduos com crenças diferentes podem obter sucessos ou fracassos diferentes em função dessas diferenças de crenças.

Esta opinião foi também defendida por Hoy e Spero (2005), ao referirem que as crenças de autoeficácia são apreciações orientadas para o futuro, relacionadas mais com a percepção de capacidade do que com a competência, sublinhando também que essa é uma distinção importante, pois, em geral, as pessoas superestimam ou subestimam as suas capacidades, e isso pode trazer efeitos para as suas ações.

Esta confirmação é igualmente feita por Zimmerman e Ringle (1981, citado por Bear, 2010), ao afirmarem que, perante uma atividade que consideram como mais difícil, os alunos com alta autoeficácia tendem a ser mais persistentes na sua compreensão e na realização do que os alunos com baixa autoeficácia.

Apesar da baixa autoeficácia afetar negativamente a aprendizagem, uma autoeficácia demasiado elevada pode não originar os melhores resultados, já que pode não conduzir à mobilização de esforço. As autoavaliações positivas conduzem a um aumento nas

perceções de autoeficácia e motivam as crianças a continuar a evoluir; no entanto, as autoavaliações negativas não são sinónimo de diminuição de autoeficácia se forem motivo de alteração de comportamento com vista ao sucesso. As autoavaliações não acontecem de forma espontânea - exercitar o treino de autoavaliação através do feedback, da definição de objetivos e da adoção de diferentes estratégias resulta, sim, num acréscimo da autoeficácia (Schunk & Zimmerman, 2007).

Bandura (1997, citado por. Neves & Faria, 2007) refere que estas observações acerca da relação positiva entre a autoeficácia académica e a concretização escolar se têm revelado estáveis ao longo de diversas investigações, pelo que aparentam poder ser generalizadas e aplicadas a todos os alunos, independentemente do género, da idade ou do seu ano de escolaridade.

Assim sendo, Hadwin et al. (2011) destacaram que as perspetivas contemporâneas de aprendizagem admitem que os alunos podem controlar ativamente a sua própria aprendizagem e os seus resultados.

Outro fator que também não deverá ser esquecido assenta na satisfação pessoal que domina o aluno na realização de uma determinada tarefa e não nas possíveis consequências ou recompensas externas que ele poderá vir a ter pela realização da mesma. França (2015) diz que

estudantes com objetivos orientados para a aprendizagem estão mais focados na evolução da aprendizagem do que na competição com os seus pares pela obtenção de determinados resultados escolares e, normalmente, são mais eficazes na sua aprendizagem do que aqueles que apresentam objetivos centrados no resultado.
(p.30)

A perspetiva de Bandura sobre a aprendizagem é de grande pertinência na sociedade atual, pois, além da aquisição de conhecimento num sentido cognitivamente reativo, a aprendizagem engloba o desenvolvimento de crenças autorreferentes e de capacidades autorregulatórias de quem aprende (Vieira, 2012).

Guimarães e Abbad (2015), ao realizarem uma análise da literatura internacional sobre a autoeficácia no uso do computador (AEC) em situações de aprendizagem, concluíram que os resultados “evidenciaram relações positivas entre a AEC e o desempenho de

aprendizes durante e após o curso ou treinamento, no que se refere à aprendizagem, à transferência e à aplicação de novos conhecimentos, habilidades e atitudes” (p.184).

Também Pereira e Celis (2018) referem que os resultados de várias investigações têm valorizado o papel da autoeficácia no sucesso escolar, ressaltando, contudo, que o desempenho dos indivíduos carece de uma interação positiva entre variáveis: só depois de assegurar que o aluno detém conhecimento, habilidades e capacidades, além de expectativas positivas de resultados, e que esses resultados são valorizados, é que as crenças de autoeficácia têm o poder de motivar os alunos, porque é em função delas que sucederão a escolha, a direção e a persistência nos comportamentos de aprendizagem.

Por sua vez, Durak et al. (2019) ao avaliarem a autoeficácia da programação, do pensamento computacional, entre outras variáveis, em atividades relacionadas com a robótica e respetiva programação, observaram que os resultados estão relacionados com a escolaridade dos alunos e que existe uma relação positiva entre os resultados de pensamento computacional, autoeficácia de programação e pensamento reflexivo.

Esta linha de pensamento também é confirmada por Türker e Pala (2020), através dos resultados obtidos sobre o efeito da educação em algoritmos nas perceções de autoeficácia da programação de computadores e no pensamento computacional, onde concluíram que a educação por algoritmos aumenta a percepção de autoeficácia dos alunos na área da programação e que este aumento leva os alunos a começar a programar com perspetivas mais positivas.

Ensino da Programação

O ensino da programação tem tido um crescente interesse ao longo dos anos, sendo esta competência destacada por diversas personalidades. A propósito, Obama (2013), por exemplo, numa mensagem de forte incentivo, acentuou: Não compre, apenas, um jogo, crie um. Não se limite a fazer download de uma nova aplicação, ajude a desenvolvê-la. Não jogue no seu telemóvel, programe-o.

No entanto, mesmo em disciplinas introdutórias do ensino superior, o ensino da programação tem-se apresentado como uma tarefa difícil e muitas vezes fracassada. Gomes et al. (2004) referem que “o insucesso generalizado verificado na aprendizagem de programação, por parte de alunos de qualquer grau de ensino, é uma realidade” (p.

187). Neste sentido, têm sido apresentadas diversas estratégias para colmatar este problema. Piva Jr e Freitas (2010) apresentam as suas estratégias focadas na codificação da mensagem, destacando a capacidade de abstração e o reconhecimento dos detalhes para a resolução de diversas situações. No entanto, e concentrando as estratégias na faixa etária desta investigação, Maloney et al. (2008), em considerações finais dos seus estudos com crianças/jovens dos 8 aos 18 anos, indicam como estratégia o uso de ambientes mais didáticos e interativos:

Youth interest in technology starts with digital media and might thus serve as a more promising pathway into programming. The broad spectrum of media designs – from video games to music videos and greeting cards – is a true indicator of youth’s interest in not only being user of digital media (as they do on a regular and personal basis) but in going beyond mere consumption to become content creators themselves, a role often denied to urban youth.(p. 371)

Apesar dos bons resultados, destes e doutros contributos, as dificuldades ainda são acentuadas, sendo necessário considerar o ritmo diferenciado de aprendizagem do aluno, que leva à inexistência de uma solução uniforme para a generalidade dos mesmos, bem patente nos resultados controversos em estudos que aplicam as mesmas estratégias, nas mesmas condições, mas a um grupo de alunos diferente (Gomes & Mendes, 1997).

A heterogeneidade do pensamento computacional, aparentemente presente nos alunos de programação, sobretudo nas primeiras atividades/anos iniciais, faz emergir, por parte do grupo/turma, uma maior necessidade de acompanhamento, orientação e esclarecimento de dúvidas por parte do professor, que, por diversos motivos, é impossível de satisfazer, facto que piora com a dimensão da turma. Outro fator, não menos importante, é que um ensino tradicional, através de conteúdos iguais/limitados para todo o grupo, não vai ao encontro das expetativas criadas por alguns alunos.

Segundo Jenkins (2002), a dificuldade de aprender a programar está associada a fatores cognitivos e porque não se pode ter só em conta uma habilidade ou um conjunto de habilidade, mas sim, uma hierarquia de aptidões. Já anteriormente, Ng e Bereiter (1991) referiam a programação como um conjunto de múltiplos processos.

Por sua vez, Gomes (2000) destaca a compreensão como a grande dificuldade em aprender a programar, mais concretamente, na criação de algoritmos para resolver os

problemas através das noções básicas, tais como as estruturas de controlo. A mesma autora salienta a necessidade de alternativas ao ensino/método tradicional e refere que,

entre os diversos motivos que levam os alunos a ter dificuldades e a solicitar ajuda, há que destacar os seguintes:

- incapacidade de estruturação dos passos de resolução de um problema;
- incapacidade de concepção de algoritmos;
- incapacidade de detectar erros de lógica de programação;
- falta de pré-requisitos sobre as matérias em questão e de técnicas de programação eficazes;
- conceitos errados sobre os recursos e técnicas utilizadas;
- problemas devido a efeitos colaterais não previstos;
- erros de sintaxe e advertências na compilação. (p.4)

A. Farias et al. (2014) seguem a mesma linha de pensamento e elencam como principais dificuldades: os erros de sintaxe e semântica, a compreensão do enunciado dos problemas e respetiva construção dos algoritmos, a deteção de erros de lógica e a falta de pré-requisitos sobre os conteúdos e as técnicas a utilizar.

A procura de soluções para ultrapassar estas dificuldades tem sido uma constante, mas uma delas, o recurso à utilização de animações, na compreensão e resolução de problemas computacionais, tal como referem Gomes et al. (2004), tem permitido bons resultados em diversos estudos, onde se verifica a eficácia da interatividade e da componente visual na aquisição de novos conceitos.

A sequência de como são abordados os conteúdos também pode dificultar ou facilitar a aprendizagem, e, tal como dizem Alves et al. (2019),

várias entidades têm realizado esforços para o desenvolvimento de diretrizes e currículos de ensino de computação na Educação Básica definindo o sequenciamento de conteúdos. Abordando especificamente o pensamento computacional, foi desenvolvido o K-12 Computer Science Framework nos Estados Unidos, o Computing at School no Reino Unido, o Australian

Curriculum, Assessment and Reporting Authority e no Brasil, a Sociedade Brasileira da Computação criou diretrizes para ensino de computação na Educação Básica. (p.2)

No entanto, e apesar de as propostas estarem diretamente associadas às ofertas educativas e serem bastante diversificadas, é notória a necessidade de estabelecer uma sequência dos conteúdos em relação à sua dificuldade de aprendizagem, não sendo possível abordar determinados conceitos sem que outros estejam bem solidificados.

Em termos mais concretos, Rich et al. (2018) analisam a sequência de três conceitos fundamentais - sequências, condições e repetições - no pensamento computacional, onde destacam a interligação com as experiências pessoais e conceitos matemáticos. Segundo os autores, a escolha da linguagem de programação influencia a escolha da sequência/detalhe dos conteúdos e facilita a aquisição de conteúdos, destacando as vantagens do Scratch.

Existem mais de 700 linguagens de programação, no entanto poucas são realmente utilizadas, sendo as mais populares a linguagem Java, C, C++, C#, PHP, Basic, Objective-C, Python, Perl, Lua, Javascript e Ruby (Zapalowski, 2011).

Perante esta diversidade, a escolha da linguagem acarreta alguma subjetividade e não existem evidências de que alguma se destaque, como afirma Jenkins (2002):

Much has been written about the best language to use for teaching (or learning), and some might argue that too much has been written. There is scant solid evidence that any language is any better or any worse than any other, and the choice continues to be driven largely by the "flavour of the month" in industry. (p. 55)

Atualmente, existem diversas plataformas que ajudam no ensino/estimulação da programação, algumas delas lançadas recentemente, outras com fortes atualizações nos últimos anos, mas todas com um grande crescimento na última década. São bons exemplos disso os seguintes projetos/plataformas destinados, principalmente, às faixas etárias mais baixas que, através das suas linguagens de programação e ambientes gráficos próprios, fazem parte de um amplo conjunto de opções disponíveis para a comunidade educativa em geral:

- MIT App Inventor – Também conhecido por App Inventor for Android, iniciado pela Google em 2010, sendo neste momento responsabilidade do Massachusetts Institute of Technology. Teve o seu ponto forte em 2013, com o lançamento de uma versão que deu ênfase à edição por bloco. Atualmente, segundo a plataforma, existem mais de 400.000 utilizadores ativos, provenientes de 195 países que já criaram quase 22 milhões de aplicações.

Sendo uma plataforma de código aberto, com um ambiente de programação visual, permite a todos, incluindo alunos de uma faixa etária mais baixa, criar aplicações para telemóveis e tablets (<https://appinventor.mit.edu/>).

- Scratch – Esta linguagem de programação foi disponibilizada em 2007, sendo um projeto da Lifelong Kindergarten Group do MIT Media Lab. Destaca-se pela sua linguagem própria, que não necessita de conhecimentos prévios de outras linguagens, tornando-se uma forte possibilidade/ajuda para quem está a começar a aprender a programar. Em termos de público alvo, recomendam-se alunos a partir dos 8 anos, verificando-se que, entre os inscritos na plataforma, a grande incidência encontra-se entre os 8 e os 19 anos. Os principais objetivos passam por criar histórias, jogos e outras aplicações interativas.

O Scratch, atualmente na versão 3.0 lançada em 2019, tem desde 2013, através da versão 2.0, a possibilidade de ser utilizado inteiramente on-line. No entanto, manteve sempre a possibilidade de download e instalação, o que, para as escolas com infraestruturas limitadas ou com problemas de ligação à internet é uma característica bastante importante. De realçar, também, que as versões anteriores, que necessitam de menores recursos, continuam disponíveis. Hoje, segundo a plataforma, esta é usada em mais de 150 países, em mais de 40 línguas e contando com mais de 56 milhões de utilizadores registados, tendo um forte crescimento a partir do final de 2019 (<https://scratch.mit.edu/>).

- Code.org – Sendo uma organização sem fins lucrativos é apoiada por muitas entidades, destacando-se Amazon, Facebook, Google, a Infosys Foundation, Microsoft entre outras, com fortes influências na área das TIC. A plataforma code.org, lançada em 2013, pelos dois irmãos Hadi e Ali Partovi, teve de imediato um forte impacto a nível mundial, graças ao seu vídeo promocional, com mais de 15 milhões de visualizações, onde diversas personalidades, tais como, Bill Gates (fundador da Microsoft), Jack Dorsey (criador do Twitter), Mark Zuckerberg (fundador do Facebook), Drew Houston (fundador da DropBox), Gabe Newell

(diretor da Valve Corporation), Chris Bosh (ex-jogador da National Basketball Association), Will.i.am (cantor e membro dos Black Eyed Peas) defendem, publicamente, as potencialidades da programação.

Esta plataforma disponibiliza cursos on-line que estão divididos pelas diferentes faixas etárias, sendo o primeiro curso para crianças entre os 4 e os 6 anos, que estão a aprender a ler. Os cursos são constituídos por diversas aulas on-line, com exercícios de programação por blocos e aulas off-line, onde são aplicados algoritmos simples em atividades do dia a dia. Esta plataforma também é conhecida pelas atividades disponibilizadas no âmbito da “Hour of Code” e por disponibilizar a opção de o professor poder acompanhar a evolução dos alunos e fazer uma seleção na disponibilização das aulas/exercícios (<https://code.org/>).

- Ubbu - code literacy - É uma plataforma portuguesa, criada pela Academia de Código, com o objetivo de preparar crianças, dos 6 aos 12 anos, para que dominem a programação, o pensamento lógico, a resolução de problemas e sejam cidadãos e cidadãs conscientes.

Esta plataforma surge no início de 2019, no seguimento do projeto Academia de Código Júnior, que nasceu em 2015. No ano letivo 2019/2020 teve quase 50 mil alunos inscritos. Está disponível em 3 línguas (português, inglês e espanhol), com planos de aulas específicos para cada ano de escolaridade, e com ambientes diferenciados para alunos e professores. O processo inicia-se com a associação dos alunos ao professor/grupo, mas o aluno apenas tem acesso às aulas/conteúdos após o professor dar a sua permissão/disponibilidade (<https://www.ubbu.io/>).

- Kodu – Este software, um ambiente de programação produzido e mantido pela Microsoft, está disponível desde 2009. Permite criar jogos em 3 dimensões através de uma linguagem de programação visual e simples. Existe a obrigatoriedade de instalar a aplicação, no entanto, estão disponíveis diversos “mundos”, uma vez que a plataforma é colaborativa e os utilizadores podem criar os seus “mundos” e disponibilizá-los para download.

Os utilizadores desenvolvem o seu raciocínio lógico quando dão instruções, através de comandos, às suas personagens para se movimentarem, assim como quando programam pontuações, entre outros desafios disponíveis, tais como a

programação da interação entre as personagens ou a criação de objetivos específicos para o jogo (<https://www.kodugamelab.com/>).

- Tinkercad – Apesar de ter sido disponibilizada em 2011, pela Autodesk, como uma ferramenta de modelação 3D, a partir de 2017 incorporou um módulo de circuitos eletrónicos, que fazia parte do Circuits.io, que entretanto foi descontinuado pela empresa. Neste momento a plataforma, totalmente on-line, permite, além da modelação 3D, a criação e simulação de circuitos elétricos. Permite também a programação através de blocos.

A plataforma torna possível uma interação personalizada entre o aluno e o professor, além de disponibilizar uma vasta biblioteca de conteúdos/exemplos (<https://www.tinkercad.com/>).

- mBlock - O mBlock, lançado em 2014, pela empresa Makeblock, apresenta-se como uma ferramenta de fácil acesso para a programação de robôs. Em termos gerais, este software é uma ramificação open source do Scratch (anteriormente apresentado). Além de dar a possibilidade de programar numa linguagem de blocos, muito semelhante ao Scratch, também possibilita programar e simular placas Arduino, muito utilizadas em circuitos eletrónicos e na área da robótica, sem a necessidade de conhecimentos em linguagens como o Java ou C++.

Tendo sempre a necessidade de ser instalada, torna-se uma boa ferramenta para as escolas com clubes de robótica, pois facilmente ensinam a programar, por exemplo, a predefinir um movimento do robô, ou simplesmente acender, de forma intermitente, um led, através desta ferramenta (<https://www.mblock.cc/>).

- Grasshopper – Disponibilizada em abril de 2018, pela Google, tem com objetivo ensinar a programar. Totalmente gratuita, apresenta uma interface visual que, sendo on-line, não necessita de qualquer instalação, apesar de também já estar disponível a versão para instalação. Esta plataforma tem a particularidade de trabalhar diretamente com linhas de código, o que poderá ser menos adequado a faixas etárias mais baixas (<https://grasshopper.app/>).

A Google também mantém disponível a Blockly, que ainda se encontra na versão Beta, e disponibiliza um percurso de oito pontos com diversas atividades onde a dificuldade vai sendo crescente, destacando-se a sua interface simples que ajuda a integração de alunos de faixas etárias mais elevadas (<https://blockly.games/>).

Muitas outras poderiam ser apresentadas, tais como Run Marco; Crunchzilla; Puzzlets; BitsBox; Construct 2; GameSalad; The Foos; CodeCombat; PENCILCode; Tynker; Kodable; Cargo-Bot; Lightbot; Robot Turtles, todas elas destinadas ao ensino da programação, e que, tendo em conta as suas especificações, podem ser utilizadas para explorar diversos conteúdos na área da programação. No entanto, a tendência geral, para os primeiros contactos com a programação, principalmente nas faixas etárias mais baixas, tem sido a programação por blocos e a utilização de interfaces apelativos.

Tendo em conta a especificidade desta investigação, as opções apresentadas são para níveis iniciais e não foram apresentados compiladores e linguagens de programação que são ensinadas num “nível” mais avançado, quer seja no secundário, quer seja no ensino superior. Destaca-se o facto de as plataformas/linguagens apresentadas, embora algumas tenham uma interface infantil, serem indicadas para quem está a iniciar, independente do seu nível de ensino e da sua faixa etária.

As semelhanças entre as opções/linguagens atuais e a origem do ensino da programação são evidentes. De lembrar que o ensino da programação surgiu em 1967/1968 com o desenvolvimento da linguagem de programação Logo, totalmente voltada para a educação e associada aos princípios de Papert (1996). De construcionista foi denominada a abordagem pela qual o aluno constrói o seu próprio conhecimento, através da interação aluno-objeto, mediado por uma linguagem de programação e por intermédio de um computador.

No entanto, e apesar da importância da aprendizagem da programação e dos seus benefícios, o papel do professor, da sociedade e do próprio sistema educativo é fundamental para se criar um bom equilíbrio, e, tal como refere Gerald (2014),

é importante ressaltar que a inclusão do ensino de programação nas escolas deve respeitar o desenvolvimento cognitivo e social das crianças. A escola, além de local de aprendizagem, é um espaço onde relações pessoais são construídas através da interação entre os colegas de turma e também com os professores. As brincadeiras e jogos próprios da infância, apesar de fortemente influenciados pelo uso das novas tecnologias, não devem ser esquecidos. As crianças não devem ser privadas do convívio social com outras crianças e também com adultos, pois esse contato é essencial para a formação do indivíduo.

Respeitando-se os estágios de desenvolvimento intelectual das crianças, é possível incluir o ensino de programação de computadores nas escolas sem causar prejuízo nenhum ao aluno e trazer vários benefícios ao processo de ensino e aprendizagem de outras disciplinas. (p.116)

Parece óbvio que, nos dias de hoje, as crianças já nascem rodeadas de tecnologia e o mundo digital tem um forte impacto nas suas vidas; porém, ao contrário do que, por vezes, aparentam, as crianças não conhecem o seu funcionamento e o ensino da programação é fundamental para que elas não se tornem apenas utilizadores passivos das ferramentas disponíveis.

METODOLOGIA

Folha em branco

Estudo 1 – Teste de Pensamento Computacional

Folha em branco

1.1. Teste de pensamento computacional desenvolvido por Marcos González

Desde cedo que uma preocupação era conseguir desenvolver/encontrar um instrumento que pudesse ir ao encontro das expectativas e que permitisse medir o nível de desenvolvimento do Pensamento Computacional dos alunos. No entanto, a criação de um novo instrumento é sem dúvida uma das tarefas mais complexas, pois é necessário conjugar diversas áreas do conhecimento. Assim, foi feito um levantamento dos instrumentos já existentes que pudessem responder aos objetivos desta tese. A vantagem de se utilizar um instrumento que já tenha respondido aos mesmos objetivos é a possibilidade de se realizarem comparações dos resultados em diferentes situações/culturas.

O teste de pensamento computacional tem como principal objetivo medir o nível de desenvolvimento do Pensamento Computacional de um determinado aluno/sujeito. Este teste foi desenvolvido por Marcos González, em 2016, no âmbito da sua Tese de Doutorado, com o objetivo de colmatar a ausência de instrumentos que possibilitassem este tipo de análise (González, 2016).

O teste é composto por vinte e oito perguntas, distribuídas por dezasseis páginas, com duas perguntas em cada uma. Todas as perguntas têm quatro opções de resposta (A, B, C ou D), das quais apenas uma é a correta.

A partir do momento em que o aluno comece o teste, tem quarenta e cinco minutos para o concluir. De realçar que o aluno não necessita de responder a todas as perguntas e que, antes de se iniciar o teste, são explorados três exemplos, para que o aluno se familiarize com o tipo de perguntas e com as personagens do mesmo.

Baseando-se nas normas estabelecidas pela Computer Science Teachers Association, o teste explora conceitos fundamentais da programação, nomeadamente: sequência, através das direções; condições, através de operações condicionais simples e compostas; e repetições, através de ciclos para repetir um determinado número de vezes, até uma determinada condição, ou enquanto uma condição for válida.

Cada pergunta aborda sempre um ou mais conceitos e os mesmos vão aparecendo de uma forma gradua. Aumenta a complexidade ao longo das questões e abre o caminho para explorar com mais rigor a capacidade que o aluno tem de formular e resolver problemas, diretamente ou de forma faseada.

TPC (Versión 2.0) - ÍTEM 9

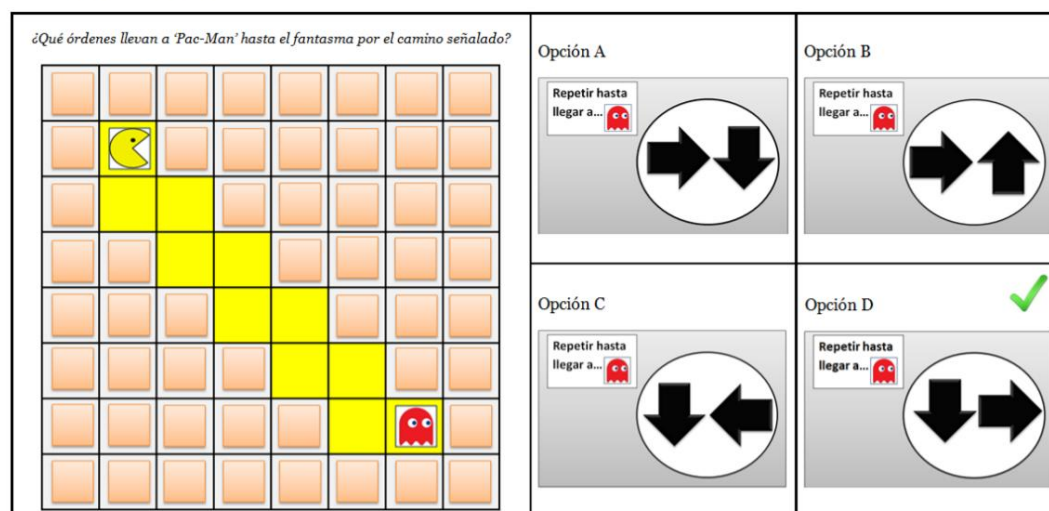


Figura 2. Exemplo de pergunta do Teste de Pensamento Computacional

Em termos gerais, pode dizer-se que o teste apresenta uma estrutura repartida, ou seja, é composto por 7 grupos de questões, nomeadamente:

- 4 perguntas de sequências simples;
- 4 perguntas por ciclo de repetição por número de vezes;
- 4 perguntas por ciclo e repetição até uma determinada condição;
- 4 perguntas por condições simples, condição – ação;
- 4 perguntas por condição composta, condição – ação – senão;
- 4 perguntas por ciclo de repetição enquanto uma condição se verificar;
- 4 perguntas com funções simples, de modo a repartir o problema e agrupar tarefas.

Este teste, segundo o seu autor, adapta-se a diversas faixas etárias. Realça-se que já foi utilizado por diversos investigadores, como por exemplo, nas investigações de Boucinha (2017), com uma amostra de 50 alunos do Brasil, com idades compreendidas entre os 11 e os 16 anos; e na investigação de Brackmann (2017), com alunos entre os 10 e os 12 anos, 35 do Brasil e 42 de Espanha. Os resultados dos estudos psicométricos têm confirmado a validade do instrumento na avaliação do constructo. Este trabalho é o primeiro estudo, em Portugal, com recurso ao instrumento não havendo estudos com a aplicação do mesmo à população portuguesa.

1.2. Procedimentos da adaptação à versão portuguesa

A tradução e adaptação de um instrumento é muito mais que a simples tradução das instruções/questões, uma vez que é necessário adequar o instrumento, tendo em conta o público alvo e a sua cultura.

Sendo um processo sequencial, é composto por vários passos em que o seu progresso depende, obrigatoriamente, da conclusão do passo anterior, tornando-se uma tarefa bastante complexa e demorada.

Neste seguimento, esta tradução seguiu os seguintes passos: análise e aprovação do autor para a realização da tradução/adaptação; elaboração da versão inicial do instrumento e análise por parte de professores que aplicam a iniciativa Introdução à “Programação no 1.º ciclo” (IP1); reflexão falada e construção da versão final.

Previam-se algumas dificuldades na tradução, nomeadamente o facto do teste utilizar uma forte componente visual, com recurso a imagens/esquemas e a nomenclatura que iria ser utilizada nos blocos de programação, uma vez que não havia uma tradução linear.

Análise e aprovação do autor

O primeiro passo para a adaptação do Teste de Pensamento Computacional foi perceber, junto do seu autor, Marcos Román González, se o mesmo seria adequado aos alunos dos 3.º e 4.º anos, uma vez que, o nível de escolaridade frequentado pela criança, variável que está diretamente relacionada com a idade, poderia ser um fator impeditivo para a utilização do mesmo.

Neste seguimento, foram estabelecidos vários contactos de forma a expor os objetivos desta tese, assim como as características dos alunos a quem se iria aplicar este instrumento.

Outro ponto, também explorado/alertado pelo autor do teste, foi a necessidade de se garantir a inexistência de discrepâncias entre as formas de aplicação, dando especial atenção à forma como os exemplos iniciais são explorados.

Assim, após fazer uma análise cuidada de todo o material enviado (objetivos, metodologia e caracterização do público-alvo) o autor deu o seu parecer positivo para a aplicação do Teste de Pensamento Computacional, aos alunos dos 3.º e 4.º anos e, também, autorização para que este instrumento fosse adaptado/traduzido para a Língua Portuguesa (cf. Anexo A).

Elaboração da versão inicial do instrumento

Após a aprovação do seu autor, foi necessário analisar, cuidadosamente, as diferenças culturais e ajustar os termos técnicos/linguagem à nossa realidade.

A pretensão, neste passo, foi manter, ao máximo, a estrutura do teste, bem como as suas “personagens”, uma vez que a sua transformação arriscaria alterar o interesse/motivação demonstrado pelos alunos, o que poderia modificar os resultados, principalmente no que concerne aos níveis de concretização. De salientar que, a motivação dos alunos para este tipo de perguntas/exercícios é, por norma, um fator decisivo e, tal como refere Fontes (2018), é necessário que os programas sejam próximos da realidade vivenciada pelos alunos e que tenham temas agradáveis.

Genericamente, foram consideradas as imagens originais do teste, disponibilizadas pelo autor, onde foram aplicadas técnicas de tratamento de imagem, às mais de 70 imagens que integram o teste.

Neste sentido, uma das principais decisões baseou-se na escolha do editor de imagens a utilizar. No mercado existem diversos programas, com diferentes tipos licenças e com especificidades/potencialidades que os caracterizam, tendo sido necessário verificar quais as vantagens e adequações para alcançar os objetivos deste passo. Este tipo de programa pretende facilitar a criação ou alteração de imagens digitais, em geral, estão agrupados em três tipologias: Raster – programas focados na pintura digital, edição e recorte de fotografias ou criação de desenhos/figuras, sendo os mais populares, o Adobe Photoshop, PaintShop Pro, Corel PhotoPaint, GIMP, Krita, Photoscape, entre outros; Vetoriais – programas de criação de imagens através de vetores (cálculos matemáticos),

como por exemplo, o Adobe Illustrator, CorelDraw, Inkscape, entre outros, e, por fim, os Tridimensionais – que fazem o tratamento de imagens tridimensionais através de diversos sólidos onde se destacam, entre outros, o 3ds Max, Blender, SkechUp, Autocad.

Tendo em conta o tipo de imagens originais e após a realização de diversos testes, em programas da tipologia Raster, constatou-se que os resultados obtidos, não só pelas suas funções, mas também pelas suas qualidades gráficas, eram mais ajustados através da utilização do Adobe Photoshop, tendo sido este o programa selecionado para a execução deste processo.

Este foi um processo bastante trabalhoso e moroso, porém, os resultados obtidos foram ao encontro das expectativas, havendo um grande nível de qualidade em todas as imagens/questões, fator fundamental para que os alunos conseguissem ser autónomos na realização do teste. Segundo Ambrósio (1999), a autonomia na criança pode ser compreendida como um processo e uma conquista que se inicia na infância precoce e que se desenrola ao longo da sua vida.

Na elaboração da tradução do teste, foi também adotado o esquema geral, seguido pelo autor, apenas 2 questões por página, assim como o modelo gráfico, tornando o teste o mais fiel possível ao teste original, visto que, segundo Moreira (2004), o gráfico do questionário, entre outros, deve ser considerado no decorrer deste processo.

Após a obtenção da primeira versão do teste, procedeu-se à comparação com outra tradução existente num país com língua oficial portuguesa, o Brasil, que apesar das disparidades no vocabulário e nas dissemelhanças em relação à sintaxe e à fonética ajudou a otimizar alguns itens.

Seguiu-se, junto de professores do grupo 550 – informática, que lecionam ou já lecionaram a “Iniciativa IP1” e de professores do 1.º ciclo, que já estiveram em coadjuvação com os anteriores, uma análise aprofundada de todos os exemplos e questões desta versão do teste, de forma a verificar se a linguagem e terminologia utilizadas eram as mais adequadas. Desta análise foram corrigidas/alteradas algumas questões no que concerne à sintaxe, havendo ajustes pontuais à linguagem utilizada.

Reflexão falada

Após a adaptação/tradução do instrumento, chegou-se à primeira versão do mesmo, onde foi aplicado o método da reflexão falada com alunos do 3.º ano de escolaridade, que nos permitiu proceder a uma análise sobre os itens que foram traduzidos e, assim, atingir uma versão melhorada.

Foi selecionado este método porque, assim, consegue-se, com o contributo de sujeitos próximos dos futuros destinatários (alunos a frequentar o mesmo ano de escolaridade e Agrupamento de Escolas onde iria ser implementado o programa de intervenção), analisar e discutir a proposta, permitindo aperfeiçoar o instrumento em causa, tanto na sua globalidade, como nas suas várias dimensões, tais como o conteúdo, a forma, a clareza, a compreensão e adequação aos objetivos (Feire & Almeida, 2008).

Neste seguimento, procedeu-se à solicitação de autorização junto da Direção do Agrupamento, que deu consentimento para se coordenarem os trabalhos com a professora titular da turma. A escolha dos alunos para a aplicação deste método foi efetuada de forma aleatória e implicou apenas a obrigação de existirem diferentes géneros (3 alunos e 2 alunas). Após esta seleção foi requerida e obtida a autorização dos Encarregados de Educação, após uma reunião onde foi explicado o âmbito deste estudo e quais os objetivos desta reflexão falada, destacando o facto desta ser uma atividade extra que não iria interferir nos conteúdos abordados pelo professor titular nem ser considerada para avaliação.

Para se conseguir o maior rendimento possível desta reflexão falada e obter bons resultados junto dos alunos/sujeitos que iriam estar envolvidos, existiu a preocupação de não ser um estranho a aplicar este método e haver alguma empatia/naturalidade. Para isso, houve o cuidado prévio de interagir com o grupo/turma em algumas atividades do Plano Anual de Atividades, além de dinamizar uma atividade em contexto de sala de aula juntamente com a professora titular.

Para uma observação mais eficiente, os alunos realizam o teste individualmente e em sala reservada, além disso, o investigador colocou-se em frente aos inquiridos e, durante o tempo de preenchimento, todos os comportamentos e comentários foram anotados. Existiu também o cuidado de ir dando um “feedback” construtivo, ou seja, mesmo quando a execução não foi a mais adequada, motivou o aluno para que colocasse em

prática as suas competências, explicando-lhe o que devia modificar e dando-lhe, no final do teste, a oportunidade de melhorar as suas opções (Jardim, 2003).

De início, a preocupação foi verificar se os itens estavam construídos de forma legível, uma vez que o teste recorre, inúmeras vezes, a imagens, chegando-se à conclusão de que teriam de ser, obrigatoriamente, impressos a cores, isto porque todos os alunos, manifestaram que a impressão a cores, além de ser mais apelativa, ajudava na compreensão e na diferenciação das personagens. Para mais, realçaram o facto de facilitar a contagem das quadrículas no momento de projetar a movimentação da personagem.

Outro aspeto explorado foi a avaliação da construção das questões, quer a nível gramatical, quer a nível de linguagem utilizada, verificando se era suficiente para a sua compreensão, atendendo à idade da criança.

Houve também especial atenção em compreender se as instruções iniciais e os três exemplos (informação escrita no início do teste) eram suficientes para que os alunos percebessem o objetivo e dinâmica do teste. Existiu também o registo do tempo médio de resposta para se perceber se os 45 minutos estipulados pelo autor do teste eram, no nosso caso, ajustados.

O tempo de execução (soma dos tempos de resposta a cada pergunta) utilizado por estes alunos raramente atingiu os 45 minutos, revelando-se, assim, suficiente para o efeito.

Todas as dificuldades e sugestões dos alunos, bem como as suas reações foram registadas através de notas de campo, de forma descritiva e reflexiva. Na descritiva deu-se especial atenção aos diálogos, atitudes e ações dos alunos, enquanto que a reflexiva consistiu no registo das primeiras impressões, bem como das preocupações no momento.


Nova versão do Teste de Pensamento Computacional

Após partilha das notas de campo com os colegas do grupo de informática e com as professoras do 1.º ciclo, chegou-se à conclusão de que nas questões 4 e 25 havia muita dificuldade na compreensão e que esta dificuldade estava relacionada com a necessidade de se utilizarem graus para projetar a rotação da personagem. Neste sentido a estas

questões foram acrescentadas imagens de um transferidor para ajudar a compreender a amplitude do ângulo utilizado.

Questão 4

Quais os passos que o Artista deve executar para desenhar o quadrado? Cada um dos lados do quadrado mede 100 pixéis.



<p>Opção A</p> <p>mover para a frente por 100 pixéis virar à direita por 90 graus mover para a frente por 100 pixéis virar à esquerda por 90 graus mover para a frente por 100 pixéis virar à direita por 90 graus mover para a frente por 100 pixéis</p>	<p>Opção B</p> <p>mover para a frente por 25 pixéis virar à direita por 90 graus mover para a frente por 25 pixéis virar à esquerda por 90 graus mover para a frente por 25 pixéis virar à direita por 90 graus mover para a frente por 25 pixéis</p>
<p>Opção C</p> <p>mover para a frente por 50 pixéis virar à direita por 90 graus mover para a frente por 50 pixéis virar à direita por 90 graus mover para a frente por 50 pixéis virar à direita por 90 graus mover para a frente por 50 pixéis</p>	<p>Opção D</p> <p>mover para a frente por 100 pixéis virar à direita por 90 graus mover para a frente por 100 pixéis virar à direita por 90 graus mover para a frente por 100 pixéis virar à direita por 90 graus mover para a frente por 100 pixéis</p>

Questão 4 A B C D

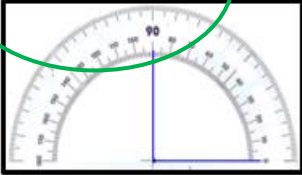


Figura 3. Exemplo de pergunta que sofreu alterações após a reflexão falada

Verificou-se também a existência de questões ambíguas em termos de formato, e algumas foram reformuladas: por exemplo, as frases com duas afirmações foram decompostas para facilitar a sua compreensão.

Assente nas informações alcançadas, atingiu-se a versão final do teste, disponível no Anexo G. Pode constatar-se que este teste de pensamento computacional se apresenta num formato simples e acessível e que a linguagem utilizada é bastante adequada, permitindo a administração a crianças que estejam a frequentar o 3.º ou 4.º ano de escolaridade.

1.3. Validação do instrumento

Ao longo deste ponto será apresentado o design do estudo de validação deste instrumento, nomeadamente: objetivos, participantes, instrumentos, procedimentos e resultados obtidos no que concerne à validação dos instrumentos.

Tal como menciona Coutinho (2008), é “fundamental que todo o investigador em educação se preocupe com a questão da fiabilidade e validade dos métodos a que recorre sejam eles de cariz quantitativo ou qualitativo” (p. 5).

Os dados foram recolhidos no final do ano letivo, mais concretamente, durante o mês de junho e, sempre que possível, após a realização das avaliações finais de ano.

Os resultados deste estudo caracterizam esta investigação, visto que a qualidade da mesma depende da fidelidade e da validade dos instrumentos utilizados para a recolha dos dados.

A validade dos instrumentos, através dos seus resultados, dá-nos a confiança necessária para saber se os mesmos estão a medir realmente aquilo que pretendem medir e, assim, irem ao encontro da investigação.

Vão ser utilizados métodos quantitativos e tal como mencionado por Gatti (2004), é necessário considerar, inicialmente, dois aspetos: o facto dos números, frequências e medidas terem propriedades que definem o tipo de cálculos e conseqüentemente os resultados obtidos; e o facto da qualidade das análises estarem diretamente relacionadas com as questões da investigação que direcionam as análises e respetivas interpretações. Caso se ignorem estas premissas poderão ser utilizados tratamentos estatísticos de forma indevida.

A consistência interna do teste de pensamento computacional vai ser avaliada com o objetivo de validar, através dos resultados obtidos, os instrumentos de medida utilizados.

1.3.1. Critérios de definição e estratificação dos participantes

Para esta investigação, estando direcionada para o pensamento computacional e para a “Iniciativa IP1”, mais especificamente para os alunos do 3.º e 4.º anos de escolaridade,

foi necessário estabelecer a dimensão da amostra, visto ser impossível chegar a todos os alunos.

Os autores Floyd e Widaman (1995), mencionam que, em regra geral, tem sido “rule of thumb regarding sample size for principal components and common factor analysis has been “the more participants, the better.””(p.289). No entanto, esta regra de quanto mais, melhor, é bastante vaga e outros autores apontam para a existência de números mínimos ou aconselhados, tendo em consideração o rácio de variáveis e participantes. Gorsuch (1983, citado por Floyd & Widaman, 1995), refere que devem existir, no mínimo, 5 participantes por cada variável, associado ao mínimo de 200 participantes. Everitt (1975), tal como Nunnally (1978, citado por Major, 2011), sugerem a presença, no mínimo, de 10 participantes para cada item avaliado.

Hair et al. (2009) abordam a questão do tamanho da amostra, aparentemente como regra mais abrangente, considerando que a amostra nunca deve ser inferior a 50 participantes e que, de preferência, a amostra deve ser maior ou igual do que 100, acrescentando a regra de, no mínimo 5 participantes por cada variável, no entanto mencionam que o mais aceitável é uma proporção de dez participantes por variável.

Tendo também em consideração que, quanto maior a amostra, mais precisos serão os resultados, mas indo ao encontro de todos os autores referenciados anteriormente, foi estipulado, para esta investigação, a utilização de um número de participantes entre 290 e 310, isto porque supera o número mínimo, além de ultrapassar também o rácio de 10 participantes por cada item, uma vez que, o instrumentos de maior dimensão utilizado é o teste de pensamento computacional composto por 28 itens.

Após a definição do tamanho da amostra foi necessário definir quais os critérios de seleção dos Agrupamentos de Escolas/participantes. Assim, definiu-se que deveriam:

- pertencer ao Quadro de Zona Pedagógica n.º 5 e estar sob orientação da Direção-Geral dos Estabelecimentos Escolares (DGEstE);
- estar a implementar a “Iniciativa IP1”, em pelo menos 1 turma do 3.º ou do 4.º ano de escolaridade;
- aglomerar todas as escolas públicas do seu concelho.

Neste seguimento, foram contactados todos os Agrupamentos de Escolas do ensino público, com a oferta do 1.º ciclo, a fim de se fazer um levantamento de quais reuniam as referidas condições para participarem nesta investigação.

Este contacto, foi feito através dos professores do grupo 550 – informática que estavam relacionados com a “Iniciativa IP1”, dos coordenadores do 1.º ciclo ou através das Direções dos Agrupamentos.

Não foi efetuado o levantamento nos concelhos de Castelo Branco, Covilhã, Fundão, Guarda e Seia, uma vez que são concelhos onde existe mais do que um Agrupamento de Escolas em funcionamento no ensino público e da sua projeção urbana, que, quer pela sua dimensão, quer pela qualidade de serem capital de distrito poderão ter um perfil diferente de aluno.

Tabela 2. Escolas com a IP1 no ano letivo 2016/2017

Distrito	Concelho	Agrupamento	Participar na IP1
Guarda	Aguiar da Beira	AE Padre José Augusto da Fonseca	Sim
Guarda	Almeida	AE de Almeida	Não
Guarda	Celorico da Beira	AE de Celorico da Beira	Não
Guarda	Figueira de Castelo Rodrigo	AE de Figueira de Castelo Rodrigo	Sim
Guarda	Fornos de Algodres	AE de Fornos de Algodres	Não
Guarda	Gouveia	AE de Gouveia	Sim
Guarda	Manteigas	AE de Manteigas	Sim
Guarda	Mêda	AE de Mêda	Não
Guarda	Pinhel	AE de Pinhel	Sim
Guarda	Sabugal	AE de Sabugal	Não
Guarda	Trancoso	AE de Trancoso	Sim
Castelo Branco	Belmonte	AE Pedro Álvares Cabral	Sim
Castelo Branco	Idanha-a-Nova	AE José Silvestre Ribeiro	Sim
Castelo Branco	Oleiros	AE Padre António de Andrade	Sim
Castelo Branco	Penamacor	AE Ribeiro Sanches	Sim
Castelo Branco	Proença-a-Nova	AE de Proença-a-Nova	Não
Castelo Branco	Sertã	AE de Sertã	Sim
Castelo Branco	Vila de Rei	AE de Vila de Rei	Sim
Castelo Branco	Vila Velha de Ródão	AE de Vila Velha de Ródão	Sim

A tabela anterior refere-se ao ano letivo 2016/2017, onde está apresentada uma síntese da informação recolhida, destacando-se o facto de, nesse ano letivo, a maioria das escolas

do Quadro de Zona pedagógica n.º 5, estarem a participar nesta iniciativa, mais concretamente, treze escolas das dezanove analisadas.

Neste estudo, dos Agrupamentos de Escolas que cumpriam os critérios de seleção, estiveram envolvidos três, de forma a satisfazer o número de participantes pretendido (entre 290 e 310), nomeadamente: Agrupamento de Escolas de Gouveia, Agrupamento de Escolas de Pinhel e Agrupamento de Escolas Pedro Álvares Cabral, que, em conjunto, perfazem um total de 17 escolas com a oferta do 1.º ciclo.

Agrupamento de Escolas de Gouveia

Com sede na Escola Secundária de Gouveia, o Agrupamento de Escolas de Gouveia, localizado no concelho de Gouveia, do distrito da Guarda, está integrado na Comunidade Intermunicipal das Beiras e Serra da Estrela. O concelho de Gouveia, com cerca de 12800 habitantes, é limitado a Norte pelo concelho de Fornos de Algodres, a Nordeste por Celorico da Beira, a Leste pela Guarda, a Sueste por Manteigas, a Sudoeste por Seia e a Noroeste por Mangualde (Agrupamento de Escolas de Gouveia, 2018).

Este agrupamento é composto por diversas infraestruturas, que compõem um total de 14 escolas, onde se leciona desde a educação pré-escolar até ao ensino secundário. Relativamente ao 1.º ciclo, este está presente em 8 escolas, nomeadamente:

- Escola Básica de Gouveia
- Escola Básica de Melo
- Escola Básica de Moimenta da Serra
- Escola Básica de Paços da Serra
- Escola Básica de São Paio
- Escola Básica de Vila Nova de Tazem
- Escola Básica de Folgosinho
- Escola Básica de Lagarinhos

Agrupamento de Escolas de Pinhel

Segundo a Câmara Municipal de Pinhel (2015), Pinhel encontra-se rodeada por colinas, montes e a notável serra da Marofa, sendo, portanto, uma cidade marcada por planaltos, fortalezas, monumentos e vastos campos. Esta cidade pertence ao distrito da Guarda, região Centro e sub-região da Beira Interior Norte, e é confrontada pelos concelhos de Mêda, Foz Côa, Figueira de Castelo Rodrigo, Almeida, Guarda, Celorico da Beira e Trancoso. Tem aproximadamente 8700 habitantes² e é conhecida por Cidade Falcão.

O Agrupamento de Escolas de Pinhel (2018), com sede na Escola Secundária de Pinhel, no ano letivo 2017/2018, teve 730 alunos, dos quais cerca de 200 pertenciam ao 1.º ciclo.

Os recursos materiais deste Agrupamento estão distribuídos por 9 escolas, com espaços interiores e exteriores. Os vários edifícios albergam os diferentes ciclos, do ensino pré-escolar ao ensino secundário, de carácter geral e profissional.

Relativamente às escolas que têm como oferta letiva o 1.º ciclo, enunciam-se 6, a saber:

- Escola Básica de Alverca da Beira
- Escola Básica das Freixedas
- Escola Básica de Pala
- Escola Básica de Pínzio
- Escola Básica de Souro Pires
- Escola Básica n.º 1 de Pinhel

Agrupamento de Escolas Pedro Álvares Cabral

O Agrupamento de Escolas Pedro Álvares Cabral (2017) foi constituído em 2003 e serve a população (cerca de 6400 habitantes) do concelho de Belmonte, do distrito de Castelo Branco. O Concelho de Belmonte, que ocupa uma área de 118,8 Km², é composto por quatro freguesias (Belmonte/Colmeal da Torre, Caria, Inguias e Maçainhas) e faz

² Dados retirados do site www.pordata.pt/

fronteira a Oeste com o concelho da Covilhã, a Sul com os concelhos de Fundão e Penamacor, a Norte com o concelho da Guarda e a Leste com o concelho do Sabugal.

No ano letivo 2017/2018, apesar da evolução do número de alunos demonstrar um decréscimo nos últimos anos, a escola era constituída por 590 alunos, dos quais 198 frequentavam o 1.º ciclo.

Em termos de instalações, este Agrupamento é composto por 6 escolas, uma delas a Escola Básica e Secundária Pedro Álvares Cabral, onde se situa a sede do Agrupamento.

Este agrupamento leciona os diversos níveis de ensino, desde o pré-escolar ao secundário, no entanto a oferta letiva do 1.º ciclo apenas está disponível em duas escolas, nomeadamente:

- Escola Básica Centro Educativo de Belmonte
- Escola Básica de São Marcos

1.3.2. Caracterização dos participantes

Foram considerados todos os alunos que tiveram a autorização do respetivo Encarregado de Educação para a participação no estudo, excetuando-se os alunos com problemas físicos, sensoriais ou intelectuais ou com dificuldades de aprendizagem que, ao abrigo do Decreto-Lei n.º 54/2018, se encontram a usufruir de medidas seletivas (medidas que visam colmatar as necessidades de suporte à aprendizagem não supridas pela aplicação de medidas universais) ou de medidas adicionais (medidas que visam colmatar dificuldades acentuadas e persistentes ao nível da comunicação, interação, cognição ou aprendizagem que exigem recursos especializados de apoio à aprendizagem e à inclusão).

Tabela 3. Distribuição dos participantes por Agrupamento

Agrupamento	N.º de Grupos	N.º de Alunos	Percentagem válida
Agrupamento de Escolas de Gouveia	15	150	49,5
Agrupamento de Escolas de Pinhel	5	94	31,0
Agrupamento de Escolas de Belmonte	6	59	19,5
Total	26	303	100,0

Tal como se pode verificar na tabela anterior, a amostra é constituída por 303 alunos, que se encontram divididos em 26 grupos, distribuídos pelos diferentes Agrupamentos. Salienta-se que o Agrupamento com maior número de participantes é o Agrupamento de Escolas de Gouveia.

A cada grupo corresponde uma turma e, portanto, um professor titular. Os grupos são constituídos, maioritariamente, por alunos de apenas um ano de escolaridade, no entanto, nos casos das escolas de menor dimensão, alguns grupos são mistos, constituídos por alunos do 3.º e do 4.º anos de escolaridade.

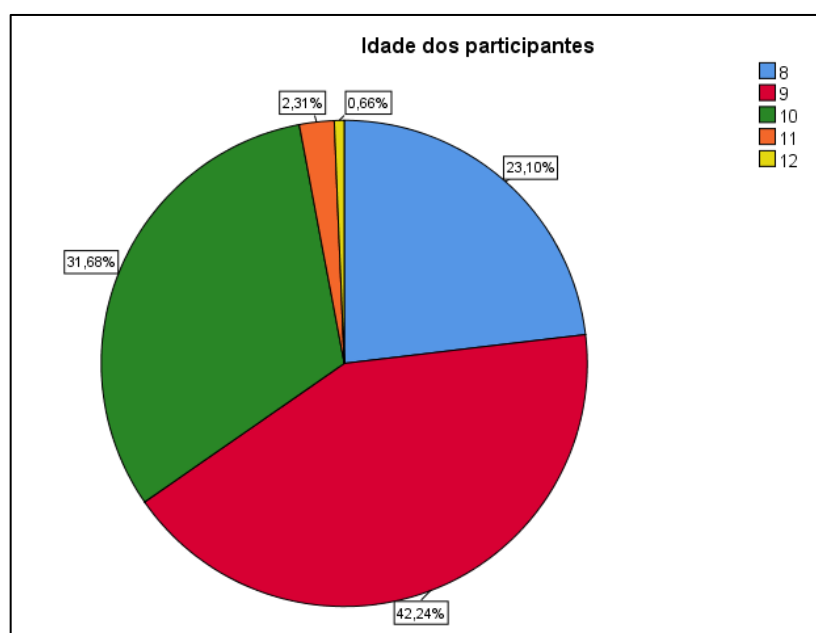


Gráfico 1 – Distribuição de participantes por idade

Os alunos, tal como mostra o gráfico anterior, de idades compreendidas entre os 8 e os 12 anos, têm, maioritariamente, 9 anos de idade, o que corresponde a, aproximadamente, 42% dos participantes.

Tabela 4. Índice do género da amostra

		Frequência	Percentagem	Percentagem válida	Percentagem cumulativa
Válido	Masculino	168	55,4	55,4	55,4
	Feminino	135	44,6	44,6	100,0
Total		303	100,0	100,0	

A distribuição dos participantes/alunos relativamente ao género é equilibrada, conforme se pode constatar na tabela anterior.

1.3.3. Instrumento de recolha de dados

A recolha de dados consiste, na recolha concreta de determinada informação junto dos participantes, para isso, e neste caso concreto, essa recolha foi concretizada através do protocolo de investigação (Anexo G) aplicado, em ambiente de turma e com a supervisão da professora titular.

O referido protocolo é constituído por vinte e uma páginas, repartido por três partes:

- **Questionário de dados sociodemográficos** – Composto por doze itens, dos quais, os três primeiros referentes a dados sociodemográficos, nomeadamente: idade, género e ano de escolaridade. Todos os outros pretendem enquadrar o participante no tema, não havendo respostas certas ou erradas, sendo apenas importante que as mesmas correspondam à sua experiência pessoal, aumentando assim, as perceções de sucesso por parte dos participantes. A existência destes itens de fácil resposta têm como objetivo produzir um efeito motivador em relação ao instrumento na sua globalidade.
- **Teste de Pensamento Computacional** – composto por vinte e oito perguntas, que são agrupadas em quatro, mediante os seus conteúdos e distribuídas por dezasseis páginas. Todas as perguntas têm apenas uma resposta correta, das quatro opções disponíveis (apresentado no estudo 1).
- **Teste de percepção de autoeficácia para a área de programação** - Composto por catorze perguntas, distribuídas por três páginas, em que todas as perguntas têm quatro opções de resposta, nas quais não existe propriamente uma resposta correta (apresentado no estudo 2).

Para dar resposta aos dados recolhidos através da Reflexão Falada, mais concretamente no que concerne à forma/apresentação dos instrumentos, os mesmos foram impressos a cores, uma vez que, este tipo de impressão, auxilia na resolução das questões no que respeita ao teste de pensamento computacional.

1.3.4. Procedimentos de Recolha

Esta investigação obedece às normas de ética próprias deste tipo de trabalho, considerando sempre que está sujeito a publicação e aos princípios que estão na base da ação humana. Como refere Fortin (2009), quaisquer que sejam os aspetos estudados, a investigação deve ser orientada no respeito pelos direitos da pessoa.

Assim, foram tidos em conta os princípios e orientações que constam na Carta de Ética da Sociedade Portuguesa de Ciências de Educação, aprovada em Assembleia Geral, em 12 de setembro de 2014, com especial cuidado no que se refere à relação dos participantes com a investigação.

Neste seguimento, desenvolveram-se os seguintes passos: pedido de autorização, à Direção-Geral de Estatísticas da Educação e Ciência, para a realização deste protocolo em meio escolar; seleção dos Agrupamentos a participar na investigação; reuniões com as diferentes Direções de Agrupamentos de Escolas e Coordenadores do 1.º Ciclo; pedido de autorização aos Encarregados de Educação; reuniões com os Coordenadores do 1.º ciclo para agilização da aplicação do protocolo; e, por fim, a aplicação do mesmo.

Pedido de autorização para a realização deste estudo/inquérito à Direção-Geral de Estatísticas da Educação e Ciência

Após a construção dos instrumentos, o primeiro procedimento foi o pedido de autorização para a realização deste estudo/inquérito à Direção-Geral de Estatísticas da Educação e Ciência, isto porque, ao abrigo do Despacho N.º 15847/2007, publicado no DR 2ª série n.º 140, de 23 de julho, não é possível a realização de qualquer estudo de investigação em meio escolar, utilizando por exemplo inquéritos ou entrevistas, sem antes se concretizar o pedido e se ter a apreciação positiva da DGE.

Este pedido foi feito através do site <http://mime.gepe.min-edu.pt>, disponível na página da Direção-Geral de Estatísticas da Educação e Ciência, concebido especificamente para este fim, titulado de Monitorização de Inquéritos em Meio Escolar (MIME).

Para a realização deste pedido foi necessário fazer o registo da entidade, especificando o âmbito desta investigação, e posteriormente, enviar, para aprovação, todos os instrumentos de inquirição, juntamente com outra informação/documentação, nomeadamente: a descrição do projeto de investigação, o seu enquadramento, notas

metodológicas, proposta de autorização para os encarregados de educação, entre outros esclarecimentos que foram solicitadas ao longo do processo.

Esta autorização (Anexo B) foi concedida para que a recolha fosse feita até 28 de julho de 2019, a cerca de 300 alunos dos 3.º e 4.º anos de escolaridade da zona Centro, condições estas que foram estritamente cumpridas.

Com a aprovação, foram também anexadas algumas observações/obrigações, destacando-se as seguintes:

- A realização dos inquéritos ficaria sujeita a autorização das Direções dos Agrupamentos de Escolas da região Centro de Portugal, a contactar para a realização do estudo.
- Seria necessária especial atenção ao modo, ao momento e às condições de aplicação dos instrumentos de recolha de dados em meio escolar, devendo fazer-se em estreita articulação com as Direções dos Agrupamentos.
- Para efeitos da proteção de dados a recolher junto dos inquiridos e em cumprimento da legislação em vigor, resultariam obrigações que o responsável teria de se propor a cumprir, tais como, dar conhecimento a todos os inquiridos e a quem interviesse na recolha e tratamento de dados pessoais. Assim, dada a natureza dos dados pessoais a recolher, teriam que, obrigatoriamente, prever-se medidas adequadas e específicas para a defesa dos direitos fundamentais e dos interesses do titular dos dados.
- As declarações de consentimento a utilizar (consentimento informado e esclarecido), tal como o inquérito, deveriam estar disponíveis na recolha do prévio consentimento dos representantes legais dos alunos (sua anuência/concordância com o que lhe seria efetivamente proposto responder).

A obtenção desta autorização foi de extrema importância, visto que, sem ela, legalmente, este estudo não poderia ser feito e nunca teríamos as autorizações, por parte das Direções dos Agrupamentos envolvidos, para se iniciar este processo.

Seleção dos Agrupamentos a participar na investigação

Após a autorização para a realização deste estudo à Direção-Geral de Estatísticas da Educação e Ciência, foi feita a seleção dos Agrupamentos de Escolas a convidar para participar na investigação, até perfazer cerca de 300 participantes.

O primeiro Agrupamento a ser selecionado foi o Agrupamento de Escolas de Pinhel, uma vez que era o Agrupamento onde o investigador estava a lecionar e tinha turmas do 3.º e 4.º anos, a participar na iniciativa.

Seguidamente, foi selecionado o Agrupamento de Escolas Pedro Álvares Cabral, dado que um dos professores do grupo de Informática tem fortes ligações com a ANPRI e dinamiza eventos/ações que envolvem a disseminação de práticas educativas da sua área. O professor envolvido aceitou prontamente e encaminhou o processo para a Direção do Agrupamento.

Por fim, e por ser conveniente à investigação um Agrupamento de Escolas onde existisse uma maior diversidade na implementação da iniciativa, nomeadamente, alunos que não participaram na iniciativa, alunos a participar pela primeira vez e alunos a participar pela segunda vez, foi selecionado o Agrupamento de Escolas de Gouveia.

Reuniões com as diferentes Direções de Agrupamentos de Escolas e Coordenadores do 1.º Ciclo

A fim de se obter a aprovação dos Agrupamentos selecionados, foi preparada uma intervenção para que, de uma forma transparente e objetiva, fosse apresentada esta investigação, o modo como iria ser implementada, os cuidados inerentes à mesma, bem como a apresentação dos instrumentos que iriam ser utilizados.

Nesta sequência, foram agendadas reuniões com as Direções, das quais resultou a intenção de participarem na investigação e o agendamento de reuniões com os Coordenadores de 1.º ciclo, a fim de expor essa mesma informação, visto a sua opinião ter um forte contributo para a aprovação em Conselho Pedagógico.

Com o decorrer do tempo, chegou a autorização, por parte dos três Agrupamentos, da participação na investigação.

Autorização dos Encarregados de Educação

Para a elaboração do pedido de autorização aos Encarregados de Educação teve-se em conta o forte papel do professor titular, uma vez que, tal como referenciado por Marques (2001), aquando da consideração do paradigma comportamentalista na análise do ensino e da aprendizagem, o professor é visto como o mais importante emissor de estímulos e que as influências ambientais, neste caso a sala de aula, são de extrema importância.

Assim, considerou-se que, se o professor titular estivesse bem informado/esclarecido sobre a investigação e interessado/motivado para que a sua turma participasse, haveria uma maior elucidação dos alunos e Encarregados de Educação, bem como um reforço positivo para que isso acontecesse.

Além disso, e tirando vantagem do facto da autorização ser em papel (Anexo C), foi feita uma abordagem geral aos alunos no momento da distribuição, além de existir, em todos os Agrupamentos, a possibilidade dos pais pedirem esclarecimentos/retirar dúvidas junto do investigador, enquanto decorria o prazo para a decisão/entrega da mesma.

Os Encarregados de Educação foram informados e deram autorização por escrito para a participação dos seus educandos no respetivo estudo.

Agilização da aplicação do inquérito

Para coordenar a aplicação do inquérito, foram realizadas várias reuniões com os Coordenadores do 1.º ciclo, de forma a esclarecer e justificar o protocolo. Para isso, foi necessário agendar, de forma cuidada, as intervenções nos diferentes Agrupamentos, uma vez que o questionário foi sempre aplicado pelo investigador.

Assim, e por sugestão dos Agrupamentos, a recolha foi feita em período específico, mais concretamente no final do 3.º período, depois de realizadas as fichas de avaliação finais, sendo para isso, em alguns casos, antecipadas/reagendadas as avaliações. Este calendário, em alguns Agrupamentos, foi até exposto e levado para aprovação nos Conselhos de professores do 1.º ciclo, para que houvesse a certeza que não iriam colidir com outras atividades ou que não iriam interferir com as aprendizagens planificadas.

Estes contactos serviram também para se fazer um balanço das autorizações entregues, solidificar a comunicação entre o aplicador/investigador e os outros intervenientes, e prestar alguns esclarecimentos.

Aplicação do protocolo

Como todos os intervenientes/alunos já estavam previamente alertados para a data e hora da aplicação do inquérito, o mesmo não trouxe qualquer surpresa, sendo, na maioria das vezes, encarado como mais uma atividade a desenvolver ao longo do dia.

A aplicação teve a duração de cerca de 60 minutos e foi sempre realizada pelo investigador, tomando as seguintes considerações, para garantir que a aplicação fosse, o mais idêntica possível, com todos os participantes:

- Uma abordagem inicial ao grupo/turma para se criar alguma empatia, estimulando os alunos a dar a sua opinião sobre a programação e a informática no geral, como por exemplo, expondo o que têm realizado nesta área ao longo do ano letivo.
- Definição de regras claras ao longo da resolução, retirando alguns níveis de ansiedade que pudessem existir. Foi assim reforçado o facto de não existir um contributo direto para a avaliação; que os resultados não iriam ser divulgados; a presença de itens, nas duas primeiras partes, onde não existia, propriamente, uma resposta correta; o facto de a resolução ser individual, sem a possibilidade de colocar dúvidas nem fazer comentários ao longo da prova; e da importância do seu contributo nesta investigação, devendo, por isso, ser o mais ajustado à realidade possível.
- Definição da hora de entrega, reforçando o facto de poderem entregar antes e de não ser obrigatório responder a todas as questões.
- Leitura da parte inicial do inquérito e exploração cuidada dos exemplos relativos ao teste de pensamento computacional.

A aplicação foi sempre acompanhada pelo professor titular de turma e sempre em ambiente escolar, sendo realizada, maioritariamente, na sala afeta à turma ou num espaço de frequência regular dos alunos, tais como a biblioteca ou a sala de informática.

Após a recolha, os testes eram assinalados com um número sequencial que serviu, posteriormente, para identificar o teste. Neste procedimento, houve a sugestão e

posterior colaboração dos professores titulares, para que os alunos que se encontravam a usufruir de medidas seletivas ao abrigo do Decreto-Lei n.º 54/2018 realizassem o teste, mas com o cuidado destes serem excluídos, sem que o grupo/turma se apercebesse, para não criar qualquer tipo de discriminação ou exclusão social. De realçar que existiu por parte de alguns professores da Educação Especial, a solicitação de mais inquéritos para realizarem junto de alunos de outros anos/faixas etárias, por considerarem que o mesmo era bastante interessante, tornando-se uma atividade bastante agradável para os alunos em geral.

1.3.5. Dificuldades encontradas

Como previsto, existiram dificuldades que tiveram de ser superadas, havendo ajustamentos constantes aos planos iniciais.

Destaca-se o facto do processo de autorização, em meio escolar, para a realização deste estudo/inquérito, à Direção-Geral de Estatísticas da Educação e Ciência, ter decorrido ao longo de oito meses. Sendo verdade que o processo passou por uma revalidação e por pedidos de esclarecimento/documentação. Não é menos verdade que se tornou um processo bastante moroso e com grandes ausências de feedback. O facto do processo ser inteiramente realizado através da plataforma, poderá ser considerado uma vantagem, principalmente, porque possibilita aos Diretores dos Agrupamento acederem aos instrumentos e confirmarem a sua autorização, mas, em contrapartida, a ausência de um contacto direto com os vogais que avaliam os pedidos, não facilita a comunicação.

Slomski e Martins (2008) consideram que “uma cultura de investigação fundamenta-se na ideia de uma ciência educativa em que cada sala de aula é um laboratório e cada professor um membro da comunidade científica”(p. 6), no entanto existiu inicialmente um sentimento de receio entre os professores do 1.º ciclo em participar na investigação, circunstância que foi ultrapassada com o apoio dos Coordenadores do 1.º ciclo e com o facto de ser o próprio investigador a aplicar o inquérito.

O processo de aceitação da colaboração por parte dos Agrupamentos acaba por ser, também, um processo bastante burocrático, pois inicia com o contacto com a Direção, que por sua vez recolhe a apreciação dos Coordenadores do 1.º ciclo, que se baseia na opinião/disponibilidade dos professores titulares de turma, o que se torna um processo demorado, pois esta partilha de informação, por norma, acontece em conselho de

professores, em reuniões de coordenação, terminando em aprovação de Conselho Pedagógico. Como é compreensível, estas reuniões apenas acontecem de forma periódica, muitas vezes mensalmente e, não sendo um ponto fundamental para o funcionamento das escolas, é apenas debatido quando, por motivo de calendário, há disponibilidade. Esta dificuldade, foi minimizada pela forma direta, clara e objetiva com que foram feitas as abordagens iniciais.

No decorrer do processo de recolha também houve constrangimentos em coordenar horários, principalmente, devido ao elevado número de grupos e ao facto das escolas estarem bastante distribuídas e distantes em termos geográficos. Esta situação agravou-se, aleando o facto dos alunos do 1.º ciclo terem as suas atividades repartidas, apenas, por 4 blocos, tendo um intervalo ao meio da manhã e um intervalo ao meio da tarde. A questão de a aplicação não poder colidir com outras atividades letivas ou não letivas, como por exemplo a hora da leitura, a prática de atividades desportivas ou a participação de alguns alunos nas atividades dos respetivos Agrupamentos, também foi considerada.

1.3.6. Resultados da validação do teste de pensamento computacional

Os dados quantitativos aqui expostos foram tratados através do software IBM SPSS – Statistic, mais concretamente o IBM SPSS – Statistic versão 25.0, para ambiente Windows, licença esta disponibilizada pela Universidade da Beira Interior, em prol desta investigação. Isto porque, segundo Pereira (2008) o software IBM SPSS – Statistic pode ser caracterizado como uma poderosa ferramenta informática que possibilita realizar cálculos estatísticos complexos e visualizar os seus resultados de uma forma bastante rápida.

Para calcular a consistência interna do teste de pensamento computacional teve-se em conta o alfa de Cronbach, no entanto, houve algumas premissas que devem ser consideradas quando se avalia o valor obtido, nomeadamente:

- As respostas a todas as questões são do tipo dicotómico, uma vez que as respostas ou estão corretas (valor 1) ou estão erradas (valor 0);
- A complexidade/dificuldade do teste vai aumentando gradualmente;
- O teste apresenta uma estrutura repartida em conjunto de 4 perguntas por tema;

Esta técnica avalia as correlações entre as perguntas após a sua utilização, permitindo determinar até que ponto cada pergunta mede o conceito da mesma forma que as outras (Fortin, 2009).

A premissa para existir uma boa consistência interna do instrumento, com a utilização desta técnica, é que o valor do coeficiente alfa de Cronbach seja superior a 0,80, sendo considerada muito boa se for superior a 0,90.

Tabela 5. Medidas de consistência interna do Teste do Pensamento Computacional

Pergunta	Média	Desvio Padrão	r	α
Questão 1	0,96	0,196	0,057	0,646
Questão 2	0,90	0,307	0,073	0,645
Questão 3	0,65	0,478	0,281	0,634
Questão 4	0,42	0,494	0,242	0,636
Questão 5	0,68	0,470	0,361	0,630
Questão 6	0,82	0,385	0,225	0,639
Questão 7	0,42	0,495	0,354	0,630
Questão 8	0,34	0,475	0,164	0,641
Questão 9	0,79	0,408	0,295	0,635
Questão 10	0,39	0,489	0,142	0,642
Questão 11	0,52	0,501	0,330	0,631
Questão 12	0,22	0,415	0,032	0,647
Questão 13	0,56	0,498	0,371	0,629
Questão 14	0,39	0,488	0,212	0,638
Questão 15	0,22	0,415	0,169	0,641
Questão 16	0,22	0,412	0,104	0,644
Questão 17	0,35	0,478	0,230	0,637
Questão 18	0,44	0,497	0,252	0,636
Questão 19	0,41	0,493	0,167	0,641
Questão 20	0,38	0,485	0,303	0,633
Questão 21	0,51	0,501	0,248	0,636
Questão 22	0,18	0,381	0,151	0,642
Questão 23	0,15	0,353	0,075	0,645
Questão 24	0,45	0,499	0,260	0,635
Questão 25	0,27	0,445	0,046	0,647
Questão 26	0,49	0,501	0,218	0,638
Questão 27	0,37	0,484	0,248	0,636
Questão 28	0,55	0,499	0,201	0,639
Alfa de Cronbach Total			0,646	

r - correlação item-total; α - alfa de Cronbach com exclusão do item

Neste seguimento, e analisando os dados expostos na tabela anterior, verifica-se que o teste apresenta uma consistência interna aceitável, mas abaixo dos valores ideais.

Outro fator que ajuda a fortalecer a validade deste instrumento é a questão de os resultados obtidos serem expressivos e consistentes, verificando-se um bom grau de eficácia, que prognostica o desempenho semelhante em grupos de alunos com características semelhantes.

Confrontando os resultados com a versão original do teste, pode afirmar-se que os resultados vão ao encontro do que se espera, visto que, segundo o autor do teste, o valor de alfa de Cronbach tende a ser mais baixo com a diminuição da faixa etária e que os valores obtidos para os alunos espanhóis com 9-10 anos foram 0,668 e para os alunos com 11-12 anos foram 0,766 (Román, 2016).

Folha em branco

**Estudo 2 – Teste de percepção de autoeficácia
para a área da programação**

Folha em branco

2.1. Conceção do Teste de percepção de autoeficácia para a área da programação

A autoeficácia segundo Schwarzer e Scholz (2000), é um motor de um alargado e estável conhecimento das competências pessoais, podendo ajudar a lidar com um leque alargado de situações que podem potenciar o stress. De destacar, que as percepções/crenças de autoeficácia influenciam o indivíduo a construir o seu próprio julgamento sobre as suas capacidades, afetando a sua motivação e os padrões de comportamento/pensamento.

Este estudo pretende, assim, analisar a percepção de autoeficácia para a área da programação e, ao mesmo tempo, vai ao encontro da opinião de Osborne e Freyberg (1991, citado por Sá, 1997), quando referem que:

se não soubermos o que as crianças pensam e porque pensam desse modo, teremos uma reduzida probabilidade de produzir impacto com o nosso ensino independentemente da habilidade com que o fizermos (...) As semelhanças e diferenças entre a ciência das crianças e a ciência dos cientistas são de uma importância crucial para o ensino e aprendizagem da ciência. (p. 43)

Esta linha de pensamento também é partilhada por Trindade e Cosme (2010), ao considerarem os alunos como portadores de inteligência e saberes que não podem ser ignorados nos fatores educativos .

Assim, os resultados obtidos poderão ajudar a justificar o tipo de envolvimento dos alunos/participantes nas atividades da área da programação, bem como a sua postura perante as adversidades/dificuldades nas tarefas propostas e em novos desafios.

A percepção de autoeficácia para a área da programação foi medida recorrendo a uma escala específica, com base na literatura (Gwénaelle et al., 2011). A escala avalia a percepção do participante sobre a sua competência na área da programação (e.g. “Sou capaz, no computador, de identificar e representar sequências de ações de atividades do quotidiano”) e a resposta encontra-se dividida numa escala de likert de 4 pontos (1 – nada verdade a 4 – totalmente verdade). De sublinhar que, quanto maior o valor selecionado na resposta, maior é também a percepção de competência e de eficácia pessoal.

O teste é composto por catorze perguntas, distribuídas por 3 páginas. Todas as perguntas têm quatro opções de resposta, não existe nelas propriamente uma resposta correta, devendo o aluno responder, mediante as suas percepções, sobre as suas capacidades de realizar uma determinada tarefa, que envolve um determinado conteúdo. Para isso, todas as questões são compostas por “sou capaz...” ou “consigo...”.

8. Num jogo de computador ou tablet, **sou capaz** de dizer qual o resultado de um conjunto de instruções dadas a uma determinada personagem (ex. as teclas/passos necessários para a personagem chegar a um determinado local).

Nada verdade

Pouco verdade

Bastante verdade

Totalmente verdade

Figura 4. Exemplo de pergunta do Teste de autoeficácia para a área da programação

Tendo em conta a especificação do público alvo desta investigação - alunos do 3.º e 4.º anos de escolaridade e, apesar dos leitores típicos a meio do 1.º ciclo do ensino básico já terem adquirido competências de descodificação, ainda existe um elevado número de alunos que, nesta fase, apresentam dificuldades na leitura e podem não conseguir compreender a natureza das tarefas apresentadas no teste (Araújo et al., 2014). Neste sentido, são apresentados, na maioria dos itens, exemplos concretos que auxiliam a questão e, na aplicação do teste, houve sempre a leitura integral do mesmo.

A formulação das perguntas teve por base conceitos relativos às tecnologias da informação e comunicação, às ciências da computação e ao pensamento computacional, bem como às linhas orientadoras para os professores da “Iniciativa IP1”, mais concretamente os seus objetivos, como, por exemplo, “resolver problemas, criar histórias animadas e construir jogos com recurso ao desenvolvimento de programas informáticos” (Direção Geral de Educação, 2015).

Cada pergunta aborda sempre a concretização de uma tarefa ou o reconhecimento de um ou mais conceitos e, em geral, a complexidade das tarefas implícitas nas questões vai sendo gradual em termos de dificuldade. Não se regista uma dependência obrigatória entre as questões, havendo algumas em que os alunos poderão ser capazes de estar totalmente de acordo sem que, obrigatoriamente, isso implique o uso do computador,

como, por exemplo, a capacidade/habilidade de identificar um problema e decompô-lo em subproblemas.

Em termos gerais, pode referir-se que o teste está dividido em três vertentes; a primeira, mais direcionada para o uso, em geral, do computador e identificação de elementos inerentes às TIC; a segunda, mais focada na utilização genérica dos computadores ou tablets, explorando-se as capacidades e visão geral do aluno; a última, centrada em tarefas relacionada com a programação.

2.2. Validação do instrumento

Ao longo deste ponto, pretende-se evidenciar as dinâmicas desenvolvidas durante o processo de validação deste instrumento, uma vez que foi criado no âmbito desta investigação, e por esse motivo, nunca tinha sido aplicado. Tal como mencionado por Raymundo (2009),

a validação é o processo de examinar a precisão de uma determinada predição ou inferência realizada a partir dos escores de um teste. Validar, mais do que a demonstração do valor de um instrumento de medida, é todo um processo de investigação. (p.87)

Este estudo tornou-se uma tarefa fundamental para o investigador planear e refletir sobre o seu próprio plano de trabalho, mais concretamente no processo metodológico, visto a validação dos instrumentos de recolha ser fundamental para, posteriormente, conseguir avaliar o programa de intervenção que foi construído ao longo desta investigação.

As questões de investigação vão ao encontro da validação do instrumento. De realçar que este processo assenta em três pilares: validação do conteúdo, ao se fazer a análise se o teste cobre todos e apenas os aspetos do seu objeto, que neste caso, foi elaborada através do estudo de diferentes intervenientes, que analisam cada questão em relação às áreas dos conteúdos e objetivos a medir; validação de critério, ao se verificar se o teste consegue identificar/distinguir os participantes; e a validação do constructo, verificando-se se o teste, realmente, mede aquilo a que se propõe, neste ponto e apesar

da nossa amostra ser extensa, apenas haverá confirmação com a aplicação do teste noutra investigação e a existir, um maior relacionamento com outras variáveis.

2.2.1. Participantes, instrumentos e procedimentos

Após a construção da primeira versão deste instrumento, o mesmo foi sujeito a uma pré-análise por parte de diversos intervenientes de diferentes áreas do conhecimento (professores do 1.º ciclo e do grupo 550 – Informática). Nesta análise solicitou-se que cada interveniente, individualmente, formulasse a sua opinião sobre os seguintes pontos: linguagem utilizada nos enunciados das questões; se as questões iam ao encontro do que se pretendia; se existia ambiguidade; qual a melhor disposição das questões; entre outras observações que considerassem oportunas.

A partir da análise concretizada pelos intervenientes, efetuou-se uma nova e ponderada extração de conhecimento, analisando cada questão, deliberando-se se ela ficaria no instrumento, e quais as alterações/melhorias que poderiam ser feitas. Como resultado desta análise, o instrumento foi reformulado, onde se retiraram três questões, das dezassete iniciais, principalmente pela sua dificuldade de compreensão para este tipo de participantes, além de terem uma forte especificidade, como por exemplo “Eu consigo criar sequências de instruções que envolvam repetição”. Além disso, foram também revistas algumas questões de forma a tornar a linguagem utilizada mais acessível.

Findo este processo, foi novamente elaborada nova análise por parte dos intervenientes anteriores, onde se concluiu, por unanimidade, que o novo teste apresentava bastantes melhorias.

Este novo instrumento, foi sujeito a uma reflexão falada com 3 alunos e 2 alunas, já referenciada no estudo anterior, de onde resultou a necessidade de reformular alguns exemplos que acompanhavam as questões, já que suscitavam algumas dúvidas por não fazerem parte do dia a dia dos alunos.

Em termos de participantes no estudo de validação, estes mantiveram-se os mesmos do estudo anterior, 303 alunos do 3.º e 4.º anos de escolaridade da zona centro, visto que este teste estava aglomerado ao instrumento de recolha geral, tal como se pode verificar no Anexo G.

Durante a aplicação do teste, o mesmo foi sempre lido em voz alta, com os objetivos de impor um ritmo correto à sua resposta, evitar troca de comentários entre os participantes, além de facilitar a compreensão das questões abordadas.

2.2.2. Resultados da validação do teste de percepção de autoeficácia para a área da programação

Baseando-se no facto deste teste ser composto por 14 questões, todas elas com respostas fechadas, que utilizam uma escala do tipo Likert, a técnica aplicada para analisar a consistência interna foi o cálculo do coeficiente alfa de Cronbach.

O resultado do teste, poderá atingir os 56 valores, num máximo de 4 por questão (1 – nada verdade; 2 – pouco verdade; 3 – bastante verdade; 4 - totalmente verdade).

A consistência interna está relacionada com o grau de concordância existente entre todas as perguntas que fazem parte do instrumento de medida que, neste caso, é o teste de autoeficácia para a área da programação.

Tabela 6. Medidas de consistência interna do Teste de Autoeficácia

	Pergunta	Média	Desvio Padrão	r	α
1.	Sou capaz de ligar e desligar corretamente um computador.	40,95	68,256	0,414	0,881
2.	Sou capaz de identificar os componentes de um computador.	41,72	65,115	0,482	0,878
3.	Consigo escrever corretamente no computador.	41,14	65,338	0,548	0,876
4.	Sou capaz de organizar a informação no computador.	41,64	61,581	0,626	0,872
5.	Sou capaz, no computador, de identificar e representar sequências de ações de atividades do quotidiano.	41,33	65,366	0,499	0,878
6.	Consigo identificar e utilizar diferentes tipos de dados.	41,21	64,139	0,579	0,874
7.	Sou capaz de distinguir operadores.	41,20	65,455	0,538	0,876
8.	Num jogo de computador ou tablet, sou capaz de dizer qual o resultado de um conjunto de instruções dadas e uma determinada personagem.	41,13	66,940	0,442	0,880
9.	Sou capaz de aplicar estratégias que aprendi num jogo num outro em que esteja a jogar pela primeira vez.	41,47	64,932	0,514	0,877

10. Sou capaz de programar uma personagem num jogo de computador ou tablet para desenhar uma forma geométrica.	41,56	59,989	0,685	0,868
11. Sou capaz de dividir um problema em partes quando estou a usar um computador ou tablet.	41,37	65,697	0,434	0,881
12. Consigo identificar e corrigir erros num programa/jogo no computador ou tablet.	41,73	59,253	0,653	0,870
13. Sou capaz de otimizar um programa ou jogo no computador ou tablet.	41,98	59,225	0,671	0,869
14. Consigo planificar e criar um projeto no computador ou tablet.	41,89	58,815	0,665	0,870
Alfa de Cronbach Total			0,882	

r - correlação item-total; α - alfa de Cronbach com exclusão do item

Nos resultados expostos na tabela anterior estão descritos os valores de alfa de Cronbach para as 14 perguntas, onde se pode verificar que todas elas obtiveram um valor de alfa de Cronbach superior a 0,80, estando maioritariamente muito próximo do 0,90.

Para analisar o valor obtido de alfa, devem, também, ter-se em conta alguns fatores que estão relacionados, diretamente, com o cálculo do mesmo, nomeadamente:

- O número de perguntas afeta o resultado. Neste caso o teste tem apenas 14 perguntas, o que poderia prejudicar o valor obtido.
- A redundância, isto é, houve o cuidado de não existirem perguntas com o mesmo tema elaborado de forma diferente, uma vez que perguntas iguais aumentam o valor de alfa sem que isso signifique um aumento na sua consistência.

Através dos resultados obtidos, onde se verificam valores elevados do alfa, pode afirmar-se que existe uma boa validade interna do instrumento.

Estudo 3 – Avaliação do impacto da “Iniciativa IP1” nos participantes em estudo

Folha em branco

3.1. Questões de investigação

A utilização das tecnologias por parte dos jovens já não é vista como uma inovação, mas sim parte do seu dia a dia, e esta mudança de hábito levou Prensky (2006), a considerar os alunos e jovens como nativos digitais, uma vez que nasceram numa era em que têm um contacto diário com inúmera tecnologia e que projetam a sua utilização no seu próprio futuro. Facilmente encontramos dados concretos acerca desta realidade, como, por exemplo, o facto de no 1.º trimestre de 2018, em Portugal, apenas 6% das crianças/jovens, entre os 10 e os 17 anos, não terem ou usarem um telemóvel, percentagem que diminuiu para 3%, se considerarmos apenas os jovens entre os 13 e os 17 anos (ANACON, 2018).

Outro fator, apesar de não estar relacionado diretamente com a escola, é que a vida familiar está agora repleta de dispositivos tecnológicos, sendo que a maioria dos adultos possui smartphones, computadores, tablets, entre outros (Anderso, 2015).

No entanto, é necessário perceber se, em termos de pensamento computacional, este se desenvolve naturalmente ou se necessita de ser estimulado.

Em geral, este estudo pretende avaliar, junto dos alunos do 3.º e 4.º anos, qual o impacto da “Iniciativa IP1” no pensamento computacional e nas perceções de autoeficácia na área da programação. Para isso, serão analisados diferentes contextos, nomeadamente, alunos que nunca participaram na iniciativa e alunos a participar pela primeira vez ou segunda vez.

Tendo como referência as linhas orientadoras da Iniciativa que envolve a programação no 1.º ciclo, documento que aborda conceitos relativos às TIC, às Ciências da Computação e ao Pensamento Computacional e onde são identificados os objetivos e os conteúdos a trabalhar com os alunos nas atividades do projeto, bem como sugestões metodológicas para as sessões com os alunos e formas de avaliação, pretende-se que este estudo avalie os impactos da iniciativa no pensamento computacional dos alunos e nos conhecimentos dos alunos em termos de princípios e conceitos fundamentais das Ciências da Computação.

Assim, neste processo de avaliação dos impactos da iniciativa e não esgotando totalmente o conjunto de perguntas a conceber, definiram-se para este estudo as seguintes questões orientadoras que serviram de guias para a recolha da informação necessária à sua abrangência:

Questão 1

Qual o impacto da iniciativa no pensamento computacional dos alunos envolvidos?

Questão 2

Os conceitos relacionados com o pensamento computacional estão a ser, todos, desenvolvidos com esta iniciativa da mesma forma?

Questão 3

Qual o impacto da iniciativa nas perceções de autoeficácia que os alunos têm sobre a área da programação?

3.2. Participantes, instrumentos e procedimentos

Os participantes, 303 alunos que se encontram divididos em 26 grupos, instrumentos e procedimentos são coincidentes aos dois estudos anteriores, no entanto para este caso é fundamental fazer-se uma análise mais aprofundada dos participantes de forma a compreender os diversos contextos/características dos mesmos.

Analisando apenas a participação na “Iniciativa IP1”, 127 alunos, estavam a participar pela segunda vez, a maioria portanto. Dos restantes, 111 participaram pela primeira vez, enquanto que 65 nunca participaram nesta iniciativa, nem noutra relacionada com o pensamento computacional ou com a informática em contexto escolar.

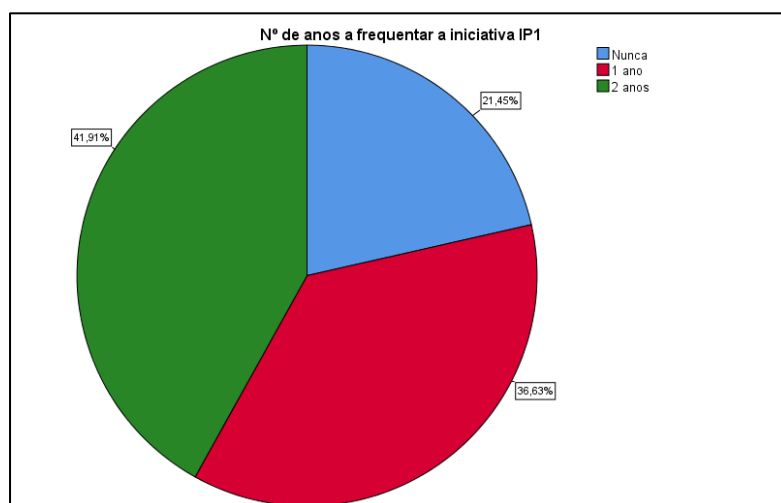


Gráfico 2 – Distribuição de participantes por anos de frequência na iniciativa

Especificando, estes 21,45% dos alunos ainda não tiveram oportunidade de integrar a “Iniciativa IP1”, devido à falta de oferta por parte do seu Agrupamento de Escolas ou por opção do seu Encarregado de Educação. Realça-se que 41,91% são alunos do 4.º ano que estão na iniciativa pelo segundo ano, facto que acontece, essencialmente, porque no momento da seleção dos Agrupamentos para a investigação, ano letivo 2016/2017, estes já estavam a implementar a iniciativa em pelo menos uma turma.

Tabela 7. Distribuição da amostra por ano letivo

Agrupamento	3.º Ano	4.º Ano	Total
Agrupamento de Escolas de Gouveia	67	83	150
Agrupamento de Escolas de Pinhel	53	41	94
Agrupamento de Escolas de Belmonte	27	32	59
Total	147	156	303

Em termos de ano de escolaridade, como mostra a tabela anterior, os participantes encontram-se no final do 3.º ou do 4.º ano de escolaridade, uma distribuição equilibrada, uma vez que 48,5% são do 3.º ano e 51,5% são do 4.º ano.

A distribuição dos alunos/participantes relativamente ao seu género é equilibrada, sendo, tal como já referido, 55% do sexo masculino e 45% do sexo feminino.

Tabela 8. Índice de idades da amostra

	Frequência	Percentagem	Percentagem válida	Percentagem cumulativa
	8	70	23,1	23,1
	9	128	42,2	65,3
Válido	10	96	31,7	97,0
	11	7	2,3	99,3
	12	2	,7	100,0
Total	303	100,0	100,0	

Os alunos, tal como mostra a tabela anterior, de idades compreendidas entre os 8 e os 12 anos, têm, maioritariamente, 9 anos de idade, o que corresponde a, aproximadamente, 42% dos participantes.

Este facto acontece devido à legislação portuguesa do Ministério da Educação e Ciência (2013), mais concretamente Despacho n.º 5048-B/2013: “A matrícula no 1.º ano do 1.º ciclo do ensino básico é obrigatória para as crianças que completem 6 anos de idade até 15 de setembro. As crianças que completem os 6 anos de idade entre 16 de setembro e 31 de dezembro podem ingressar no 1.º ciclo do ensino básico se tal for requerido pelo encarregado de educação, dependendo a sua aceitação definitiva da existência de vaga nas turmas já constituídas (...)”. Daqui resulta que, por norma, os alunos terminam o 3.º ano de escolaridade com 9 anos, mas também é possível que, alguns, no final do 4.º ano de escolaridade, tenham 9 anos.

Assumindo que todos os alunos têm sucesso ao longo do seu percurso escolar, é expectável que a idade não excedesse os 10 anos, uma vez que é a idade esperada no final do 4.º ano para um aluno não sujeito ao requerimento previsto no Despacho n.º 5048-B/2013.

Da análise dos dados pôde colocar-se a hipótese de 9 alunos terem uma ou duas retenções. No entanto, não é correto fazer tal afirmação, visto que este desfasamento poder também ser explicado por outros fatores, como a entrada mais tardia no 1.º ciclo ou a existência de um processo de transferência do estrangeiro, e por este motivo não terem sido retirados os participantes.

3.3. Resultados

Os resultados dividem-se em dois pontos: os resultados acerca do pensamento computacional dos alunos participantes e o estudo sobre a sua perceção de autoeficácia para a área da programação. Relativamente ao primeiro ponto, há também o estudo detalhado sobre qual a distribuição do impacto da iniciativa nos diferentes conteúdos/conceitos.

3.3.1. Resultados do teste de pensamento computacional

Os resultados obtidos, numa escala de 0 a 28 valores (1 valor por cada resposta correta), variaram entre 4 e 23 valores, com uma média geral de 12,5.

Tabela 9. Resultados do Teste do Pensamento Computacional por gênero

Gênero do Aluno	N	Mínimo	Máximo	Desvio Padrão	Média
Masculino	168	4	23	3,706	12,83
Feminino	135	7	20	2,965	12,10
Total	303	4	23	3,410	12,50

Tal como se pode verificar na tabela anterior, os resultados obtidos pelos diferentes gêneros são bastante idênticos, em termos de médias gerais encontram-se ambas entre 12 e 13 valores, sendo o valor mínimo, 4 valores, mais baixo no gênero masculino em contraposição com um valor máximo menor, 20 valores, no gênero feminino.

A Ciência e a Matemática são apresentadas frequentemente como atividades mais relacionadas com o gênero masculino. Segundo Melo (2017), isto acontece porque

geralmente chegamos a esta conclusão através de, entre outros, dois fatores consideráveis: a escassez de pensadoras e construtoras de saberes matemáticos que nos é apresentada, e a diferença expressiva do número de homens e mulheres cursando áreas voltadas para as ciências exatas e/ou atuando no mercado de trabalho dessa área, principalmente dentro das academias, no ensino de Matemática. (p. 190)

Este tipo de pensamento poderá levar os professores a criar expectativas erradas em relação ao desempenho escolar dos diferentes gêneros.

No entanto, os resultados obtidos vão ao encontro dos resultados obtidos noutras investigações, por exemplo nas desenvolvidas por Luiz e Dantas (2019), onde os resultados obtidos indicam que não existem diferenças significativas entre o pensamento computacional no gênero masculino e do gênero feminino.

De forma a fazer-se uma análise mais coerente e exaustiva sobre os impactos da iniciativa no pensamento computacional, analisaram-se os resultados mediante algumas variantes.

Tabela 10. Resultados do Teste do Pensamento Computacional em relação à frequência na iniciativa

Atividades Frequentadas	N	Mínimo	Máximo	Desvio Padrão	Média
Nunca	65	6	19	2,858	10,92
1 ano	111	8	22	3,132	12,29
2 anos	127	4	23	3,583	13,50
Total	303	4	23	3,410	12,50

Em geral, e fazendo a divisão dos alunos pelo número de anos que frequentaram a “Iniciativa IP1” (tabela anterior), denota-se que o aumento da frequência é acompanhado pelo aumento dos resultados obtidos no teste do pensamento computacional, havendo assim indicadores que a iniciativa poderá estar a ter um impacto positivo.

Tabela 11. Resultados do Teste do Pensamento Computacional em relação ao ano de escolaridade

Ano de escolaridade	Atividades Frequentadas	N	Mínimo	Máximo	Desvio Padrão	Média
3.º Ano	Nunca	62	6	19	2,911	10,95
	1 ano	85	8	22	3,098	12,18
4.º Ano	Nunca	3	9	12	1,528	10,33
	1 ano	26	8	21	3,273	12,65
	2 anos	127	4	23	3,583	13,50
	Total	303	4	23	3,410	12,50

Outra evidência, visível na tabela anterior, que indica o impacto positivo da iniciativa, é a comparação de médias entre os 2 anos letivos, 3.º e 4.º ano, onde se pode verificar que, apesar de anos diferentes, as médias dos alunos que frequentaram a iniciativa durante um ano são muito similares (12,18 e 12,65).

Tabela 12. Resultados do Teste do Pensamento Computacional em relação à idade

Idade do Aluno	N	Mínimo	Máximo	Desvio Padrão	Média
8	70	7	19	2,879	11,73
9	128	7	22	3,432	12,72
10	96	4	23	3,738	12,89
11	7	9	16	2,215	11,71
12	2	8	12	2,828	10,00
Total	303	4	23	3,410	12,50

A reforçar estes resultados, está o facto de os resultados obtidos não estarem relacionados com a idade dos alunos, visto a média dos que frequentaram a iniciativa em 2 anos ser superior a qualquer média obtida por idades.

Para avaliar os impactos da Iniciativa nos conhecimentos dos alunos em termos de conceitos fundamentais das Ciências da Computação, respondendo assim a uma das questões desta investigação/estudo, é necessário comparar os resultados obtidos nas diferentes questões e nos quatro diferentes grupos de questões do teste do pensamento computacional.

Tabela 13. Resultados do Teste do Pensamento Computacional em relação aos 4 grupos de questões

		<i>Alunos com a Iniciativa</i>			<i>Alunos sem a Iniciativa</i>		
		N	Desvio Padrão	Média	N	Desvio Padrão	Média
Sequências simples	Questão 1	238	0,210	0,95	65	0,269	0,92
	Questão 2	236	0,334	0,87	64	0,333	0,88
	Questão 3	230	0,473	0,67	64	0,479	0,66
	Questão 4	224	0,494	0,42	65	0,504	0,49
Ciclo de repetição por número de vezes	Questão 5	235	0,448	0,72	65	0,497	0,42
	Questão 6	234	0,362	0,85	65	0,443	0,74
	Questão 7	227	0,492	0,41	62	0,491	0,61
	Questão 8	226	0,467	0,32	62	0,441	0,26
Ciclo de repetição até uma determinada condição	Questão 9	237	0,351	0,86	61	0,499	0,57
	Questão 10	238	0,485	0,37	64	0,427	0,23
	Questão 11	233	0,495	0,58	63	0,455	0,29
	Questão 12	231	0,429	0,24	63	0,455	0,29
Condições simples, condição – ação	Questão 13	232	0,489	0,61	62	0,487	0,37
	Questão 14	236	0,489	0,39	62	0,487	0,37
	Questão 15	236	0,412	0,22	62	0,385	0,18
	Questão 16	230	0,404	0,20	63	0,439	0,25
Condição composta, condição – ação – senão	Questão 17	233	0,480	0,36	59	0,471	0,32
	Questão 18	235	0,501	0,48	60	0,490	0,38
	Questão 19	227	0,490	0,40	55	0,480	0,35
	Questão 20	232	0,484	0,37	58	0,493	0,40
Ciclo de repetição enquanto uma	Questão 21	235	0,500	0,53	60	0,497	0,42
	Questão 22	232	0,367	0,16	59	0,393	0,19
	Questão 23	233	0,366	0,16	58	0,329	0,12

condição se verificar	Questão 24	231	0,500	0,47	57	0,469	0,32
Funções simples de modo a repartir o problema e agrupar tarefas	Questão 25	220	0,449	0,28	51	0,440	0,25
	Questão 26	226	0,500	0,47	52	0,503	0,46
	Questão 27	226	0,487	0,38	53	0,505	0,49
	Questão 28	227	0,488	0,61	53	0,503	0,45

Na tabela anterior estão patentes as médias obtidas a cada questão, pelos alunos que nunca frequentaram a iniciativa e pelos alunos que já frequentaram a iniciativa pelo menos durante um ano.

No primeiro grupo, da questão 1 à 4, que se refere à implementação de sequências simples, e no sétimo grupo, da questão 25 à 28, que se refere à utilização de funções simples de modo a repartir o problema e agrupar tarefas, as médias obtidas são semelhantes, subentendendo-se que nestas áreas a iniciativa poderá não estar a ter impacto e que os alunos já desenvolvem naturalmente parte destes conceitos no seu dia a dia.

A utilização de ciclo de repetição até uma determinada condição, presente no terceiro grupo, da questão 9 à 12, é onde a iniciativa poderá estar a ter o seu maior impacto. Este impacto parece também ser significativo na utilização de condições simples, condição – ação presente no quarto grupo, da questão 13 à 16, e na utilização do ciclo de repetição, quer em ciclos de repetição por número de vezes, presente da questão 5 à 8, segundo grupo, quer no ciclo de repetição enquanto uma condição estiver presente, sexto grupo, da questão 21 à 24.

No quinto grupo, da questão 17 à 20, referente à utilização condição composta, condição – ação – senão, nota-se uma melhoria menos acentuada que as referidas anteriormente.

Em suma, a iniciativa está a ter um maior impacto em termos de estruturas de controlo, sejam estruturas de decisão ou de repetição, principalmente quando utilizadas sem recurso a decisões compostas.

Os resultados obtidos, na sua generalidade, são semelhantes ao padrão de resultados obtidos noutros estudos (Serafini, 2011; Geraldés, 2014; González, 2016; Brackmann, 2017), destacando-se que uma tem tendência para um efeito positivo em relação ao Pensamento Computacional, quando são aplicadas atividades/programas com cariz semelhante ao da “Iniciativa IP1”.

3.3.2. Resultados do teste de percepção de autoeficácia para a área da programação

Ao fazer-se a análise dos resultados obtidos no teste de percepção de autoeficácia, pretende-se analisar se a iniciativa está a ter um impacto positivo ou negativo na autoeficácia dos alunos nesta área. Este facto poderá ser importante para compreender a forma como os alunos abordam as diferentes tarefas na área da programação, visto que as crenças que as pessoas têm acerca das suas capacidades para serem bem sucedidas nas suas tarefas é um fator “vital” para o sucesso ou fracasso que terão nas suas atividades (Pajares, 2006).

Tabela 14. Autoeficácia perante a iniciativa

Anos na Iniciativa	N	Mínimo	Máximo	Desvio Padrão	Média
Nunca	65	16	47	7,424	34,63
1 ano	111	27	56	7,080	46,50
2 anos	127	28	56	6,030	48,13
Total	303	16	56	8,549	44,64

Os resultados obtidos, explanados na tabela anterior, podem significar que as percepções de autoeficácia são mais elevadas nos alunos que frequentaram a iniciativa.

Tabela 15. Autoeficácia perante a iniciativa por ano de escolaridade

Ano de Escolaridade	Anos na Iniciativa	N	Mínimo	Máximo	Desvio Padrão	Média
3.º Ano	Nunca	62	16	47	7,411	34,27
	1 ano	85	27	56	7,139	46,94
	Total	147	16	56	9,574	41,60
4.º Ano	Nunca	3	41	44	1,732	42,00
	1 ano	26	30	56	6,823	45,08
	2 anos	127	28	56	6,030	48,13
	Total	156	28	56	6,245	47,51

Na comparação entre os anos de escolaridade, há indicadores de que a percepção de autoeficácia aumenta com a frequência na iniciativa, até porque, as médias dos alunos que participaram ao longo de um ano letivo, no 3.º e no 4.º ano, são bastante idênticas, respetivamente 46,94 e 45,08 e inferior ao alcançado pelos alunos que se encontram na atividade pela segunda vez, 48,13.

Tabela 16. Autoeficácia perante a iniciativa por sexo

Género do Aluno	Atividades Frequentadas	N	Mínimo	Máximo	Desvio Padrão	Média
Masculino	Nunca	34	18	47	7,684	34,74
	1 ano	60	29	56	7,079	48,60
	2 anos	74	28	56	5,970	49,12
	Total	168	18	56	8,805	46,02
Feminino	Nunca	31	16	46	7,252	34,52
	1 ano	51	27	56	6,299	44,04
	2 anos	53	32	56	5,893	46,75
	Total	135	16	56	7,918	42,92
Total	Nunca	65	16	47	7,424	34,63
	1 ano	111	27	56	7,080	46,50
	2 anos	127	28	56	6,030	48,13
	Total	303	16	56	8,549	44,64

O aumento regista-se independentemente do género do aluno, mesmo que o masculino apresente resultados ligeiramente mais elevados.

Estes dados vão ao encontro do referenciado por Durak et al. (2019) ou por Türker Pala (2020), uma vez que os resultados obtidos mostram indicadores de que a percepção de autoeficácia da programação está relacionada positivamente com o pensamento computacional.

Estudo 4 – Programa de Intervenção

Folha em branco

4.1. Aplicação Scratch

O programa que aqui se apresenta foi elaborado no âmbito do projeto-piloto “Iniciação à Programação no 1.º Ciclo do Ensino Básico”, promovido pela DGE, como proposta de intervenção para a área que tem integrado a oferta educativa de várias escolas com recurso à aplicação Scratch.

A escolha deste recurso foi ao encontro do defendido por diversos autores, como por exemplo Brennan e Resnick (2012), que afirmam que, com a utilização do Scratch, é possível relacionar diversos conceitos do pensamento computacional que são comuns e altamente úteis em inúmeras linguagens. Estes autores, ao longo do seu estudo, deram também algumas sugestões, para facilitar a avaliação do pensamento computacional através da programação, que estiveram presentes na construção deste programa de intervenção, como, por exemplo, a introdução da aprendizagem de variáveis e contadores, para que os alunos possam explorar temas e objetivos diferentes. Considere-se o caso de acrescentar uma pontuação na elaboração de um jogo, a elaboração de projetos individuais para que possam, de forma autónoma e sustentada, desenvolver diversas capacidades de criação e análise crítica dos seus próprios projetos, ou a possibilidade de ajudar e complementar os seus exercícios como uma atividade metacognitiva, encorajando-os a pensar sobre o seu pensamento.

Outro fator decisivo na escolha do Scratch para a construção deste programa foram os relatos de diversos autores sobre os resultados obtidos, utilizando este software, tais como Maloney et al. (2008), que analisaram 536 projetos e destacam o funcionamento/interface, mencionando que “we think that the multimedia aspect of Scratch facilitated urban youth’s engagement in programming” (p. 371) ou Farias et al., (2015), que nas suas conclusões referem que “os resultados mostram que a viabilidade do uso do Scratch é recomendada para qualquer tipo de usuário, mesmo para os que não têm muita experiência em programação.” (p.954).

O Scratch oferece mais de 100 blocos de programação agrupados em oito categorias na versão 1.4 (movement, appearance, sound, pencil, data, events, control, detection, operators and variables) e dez categorias na versão 2.0 (movement, appearance, sound, pencil, data, control, sensors, operators, events and more blocks), que podem ser manipulados para compor programas simples, que permitem criar jogos básicos. Tal como outras linguagens de programação, apresenta as seguintes características: são fáceis de aprender, fornecem feedback visual sobre os programas desenvolvidos (sob a forma de objetos animados) e permitem aos estudantes criar elementos interativos, como animações e jogos.

O Scratch tem sido avaliado como acessível e útil, mesmo pelos professores mais céticos (Clark et al., 2013; Maya et al., 2015)

Na literatura, podemos encontrar diversos exercícios e formas de utilizar o Scratch em diferentes níveis de ensino (Aureliano & Tedesco, 2012; Sáez-López & Sevillano-García, 2017) e, face ao desafio lançado pelo projeto-piloto, cada professor e escola construiu a sua própria intervenção. De facto, muitas delas são constituídas por atividades avulsas, desancoradas de um marco teórico que as estructure, de uma lógica organizadora que lhes dê sentido e coerência. Há também muitos professores a trabalhar dedicadamente, mas sem “guias” que orientem o seu trabalho, sem avaliação da eficácia da sua intervenção.

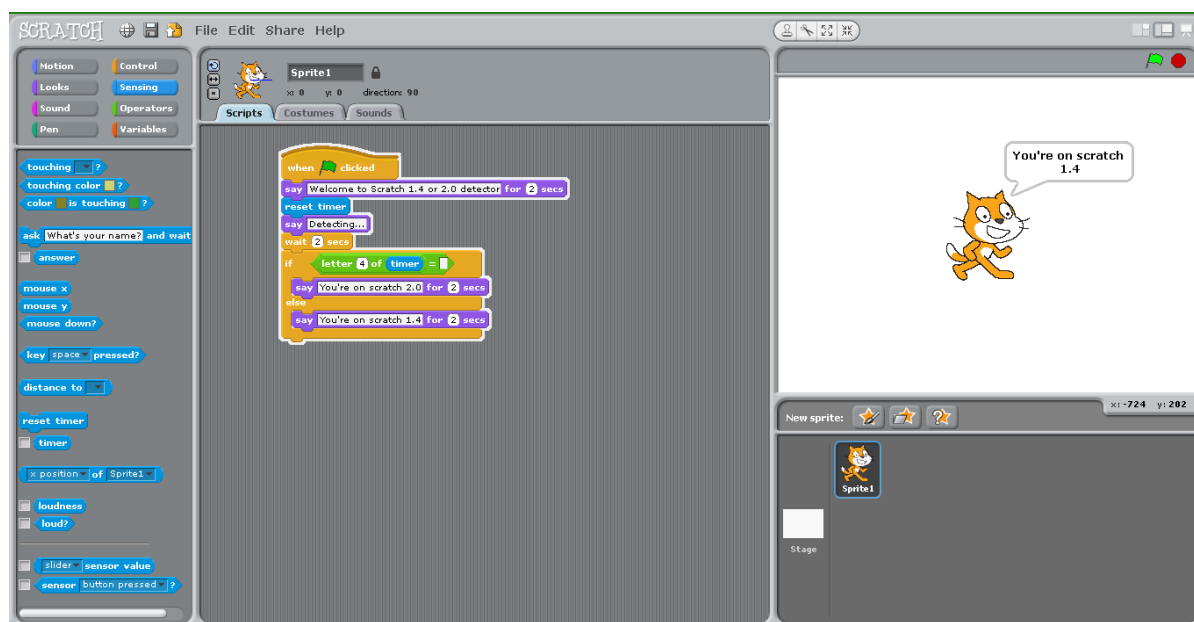


Figura 5. Ambiente de Trabalho do Scratch

4.2. Sessões do Programa

Este programa é, então, uma proposta de guião que orienta o trabalho dos professores da área da programação em parceria com os professores das outras áreas curriculares.

Logo de início, o professor poderá tentar uma abordagem mais interativa e menos expositiva, introduzindo os conceitos à medida das necessidades, de acordo com o desempenho dos alunos e do seu nível cognitivo, tendo por base uma taxonomia de objetivos educacionais.

Assim, e seguindo as linhas orientadoras do Ministério da Educação para o programa e as metas curriculares a serem atingidas pelos estudantes do terceiro ano, em conjunto com quatro professores do grupo 550 – Informática, foram desenvolvidas vinte sessões.

Este grupo de professores, que está ou esteve envolvido neste projeto-piloto, acompanhou a planificação e construção das sessões fazendo, progressivamente, a sua análise. Dessa análise conjunta, resultaram sugestões que promoveram que algumas sessões fossem totalmente remodeladas, atendendo ao seu carácter ou à sobreposição parcial de conteúdos. De realçar que as sessões apenas foram consideradas definitivas quando houve consenso.

As sessões foram planificadas para serem aplicadas em tempos de 45 minutos, com o objetivo geral de promover o pensamento computacional dos estudantes participantes e de colocar em prática alguns conteúdos abordados nas diferentes áreas curriculares.

O grupo de trabalho teve em consideração o contributo e as sugestões deixadas por vários autores que serviram de base de trabalho na escolha do método/exercícios a aplicar ao longo das sessões. Pode destacar-se o cuidado a ter com a utilização das tecnologias de informação e comunicação, que deverão ser sempre integradas de forma coerente com os conteúdos, tal como chama a atenção Bourguignon (1994), e as condições que vários autores (Sandholtz; Haymore Judith; And Others, 1997) consideram para que a integração das TIC melhore a qualidade da educação.

Tabela 17. Resumo das sessões

Sessão	Atividade	Conteúdos
1	Apresentação do Scratch	Conceitos gerais, exploração do ambiente de trabalho e programação da alteração de trajes.
2	Letras animadas	Inserção de diferentes letras criando novos trajes (diferentes cores). Programação da alteração de trajes.
3	Caminhar no deserto	Personalização do Palco, inserção de personagem com trajes, programação da alteração de trajes e movimento.
4	Aquário	
5	A bruxa	Introdução de pensamentos/falas e efeitos de som e cores.
6	Dança	
7	Formas geométricas	Criação de atores (formas geométricas) e programação da interação entre os diferentes atores.
8	Cores em inglês	
9	Animais	Personalização do palco e programação de diferentes sons e comportamentos para os diferentes atores.
10	Carros	Personalização do palco e atores com diferentes formas/estilos e programação de atores com diferentes comportamentos (andar, aparecer, deslizar, desaparecer).
11	O meu carro	
12	Jogo de basquetebol	Programação de diálogos devidamente coordenados no tempo.
13	Pista de dança	
14	Estações do ano	Programação do comportamento dos atores com alteração de cenários ao longo da animação.
15	Descobrir o mundo	
16	Apanhar borboletas	Conceito de números aleatórios conjugado com a programação do movimento e a interação entre atores (ação/reação).
17	Apanhar ratos	Introdução de variáveis associadas a um comportamento.
18	Labirinto	Programação de comportamentos através de condições que estimulem o cálculo mental (Figura 1 – exemplo em que o ator (formiga) terá de encontrar o caminho correto perfazendo a soma de 33).
19	Tiro ao alvo	Introdução de funções para otimização de código. Personalização/programação do cursor.
20	Questionário	Personalização de cenários e atores que através de funções podem explorar diferentes conteúdos.
21	Avaliação	Exploração/Avaliação dos conteúdos explorados.

Assim, como mostra a tabela anterior (resumo de sessões), os conteúdos vão sendo introduzidos ao longo das sessões de uma forma sequencial, e em alguns casos, está prevista mais que uma sessão para que seja feito o reforço/consolidação das aprendizagens.

As sessões estão planificadas para despertar o interesse dos alunos pela área da programação, possibilitando a aprendizagem de conceitos básicos de computação, tais como: sequências, execuções em paralelo, eventos, condições/estruturas de decisão,

ciclos, operadores, dados/variáveis e funções simples. Assim, todas as sessões recorrem à ferramenta Scratch (e.g. personagens, palcos, cenários, movimentos, som, sensores) para que, de uma forma lúdica, os alunos possam atingir os objetivos do projeto.

O Programa baseia-se também em Callear (2000), que sugere alguns pontos no método de ensino da programação que devem ser considerados: ensinar a um ritmo geralmente lento, de forma a dar tempo à assimilação de novos conceitos, sendo este o principal motivo pelo qual se considerou o número de sessões; a introdução de um tópico de cada vez, pois a introdução de vários tópicos ao mesmo tempo poderá confundir os alunos; voltar aos tópicos anteriores usando exercícios, algo que o programa faz de forma implícita, e, quando isso não acontece, o conteúdo é reforçado na sessão seguinte com a repetição dos mesmos; ter especial atenção com a ordem dos temas, ou seja, os temas mais acessíveis, que servem de base para outros, devem ser lecionados/explorados primeiro, sendo esta uma boa forma de dar confiança aos alunos.

O programa teve sempre como principal função criar um conjunto de aulas orientadas, em que o ensino e a aprendizagem estejam centrados no aluno, de acordo com os princípios do construtivismo, ou seja, pretende-se que o aluno seja um ser ativo que está envolvido na construção do seu saber e não se limite a assimilar passivamente os conteúdos transmitidos nas aulas. Assim, o aluno poderá reconstruir o conhecimento existente, dando-lhe um novo significado. Apesar desta aproximação, o programa apresenta alguma flexibilidade, uma vez que todos os exercícios são, a princípio, essencialmente orientados para ganhar progresso cognitivo e fazer com que a nota informação seja alojada nos conceitos previamente adquiridos na estrutura cognitiva do aluno, possibilitando assim a aprendizagem.

Os exercícios estão também organizados para proporcionar uma aprendizagem experiencial, com o intuito de transformar a prática de cada sessão numa aprendizagem que envolva o experimentar, refletir, pensar e agir.

Tabela 18. Resumo de Conteúdos abordados

Sessão	Atividade	Objetivo geral	Seqüências	Interação passiva do utilizador	Ciclos de Repetição	Instruções Condicionais	Comunicação e Sincronização	Lógica Booleana	Números Aleatórios	Variáveis	Interação direta do utilizador
1	Apresentação do Scratch	Apresentação do Scratch	X	X							
2	Letras animadas	Personalização de atores	X	X	X						
3	Caminhar no deserto	Inserção de palcos	X	X							
4	Aquário	Instruções condicionais	X	X		X					
5	A bruxa	Mensagens e sons	X	X	X	X	X				
6	Dança	Efeitos	X	X	X						
7	Formas geométricas	Sensores	X	X	X	X	X				
8	Cores em inglês	Sensores II	X	X	X	X	X				
9	Animais	Personalização de palcos	X	X	X	X	X				
10	Carros	Visibilidade (Sim/Não)	X	X				X			
11	O meu carro	Personalização de palcos e de atores	X	X							
12	Jogo de basquetebol	Comunicação sincronizada	X	X			X				
13	Pista de dança	Sobreposição de ações	X	X	X		X				
14	Estações do ano	Troca de palcos	X	X	X	X	X				

15	Descobrir o mundo	Troca de palcos e atores de forma sincronizada	X	X			X			
16	Apanhar borboletas	Números aleatórios	X	X	X	X	X	X	X	
17	Apanhar ratos	Variáveis	X	X	X	X	X	X	X	X
18	Labirinto	Interação do ator com o palco	X	X	X	X	X	X	X	X
19	Tiro ao alvo	Personalização do cursor e utilização de várias variáveis	X	X	X		X		X	X
20	Questionário	Interação direta com o utilizador	X	X	X	X	X	X	X	X
21	Avaliação	Avaliação	X	X	X	X	X	X	X	X

As sessões têm um grau de dificuldade crescente, como se pode confirmar na tabela anterior (resumo de conteúdos), que, em boa parte, é feita com a introdução de novos blocos de código, explorados no início de cada sessão. Esta opção baseou-se na perspectiva de Vygotsky, onde o papel do professor passa pela ajuda ao aluno, proporcionando-lhe apoio e recursos para que seja capaz de aplicar um nível de conhecimento mais elevado, através de aprendizagens orientadas que conduzam a um avanço no desenvolvimento. Das várias tentativas de encontrar a melhor forma para motivar os alunos no que respeita ao que se pretende ensinar, no caso dos conteúdos, em termos de programação pode-se destacar como fundamental a ideia apresentada em Pattis (1990), que aplica a filosofia para o ensino da programação em que os alunos desenvolvem projetos concretos e reais (com simplificações adequadas), através de um método de desenvolvimento gradual, tal como é feito ao longo deste programa.

As sessões têm, também, por regra, findarem com uma atividade complementar de consolidação/expansão de conhecimento para possibilitar que o aluno desenvolva a sua imaginação ao mesmo tempo que aplica estratégias de resolução de problemas. Este fator faz também com que nenhum aluno se sinta frustrado por não conseguir terminar a sessão e ajuda a criar um sentimento de sucesso, já que as sessões estão preparadas para que, mesmo os alunos que tenham mais dificuldades, cheguem à questão opcional.

Tabela 19. Interação com os elementos do Scratch

Sessão	Atividade	Controle	Palco importado	Palco criado	Atores importados	Atores criados	Movimento	Aparência	Som	Sensores	Operador	Variáveis
1	Apresentação do Scratch	X	X		Apenas 1			Traje				
2	Letras animadas	X	X		Vários	X		Traje				
3	Caminhar no deserto	X	X		Apenas 1		X	Traje				
4	Aquário	X	X		Vários		X	Traje				
5	A bruxa	X	X		Vários		X	Traje e pensamento	X			
6	Dança	X	X		Vários		X	Traje e efeito	X			
7	Formas geométricas	X	X		Apenas 1	X	X	Traje e fala			X	
8	Cores em inglês	X	X		Apenas 1	X	X	Traje e pensamento			X	
9	Animais	X		X	Vários		X	Traje	X			
10	Carros	X		X	Vários		X	Visibilidade				
11	O meu carro	X		X	Vários	X	X					
12	Jogo de basquetebol	Mensagens	X		Vários			Fala				
13	Pista de dança	X	X		Vários			Traje, fala e efeitos	X			
14	Estações do ano	X		X	Apenas 1		X	Trajes e troca de palco				
15	Descobrir o mundo	X	X		Vários			Visibilidade				
16	Apanhar borboletas	X	X		Vários		X	Visibilidade		X	X	
17	Apanhar ratos	X	X		Vários	X	X	Visibilidade		X	X	X
18	Labirinto	X		X	Vários		X	Visibilidade e fala		X	X	X
19	Tiro ao alvo	X		X	Apenas 1	X	X	Traje e fala		X	X	X (2)
20	Questionário	X		X	Vários			Visibilidade e fala	X	X	X	X
21	Avaliação	X	X	X	X	X	X	Trajes e troca de palco	X	X	X	X

A análise/disponibilização das tabelas anteriores são de extrema importância para os professores que vão aplicar este programa de intervenção, por dois motivos: o primeiro, porque os ajuda a saber quais os elementos que terão de dominar para poder preparar a sessão; o segundo, porque permite, através de uma visão simples e adequada, visualizar onde os elementos já foram explorados e, se necessário, ir às sessões anteriores consolidar conteúdos.

Em geral, o que se pretende é despertar a autonomia nos alunos estimulando a interpretação dos exercícios já realizados em sessões anteriores, mas para isso o professor que está a aplicar o programa deve conseguir, rapidamente, indicar ao aluno qual a sessão que deve rever.

Os exercícios do programa têm também em consideração Almeida e Valente(2011), que defendem a criação de ambientes de aprendizagem interativos, através da utilização das tecnologias digitais de informação e comunicação, para, desta forma, estimular novas formas de aprendizagem e, assim, preparar os alunos para o futuro.

Outro pilar deste programa é a disponibilização de exercícios que motivem os alunos, exercícios baseados nas suas experiências, uma vez que a motivação dos alunos é, por norma, um fator decisivo ou, tal como refere Fontes (2018), a motivação é um dos fatores básicos da aprendizagem.

4.3. Procedimentos

A escolha dos participantes deste estudo teve como ponto de partida o Agrupamento de Escolas onde o investigador estava a lecionar, uma vez que seria, provavelmente, a melhor forma de conseguir criar as condições ideais para a implementação do programa de intervenção. Esta premissa assenta em dois pilares: o primeiro, o facto de se conhecerem bem as salas e equipamentos disponíveis; segundo, existir já uma boa relação com a comunidade escolar em geral, o que se previa ser fundamental quer para a inscrição dos alunos na iniciativa quer para a autorização dos Encarregados de Educação na participação nesta investigação. Esta forte ligação com o Agrupamento foi também crucial na criação de uma relação de cooperação com a professora Titular de Turma.

O plano de amostragem é uma etapa do processo de investigação de grande relevância, já que se baseia no processo de seleção de uma parte da população para representar, fielmente, a sua totalidade (Fortin, 2009).

A escolha dos participantes aconteceu de forma aleatória, mas condicionada por algumas variáveis.

Após a aprovação/apoio por parte dos órgãos de gestão do Agrupamento, foram marcadas reuniões com o coordenador do 1.º ciclo, a fim de o inteirar do tema desta investigação, bem como discutir a forma e condições em que o programa poderia vir a ser implementado.

O objetivo inicial era criar condições para que o próprio investigador pudesse colocar em prática a iniciativa, com a aplicação do plano de intervenção desenvolvido, de forma a poder controlar, tanto quanto possível, as variáveis inerentes ao estudo. Para isso, além da atribuição da turma, necessitava ainda que os alunos se inscrevessem na iniciativa.

Neste seguimento, foi solicitada a atribuição, ao investigador, de uma turma do 3.º ano, que tivesse como oferta educativa a “Iniciativa IP1”. Não havendo qualquer outra orientação para indicar uma determinada turma.

No entanto, e havendo a vontade de se criarem as melhores condições possíveis para a implementação do programa de intervenção, o mais lógico era que a turma respeitasse algumas condições, tais como: não fosse uma turma mista, de forma a ter o maior número de participantes possível; pertencesse à escola mais próxima da escola sede, para que fosse possível deslocar, para lá, os alunos, visto que nenhuma escola com oferta do 1.º ciclo dispõe de sala de informática efetiva; e for fim, mas não menos importante, a professora titular da turma estivesse interessada em colaborar com a investigação.

Após esta deliberação, foi necessário fazer a escolha da turma que iria servir de grupo de controlo. Para isso, excluíram-se, logo à partida, todas as turmas da escola anteriormente escolhida e todas as turmas da escola onde tinha sido realizada a reflexão falada, aquando da construção do instrumento, para se garantir a inexistência de qualquer tipo de influência sobre os participantes/grupo de controlo.

Por fim, após se confrontarem as diversas variáveis, foi deliberado que o programa de intervenção iria ser implementado numa das turmas do 3.º ano da Escola Básica n.º 1 de Pinhel e que a turma/grupo de controlo, do mesmo ano, pertenceria à Escola Básica de Souro Pires.

Inicialmente, na turma do 3.º ano que foi atribuída ao investigador, já havia algumas inscrições na iniciativa, fruto das escolhas dos Encarregados de Educação, sobre a oferta disponível, aquando do ato da matrícula dos seus educandos. No entanto, como era uma amostra muito pequena, ainda se equacionou se não seria benéfica a implementação do programa de intervenção por outro professor, deste ou de outro Agrupamento, que estivesse a lecionar a mais alunos.

Em busca da melhor solução, apostou-se no contacto direto com os participantes, alunos e Encarregados de Educação, no início do ano letivo (na turma do 3.º ano da Escola Básica n.º 1 de Pinhel e que na turma, do mesmo ano, da Escola Básica de Souro Pires), para informar e esclarecer sobre todos os aspetos relativos à participação dos alunos no estudo. Este contacto, na sua maioria, foi feito em reunião geral Encarregados de Educação, onde foram apresentados os objetivos, a forma de funcionamento e a garantia do direito ao anonimato, privacidade e confidencialidade.

Revelou-se a preocupação de nunca mencionar nomes e de nunca serem publicadas imagens onde os participantes possam ser identificados. Houve ainda o cuidado de, posteriormente, contactar todos os que não puderam estar presentes. Após este contacto, os Encarregados de Educação deram o seu consentimento de forma escrita, através do preenchimento da autorização presente no Anexo C.

De referir que, depois da reunião inicial com os Encarregados de Educação, através do apoio da professora titular de turma e da partilha da opinião dos alunos já inscritos, sobre as atividades de enquadramento que foram lecionadas antes da aplicação do programa de intervenção, o número de inscrições foi crescendo, chegando a um número satisfatório.

Uma vez que se verifica a existência do papel de professor e de investigador em simultâneo, houve ainda a necessidade de abordar outro aspeto relacionado com a dimensão ética. Como professor, impõe-se a função de avaliar/classificar os alunos, neste caso, através de uma avaliação contínua, em 3 períodos, e de forma qualitativa. Assim, foi também dada a garantia de que os resultados obtidos através do teste de pensamento computacional, quer no pré-teste, quer no pós-teste não teriam qualquer influência na classificação obtida e que a classificação dos alunos iria respeitar os critérios definidos pelo grupo de informática, aprovados em conselho pedagógico, para todas as turmas do 3.º ano.

Foi ainda dado a conhecer que os alunos poderiam, a qualquer momento, desistir da sua participação, sem inconveniente, dispor de informação acerca do decorrer da investigação e da implementação do programa.

Houve, de mais, a preocupação de manter uma relação pautada por atitudes de sensibilidade relacional e de harmonia, com os princípios basilares para uma boa relação pedagógica, de responsabilidade, respeito mútuo, integridade e competência profissional com a professora titular de turma que acompanhou todas as sessões.

Estavam, assim, criadas todas as condições para a elaboração de um estudo experimental com o objetivo principal de testar/mediar a eficácia do programa de intervenção.

Os dados foram recolhidos em dois momentos temporais diferentes: o primeiro aconteceu no final do primeiro período e antes da implementação do programa de intervenção, sendo por isso chamado de pré-teste; o segundo momento aconteceu no final do ano letivo e após a conclusão da implementação do programa de intervenção no grupo de teste, formando assim o pós-teste.

O pré-teste teve como função avaliar os conhecimentos iniciais sobre o pensamento computacional dos alunos, bem como a sua perceção de autoeficácia na área da programação, enquanto o pós-teste pretendia recolher os mesmos dados, mas após a implementação, ou não, do programa de intervenção.

A recolha de dados, à semelhança do estudo anterior, foi feita através do protocolo que se encontra no Anexo G, em ambiente de turma e com a supervisão da professora titular.

O objetivo nunca foi construir um programa perfeito, mas sim um programa possível, avançando, simultaneamente, com a compreensão dos resultados que estavam a ser obtidos através de uma metodologia de investigação centrada na aprendizagem a partir de uma determinada estratégia/programa.

É incontestável, que ao desempenhar o papel de investigador e de professor em simultâneo, se esteja sempre a ter uma atitude reflexiva que, obviamente, vai modificar/melhorar a nossa prática.

Tal como menciona Cochran-Smith (2005), numa reflexão feita acerca da sua experiência nesta situação, não existem momentos distintos entre ser investigador e professor, caracterizando a relação entre as duas funções como recíproca, recursiva e simbiótica.

Ponte (2002) considera que a investigação na prática profissional do professor é um processo privilegiado de construção do conhecimento; no entanto, também aponta algumas dificuldades e critérios de qualidade que devem estar presentes.

Em síntese, não é um papel fácil de desempenhar. No entanto, é-o, sem dúvida, bastante enriquecedor.

4.4. Condições de implementação do programa

As experiências de aprendizagem em sala de aula necessitam de ser planeadas para desenvolver competências gerais, tais como: trabalho em grupo, resolução de problemas, autodomínio, pensamento crítico, poder de comunicação, entre outras. Essas experiências de aprendizagem devem habilitar o aluno como pessoa e, obviamente, como cidadão, agente de alterações responsável, criativo, inovador e capaz de contribuir para a sociedade, para a nação e para um mundo civilizado (Hadinugrahaningsiha et al., 2017).

As condições em sala de aula são fundamentais, de forma a criar o ambiente e as oportunidades ideais para todos os alunos. Assim, foi preparada uma sala de informática com 16 computadores, para assim garantir um computador por cada aluno, tomando como referência a dimensão do grupo experimental.

Esta opção não pretendeu nem implicou que os alunos não trabalhassem em grupo, até porque a tutoria de pares na sala de aula aumenta o envolvimento académico dos alunos sem o aumento do reforço pelo professor (Hayling et al., 2007). O único princípio defendido com esta opção foi que, sempre que necessário, o aluno tivesse ao seu dispor o material informático necessário.

Em termos de infraestruturas de rede, a sala funcionou através de rede por cabo, estando sempre disponível a ligação à internet através da ligação disponibilizada pelo Ministério da Educação, a qual pressupõe algumas restrições de utilização, para que diversos tipos de conteúdos/serviços não estejam disponíveis aos alunos.

De forma a salvaguardar a segurança e privacidade dos dados destes alunos, foi criada uma conta de utilizador apenas para esta turma, facilitando, assim, as tarefas de gerir os

ficheiros resultantes dos diversos projetos, além de excluir a hipótese de serem eliminados por outro aluno/utilizador.

A sala de aula estava também equipada com videoprojector e com algum material multimédia, nomeadamente colunas e auscultadores, que foram utilizados para auxiliar nas tarefas que previam a utilização de som.

Relativamente aos computadores, equipamentos que foram adquiridos pela escola no âmbito do Plano Tecnológico da Educação, entre 2007 e 2011, estavam em ótimas condições e nunca foram um obstáculo ao bom funcionamento das aulas nem, conseqüentemente, à implementação do programa. Estes computadores são constituídos pelas seguintes especificações:

- Monitor TFT de 17 polegadas, maioritariamente da HP, mas existindo de outras marcas;
- Processador Intel Core2 Duo E8400 e 3.00GHz;
- 2 GB de memória RAM;
- Disco rígido de 64 Gb repartido por 2 unidades;
- Diversos periféricos (rato, teclado)

Em termos de software, houve sempre o cuidado de realizar manutenções frequentes, de forma a garantir a operacionalidade do equipamento, estando instalado:

- Sistema operativo Windows 7;
- Microsoft Office 2007;
- Scratch 1.4;
- Gimp;
- Mozilla Firefox;
- Diversos softwares (KODU Game Lab, Antivírus (AVG)).

4.5. Dificuldades encontradas

A incerteza inicial, associada à burocracia inerente a este processo, fez com que fosse difícil e tardia a elaboração de um rumo concreto para este estudo. Foram necessários muitos ajustes, associados a uma capacidade de interajuda entre o professor titular e o investigador.

A premissa inicial de que era fundamental a motivação dos participantes para a iniciativa, estava também associada a uma certa resistência por parte dos Encarregados de Educação, aparentemente, relacionada com a desvalorização das atividades de oferta complementar. Este sentimento foi manifestado, em reunião própria, por considerarem que “os alunos já passavam demasiado tempo na escola” e que este tipo de atividades “poderia ser uma perda de tempo”. Estas afirmações foram fundamentadas com exemplos de más práticas anteriores, menos proveitosas. Além disso, houve também, Encarregados de Educação que estavam receosos pela possibilidade de “acrescentar conteúdos a um programa já, por si, tão extenso”. Estas opiniões foram moldadas através do apoio do professor titular, do diálogo e da partilha, realçando sempre o facto de se estarem a preparar as crianças para enfrentarem os desafios futuros, através da estimulação do seu pensamento crítico, capacidade de resolução de problemas, colaboração, agilidade e adaptabilidade, além de despertar a sua curiosidade e imaginação. Também foi dada a garantia de que os alunos poderiam, a qualquer momento, desistir da iniciativa, fator de instabilidade para o investigador, mas que, felizmente, não ocorreu com nenhum dos participantes.

A agilização/calendarização das sessões não se revelou tarefa fácil, era fundamental que todos os participantes estivessem presentes nas sessões, o que no dia a dia de uma escola, e tendo em conta o número de participantes, se veio a revelar uma tarefa bastante desafiadora. Esta dificuldade apenas foi superada, mais uma vez, graças à colaboração da professora titular, que permitiu o reagendamento de algumas sessões, aliado ao facto de, aquando da sua planificação, em termos temporais, já estar contemplada a possibilidade de existir outro tipo de intervenção, para quando, por exemplo, se registasse falta de alunos, por motivos de saúde, ou quando houvesse uma necessidade específica por parte da professora titular, como a participação em projetos de escola ou desenvolvimento de pesquisas específicas.

4.6. Avaliação do Programa de Intervenção

De seguida, será apresentado o design do estudo levado a cabo para avaliação do programa de intervenção, nomeadamente: objetivos, considerações metodológicas, bem como os resultados obtidos.

4.6.1. Objetivos

Este estudo pretende analisar em que medida é que o programa desenvolvido contribui para melhorar o pensamento computacional dos alunos que frequentem o programa - grupo experimental (GE), em comparação com um ensino em que não existe a aplicação de qualquer iniciativa relacionada com a programação no 1.º ciclo - grupo de controlo (GC).

Este estudo tem como objetivo fazer a comparação entre os resultados no domínio do pensamento computacional, no GC e GE, verificando se existe relação com os resultados obtidos noutras áreas do saber.

Foi seguido um modelo quasi-experimental, com avaliação de um pré-teste e de um pós-teste, em dois grupos.

O primeiro grupo, o GE, ou seja, um grupo de alunos que participou pela primeira vez na “Iniciativa IP1”, e o segundo grupo, o GC, um grupo de alunos do mesmo Agrupamento de Escolas, do mesmo ano, mas que não estavam inscritos na iniciativa, nem nunca participaram na mesma.

A escolha dos alunos foi feita de forma aleatória dentro dos alunos inscritos e não inscritos na iniciativa.

Ao primeiro grupo foi aplicado o programa de intervenção desenvolvido no âmbito desta tese e disponível no Anexo F. Para isso, e com o consentimento e aprovação/apoio da Direção do Agrupamento de Escolas de Pinhel, foram criadas as melhores condições possíveis para esta intervenção.

Este estudo teria também o objetivo de tentar perceber se é possível estabelecer uma ligação entre o pensamento computacional e outras áreas do conhecimento e entre o

pensamento computacional e a percepção de autoeficácia do aluno na área da programação.

4.6.2. Caracterização dos participantes

Os participantes deste estudo foram constituídos, por todos os alunos de uma determinada turma, uma vez que todos acabaram por se inscrever, e por alunos pertencentes a outra escola do Agrupamento que não estavam a ter qualquer contacto com a iniciativa. Além disso, considerou-se benéfico não excluir ou integrar qualquer outro sujeito na amostra, não sendo necessário definir critérios para este fim.

Desta forma, os participantes têm o perfil que caracteriza uma turma, embora se pretenda analisar os contributos dos resultados de forma a generalizar a outros alunos. Globalmente, grupo é formado 26 participantes/alunos, todos eles alunos do 3.º ano de escolaridade, que nunca tinham estado envolvidos diretamente ou indiretamente com a “Iniciativa IP1.”

Os participantes foram divididos em dois grupos: o grupo de teste, com 16 alunos, e o grupo de controlo, com 10 alunos. A constituição do primeiro grupo corresponde aos alunos da turma X, enquanto o segundo grupo corresponde aos alunos da turma Y (que não participaram na “Iniciativa IP1”).

Tabela 20. Índice de idades dos participantes

	Frequência	Percentagem	Percentagem válida	Percentagem acumulativa
	8	20	76,9	76,9
Válido	9	5	19,2	96,2
	10	1	3,8	100,0
Total	26	100,0	100,0	

A idade dos alunos, aquando da elaboração do pré-teste, variou entre os 8 e os 10 anos, conforme mostra a tabela seguinte, sendo de salientar que, aproximadamente, 80% se encontrava com 8 anos de idade, facto que era expectável devido ao enquadramento temporal (mês de dezembro e alunos do 3.º ano de escolaridade).

O aluno com 10 anos de idade, contabilizado na tabela, é um discente com nacionalidade estrangeira, que se encontrava enquadrado nesta turma depois de ter decorrido o processo de transferência/equivalência, não sendo descartado da amostra por não usufruir de medidas seletivas.

Tabela 21. Índice de idades por grupos (GT e GC)

	Grupo			Total
	Grupo de Teste	Grupo de Controlo		
	(GT)	(GC)		
	8	12	8	20
Idade do Aluno	9	3	2	5
	10	1	0	1
Total	16	10		26

Em termos de divisão por grupos, GT e GC, tal como se pode confirmar na tabela seguinte, mantém-se a tendência, tendo a maioria dos alunos 8 anos de idade.

Tabela 22. Índice do género da amostra do grupo experimental

	Frequência	Percentagem	Percentagem válida	Percentagem cumulativa
Masculino	14	53,8	53,8	53,8
Válido Feminino	12	46,2	46,2	100,0
Total	26	100,0	100,0	

No que concerne ao género, a distribuição dos alunos é equilibrada, havendo cerca de 54% do género masculino e, aproximadamente, 46% do género feminino, como se pode verificar na tabela anterior.

Tabela 23. Distribuição do género pelo GT e GC

		Grupo		Total
		Grupo de Teste	Grupo de Controlo	
Género do Aluno	Masculino	7	7	14
	Feminino	9	3	12
Total		16	10	26

Ao compararmos os diferentes grupos (ver tabela anterior), verifica-se que o número de alunos do género masculino se mantém igual nos dois grupos. No entanto, o mesmo não

acontece no género feminino - o número de elementos no grupo de controlo diminui bastante em relação ao grupo de teste. Este facto não pôde ser colmatado, uma vez que não havia mais alunos a reunir as condições necessárias para integrar o grupo de controlo.

4.6.3. Resultados do Pensamento Computacional

Ao analisarmos os dados alcançados neste estudo, mais concretamente ao compararmos os resultados obtidos pelos alunos antes e depois de ser aplicado o programa de intervenção, pretendemos verificar se há diferenças de médias no domínio do pensamento computacional.

Tabela 24. Médias pré-teste e pós-teste

Grupo	Total de alunos	Pré-teste	Pós-teste	Diferença
Grupo de Teste	16	9,1875	15,5625	6,375
Grupo de Controlo	10	9,0000	11,5000	2,500

Considerando apenas a diferença entre as médias obtidas, pode afirmar-se que a turma onde o programa de intervenção foi implementando apresenta um incremento considerável do pensamento computacional; no entanto, este facto carece de testes complementares para determinar a sua relevância estatística.

Neste seguimento, com o apoio do software IBM SPSS – Statistic, verificou-se se os dados obtidos para o pensamento computacional se enquadravam numa distribuição normal. Posteriormente, este resultado afetará as opções a serem utilizadas, a nível estatístico.

Tabela 25. Resultados do teste de Shapiro-Wilk

	Grupo	Shapiro-Wilk		
		Estatística	df	Sig.
Pré-teste	Grupo de Teste	0,917	16	0,151
	Grupo de Controlo	0,886	10	0,153
Pós-teste	Grupo de Teste	0,944	16	0,400
	Grupo de Controlo	0,968	10	0,876

De forma a testar a normalidade da amostra, foi utilizado o teste de Shapiro-Wilk (1965), devido às suas especificações e desempenho, destacando-se o facto de a sua versão original ser desenvolvida para amostras inferiores a 50 elementos, tal é o caso.

A utilização deste teste pressupõe a definição de duas hipóteses:

HO: A amostra resulta de uma população normal;

H1: A amostra não resulta de uma população normal.

Analisando a tabela anterior e considerando o teste de Shapiro-Wilk, para um nível de significância de 5%, fica validado o pressuposto - as variáveis em todas as amostras provêm de uma população normal, isto porque os resultados obtidos no nível de significância são maiores que 0,05, fazendo com que não seja possível rejeitar a primeira hipótese (HO).

Os gráficos seguintes, onde se pode visualizar a aleatoriedade dos pontos junto à reta, confirmam o pressuposto anterior.

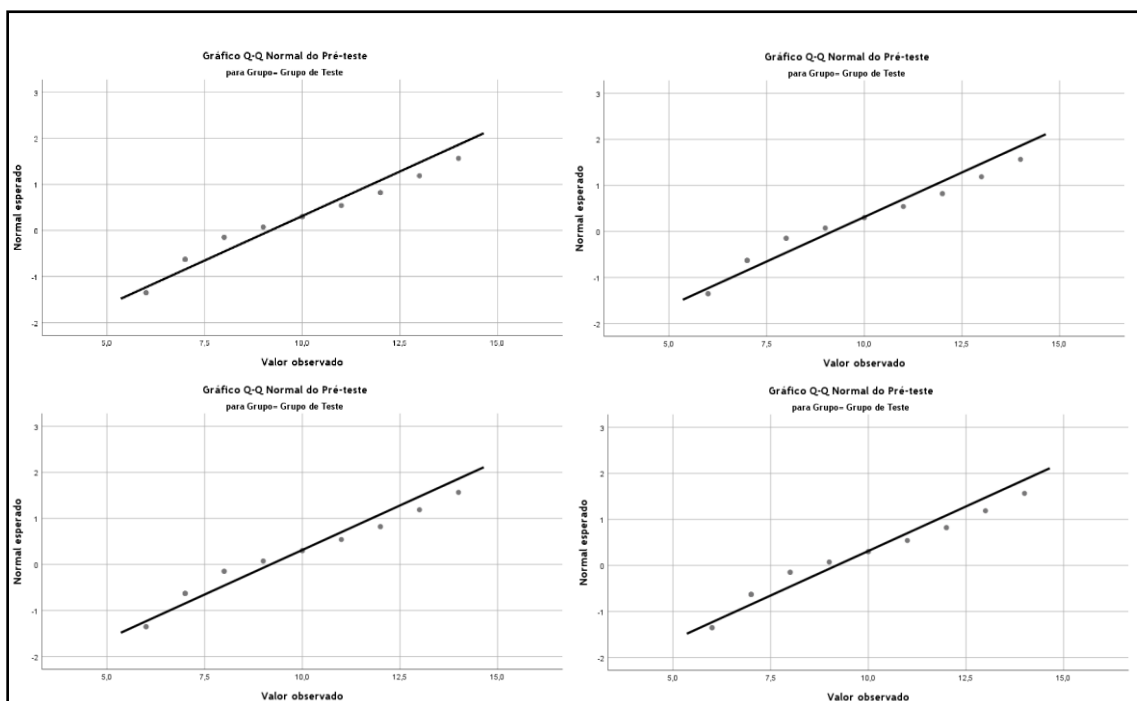


Gráfico 3 – Gráficos de normalidade do pré-teste e do pós-teste

A normalidade da amostra permite a aplicação de testes paramétricos. O teste utilizado foi o Teste-T, uma vez que permite fazer a análise da diferença entre duas médias e é o mais adequado quando os participantes são observados antes (pré-teste) e depois de uma determinada intervenção (pós-teste), ou seja, faz a comparação da mudança de uma média num par de valores, o que se adequa perfeitamente a esta investigação.

De forma a explorar os resultados foram definidas duas hipóteses: a primeira, a hipótese nula, em que se verifica a ausência de diferenças entre as médias dos grupos; a segunda onde se verifica que as médias são diferentes, ou seja as hipóteses são:

Ho: $\mu_{\text{Pré-teste do Pensamento Computacional}} = \mu_{\text{Pós-teste do Pensamento Computacional}}$;

H1: $\mu_{\text{Pré-teste do Pensamento Computacional}} \neq \mu_{\text{Pós-teste do Pensamento Computacional}}$;

Tabela 26. Tabela de dados com comparação de médias do grupo de teste

<i>Estatísticas de amostras emparelhadas</i>					
		Média	N	Desvio Padrão	Desvio padrão da média
Par 1	Pós-teste	15,5625	16	2,78014	,69503
	Pré-teste	9,1875	16	2,58763	,64691

<i>Correlações de amostras emparelhadas</i>				
		N	Correlação	Sig.
Par 1	Pré-teste & Pós-teste	16	,503	,047

<i>Teste de amostras emparelhadas</i>			
		Par 1	
		Pré-teste – Pós-teste	
Diferenças emparelhadas	Média	6,37500	
	Desvio Padrão	2,68017	
	Desvio padrão da média	,67004	
	95% Intervalo de Confiança da	Inferior	4,94684
	Diferença	Superior	7,80316
	t		9,514
	df		15
Sig. (2 extremidades)		,000	

Analisando os dados da tabela anterior, referentes ao grupo de teste, pode afirmar-se que as médias dos grupos são diferentes, uma vez que o resultado 0,000 da significância descarta a primeira hipótese.

Tabela 27. Tabela de dados com comparação de médias do grupo de controlo

<i>Estadísticas de amostras emparelhadas</i>					
		Média	N	Desvio Padrão	Desvio padrão da média
Par 1	Pós-teste	11,5000	10	1,43372	,45338
	Pré-teste	9,0000	10	1,76383	,55777

<i>Correlações de amostras emparelhadas</i>				
		N	Correlação	Sig.
Par 1	Pré-teste & Pós-teste	10	,615	,058

<i>Teste de amostras emparelhadas</i>			
			Par 1
			Pré-teste – Pós-teste
	Média		2,50000
	Desvio Padrão		1,43372
Diferenças emparelhadas	Desvio padrão da média		,45338
	95% Intervalo de Confiança da	Inferior	1,47438
	Diferença	Superior	3,52562
	t		5,514
	df		9
	Sig. (2 extremidades)		,000

As médias também são diferentes relativamente ao grupo de controlo, tal como se pode verificar na tabela anterior.

Tabela 28. Tabela de dados com as diferenças do pré-teste e do pós-teste

Grupo	N	Mínimo	Máximo	Desvio Padrão	Média
Grupo de Teste	16	1,00	10,00	2,68017	6,3750
Grupo de Controlo	10	1,00	5,00	1,43372	2,5000
Total	26	1,00	10,00	2,95739	4,8846

Com o intuito de validar a importância do programa de intervenção, foram calculadas as diferenças entre o pré-teste e o pós-teste, nos dois grupos (tabela anterior).

Ao analisarmos as diferenças entre o grupo de controlo e o grupo de teste, destaca-se o facto de o valor mínimo coincidir (1 valor), mas de o valor máximo ser o dobro, ou seja, no máximo, o valor do teste de pensamento computacional, no grupo de teste, é de 10 valores, enquanto no grupo de controlo é de, apenas, 5 valores. Isto verifica-se numa escala de 0 a 28 valores.

Outro ponto que se salienta nesta análise é o facto de a média das diferenças no pós-teste ser superior ao valor máximo de diferenças atingido no pré-teste. Este resultado, conforme pode ser observado nos gráficos seguintes, resulta do facto de a maioria dos alunos do pós-teste ter conseguido atingir uma diferença de 7 ou 8 valores.

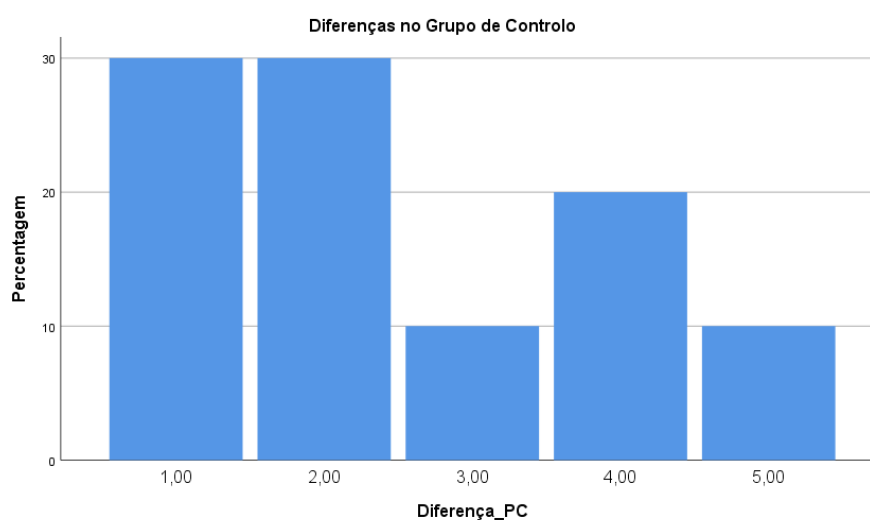


Gráfico 4 – Gráfico de diferenças no grupo de controlo

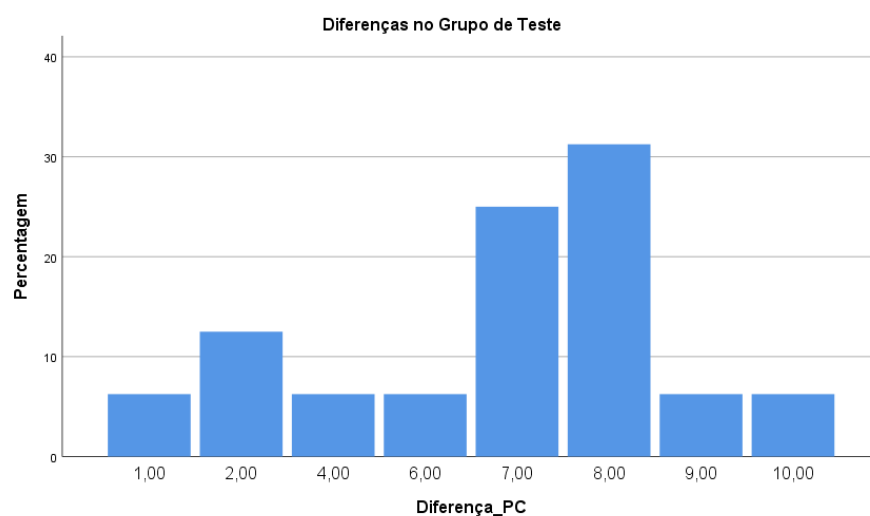


Gráfico 5 – Gráfico de diferenças no grupo de teste

Tabela 29. Tabela de dados com as diferenças do pré-teste e do pós-teste

<i>Estatísticas de grupo</i>					
	Identificação do Grupo	N	Média	Desvio Padrão	Desvio padrão da média
Diferença	Grupo de Teste	16	6,3750	2,68017	,67004
	Grupo de Controle	10	2,5000	1,43372	,45338

<i>Teste de amostras independentes</i>					
		Diferença_PC			
		Variâncias iguais assumidas		Variâncias iguais não assumidas	
Teste de Levene para igualdade de variâncias	Z		3,004		
	Sig.		,096		
	t		4,191		4,790
	df		24		23,626
teste-t para Igualdade de Médias	Sig. (2 extremidades)		,000		,000
	Diferença média		3,87500		3,87500
	Desvio padrão de diferença		,92456		,80902
	95% Intervalo de	Inferior	1,96679		2,20386
	Confiança da	Superior	5,78321		5,54614
	Diferença				

Recorrendo, mais uma vez, ao Teste-T, para analisar as discrepâncias, comprova-se (conforme tabela anterior) que as diferenças dos resultados obtidos nos dois grupos são significativas, uma vez que surgem alterações estatisticamente significativas para um intervalo de 95% de confiança (a significância associada ao Teste-T foi inferior a 0,05).

Com base nos resultados obtidos elaborou-se, o cálculo do tamanho do efeito. Este cálculo tem a vantagem de não estar sujeito ao tamanho da amostra e de permitir obter informação acerca do significado dos resultados, além de consistir numa métrica comum para comparar resultados de diferentes estudos.

Quando se está na presença de dois grupos, a análise das diferenças padronizadas entre as médias pertence à família d , que inclui o d de Cohen, o g de Hedges e o delta de Glass. Optou-se pelo g de Hedges, uma vez que os seus requisitos vão ao encontro das características da nossa amostra, visto que é recomendado quando os grupos têm pequena dimensão e são de diferentes tamanhos, pelo que os desvios padrão de cada grupo devem ser ponderados segundo a dimensão da amostra (Hedges, 1981).

Procedeu-se, assim, ao cálculo do tamanho do efeito, com o auxílio do site <https://www.socscistatistics.com/effectsize/default3.aspx>, onde se verificou que o Hedges' $g = (2500000 - 6375000)/2293559.762381 = 1.689513$.

Desta forma, aplicando a escala de Cohen (1998, p. 40), e uma vez que é superior a 0,80, o valor representa um tamanho grande em termos de efeito.

Tabela 30. Tabela de dados por género no pré-teste

	Género do Aluno	N	Média	Desvio Padrão	Desvio padrão da média
Pré-teste do	Masculino	14	9,2143	1,96815	,52601
Pensamento Computacional	Feminino	12	9,0000	2,66288	,76871

Teste de amostras independentes					
		PComputacional			
		Variâncias iguais assumidas		Variâncias iguais não assumidas	
Teste de Levene para	Z			1,599	
igualdade de variâncias	Sig.			,218	
	t			,236	,230
	df			24	20,002
Teste-T para Igualdade de Médias	Sig. (2 extremidades)			,816	,820
	Diferença média			,21429	,21429
	Desvio padrão de diferença			,90978	,93145
	95% Intervalo de	Inferior		-1,66341	-1,72867
	Confiança da Diferença	Superior		2,09198	2,15724

Averiguou-se ainda que não se verificou relação entre o género do participante/aluno com o valor do pensamento computacional (ver tabela anterior), uma vez que a significância associada ao Teste-T foi superior a 0,05, mais concretamente 0,816.

Uma vez que existem estudos que relacionam o pensamento computacional a outras áreas, tal como já referido anteriormente, avaliaram-se os impactos da iniciativa nos resultados dos alunos a nível de Matemática, Português e Estudo do Meio. Para isso, foram consideradas as classificações dos alunos, quantitativamente, na escala de 0 a 100 valores, obtidas nas diferentes disciplinas, no final do 1.º período e no final do ano letivo.

Tabela 31. Tabela de correlação no pré-teste

		Avaliação	Avaliação	Avaliação	PC
		Port	Mat	EM	
Avaliação	Correlação de Pearson	1	,884	,871	,226
Port	Sig. (2 extremidades)		,000	,000	,266
	N	26	26	26	26
Avaliação	Correlação de Pearson	,884	1	,839	,191
Mat	Sig. (2 extremidades)	,000		,000	,350
	N	26	26	26	26
Avaliação	Correlação de Pearson	,871	,839	1	,214
EM	Sig. (2 extremidades)	,000	,000		,295
	N	26	26	26	26
PC	Correlação de Pearson	,226	,191	,214	1
	Sig. (2 extremidades)	,266	,350	,295	
	N	26	26	26	26

Tal como se pode confirmar na tabela anterior, ao fazer-se a análise da correlação, pelo coeficiente de correlação de Pearson, dos resultados obtidos nas diferentes disciplinas no final do 1.º período, juntamente com o resultado obtido no teste de pensamento computacional (pré-teste), verifica-se que a variável referente ao pensamento computacional não está correlacionada com nenhuma das outras.

Tabela 32. Tabela de correlação no pós-teste

		Avaliação	Avaliação	Avaliação	PC
		Port	Mat	EM	
Avaliação	Correlação de Pearson	1	,938**	,911**	,279
Port	Sig. (2 extremidades)		,000	,000	,168
	N	26	26	26	26
Avaliação	Correlação de Pearson	,938**	1	,910**	,308
Mat	Sig. (2 extremidades)	,000		,000	,126
	N	26	26	26	26
Avaliação	Correlação de Pearson	,911**	,910**	1	,282
EM	Sig. (2 extremidades)	,000	,000		,162
	N	26	26	26	26
PC	Correlação de Pearson	,279	,308	,282	1
	Sig. (2 extremidades)	,168	,126	,162	
	N	26	26	26	26

O mesmo estudo foi feito para os resultados obtidos no final do ano letivo e no pós-Teste do Pensamento Computacional, verificando-se (ver tabela anterior) o mesmo tipo de resultados.

Fazendo uma apreciação dos indicadores aqui obtidos, verifica-se que os mesmos não vão ao encontro do sugerido na literatura (Serafini, 2011; Carmo, 2013; Garofalo, 2018; Shailaja & Sridaran, 2015) dado que não se verificou a existência de quaisquer relações entre os resultados obtidos nas diferentes disciplinas e os resultados obtidos no teste de Pensamento Computacional, nem antes, nem depois da implementação do programa de intervenção.

4.6.4. Resultados da percepção de autoeficácia para a área da programação

No que concerne aos resultados obtidos no teste de autoeficácia para a área de programação, que se encontra na escala de 0 a 56, os resultados obtidos entre os grupos nos dois momentos são bastante coincidentes, não havendo grandes variações nos resultados obtidos.

Tabela 33. Tabela de resultados de autoeficácia

Identificação do Grupo	Total de alunos	Pré-teste	Pós-teste	Diferença
Grupo de Teste	16	42,75	44,44	1,69
Grupo de Controlo	10	41,60	45,10	3,50

Considerando que, para o grupo de teste, os resultados foram obtidos em dois pontos distintos em termos temporais, para cada aluno, antes e depois da intervenção, e de forma a obter-se uma comparação mais precisa, optou-se por uma análise das amostras pareadas de Teste-T.

Tabela 34. Tabela da análise das amostras pareadas de autoeficácia

		Média	N	Desvio Padrão	Desvio padrão da média
Par 1	Pré Teste	42,75	16	6,981	1,745
	Pós Teste	44,44	16	8,625	2,156
<i>Correlações de amostras emparelhadas</i>					
			N	Correlação	Sig.
Par 1	Pré Teste - & Pós Teste		16	,654	,006
<i>Teste de amostras emparelhadas</i>					
				Par 1	
				Pré Teste - Pós Teste	
				Teste	
Diferenças emparelhadas	Média				-1,688
	Desvio Padrão				6,661
	Desvio padrão da média				1,665
	95% Intervalo de Confiança da Diferença		Inferior		-5,237
			Superior		1,862
t					-1,013
df					15
Sig. (2 extremidades)					,327

Analisando os resultados presentes na tabela anterior, verifica-se que não existe efeito da intervenção nas percepções de autoeficácia para a área da programação.

A análise destes resultados poderá ter duas perspetivas diferentes, mas complementares: se por um lado os resultados não vão na linha de pensamento de Guimarães e Abbad (2015), posto que não se verificou uma relação positiva entre pensamento computacional e as percepções da autoeficácia; por outro, temos as convicções de Costa (2002;) e Valle et al. (2003) que poderão justificar estes valores pela alta e constante motivação dos alunos que resultam, em ambos os momentos (pré e pós-teste), em valores muito similares em termos de percepção da autoeficácia para a área da programação.

CONCLUSÕES E LIMITAÇÕES DO ESTUDO

Folha em branco

1.1. Conclusão da Investigação

Com o culminar desta investigação e após a apresentação do enquadramento teórico, metodologia e resultados, impõe-se uma reflexão global relativa ao seu contributo para o desenvolvimento do pensamento computacional, nos alunos do 1.º ciclo, bem como das suas limitações e expectativas futuras.

No âmbito geral, existe um sentimento de satisfação sobre o trabalho desenvolvido, que, apesar da pretensão do investigador, estar longe de poder considerar-se um trabalho ultimado, aliás, seria utópico pensar-se de outra forma. A consciência de que muitas questões ficaram por abordar mantém vivo o desejo de prosseguir, um dia, com novas investigações sobre o pensamento computacional no ensino público.

Concluída agora esta investigação, e de acordo com as hipóteses colocadas e os objetivos propostos, apraz-se referir que se conseguiu alcançar a meta de construir um programa de intervenção que tem um impacto bastante positivo no pensamento computacional dos alunos do 3.º ano de escolaridade, uma vez que se verificou que os alunos que participaram na “Iniciativa IP1”, utilizando este recurso educativo, obtiveram melhores resultados que os restantes, não sujeitos ao mesmo programa. Em termos gerais, os alunos do 3.º ano que estão na iniciativa há um ano letivo apresentam uma média no teste do pensamento computacional de 12,65 valores, enquanto os do grupo de teste apresentam uma média de 15,65 valores, média esta que é superior à dos alunos do 4.º ano que estão na iniciativa há dois anos letivos.

De modo geral, pode-se afirmar que a iniciativa está a ter um impacto positivo no pensamento computacional dos alunos do 3.º e 4.º ano de escolaridade, verificando-se que este se desenvolve mediante a frequência da iniciativa, independentemente do ano que se frequenta, da idade ou do sexo do aluno. Assim, pode-se afirmar que os professores que dinamizam a iniciativa estão à altura do desafio que o sistema educativo lhes proporcionou e que, apesar da diversidade e heterogeneidade presentes em cada grupo/sala de aula, têm conseguido fazer a diferença.

No entanto, não é menos verdade que o impacto poderia ser muito maior caso as condições de lecionação e os programas/metodologias adotados fossem outros. Nesta linha, e em termos de implicações a nível do ensino e da aprendizagem, o trabalho desenvolvido ajudará os professores, num futuro próximo, a responder, com maior

eficácia, a este desafio, uma vez que têm ao seu dispor um programa de intervenção que desperta um conjunto de boas práticas, fomentando uma maior qualidade nos processos e resultados do ensino e da aprendizagem.

Outro benefício desta investigação, e não menos importante, é o facto de se apurarem, em concreto, os conceitos que os alunos estão a ter maior dificuldade em adquirir, nomeadamente, a utilização de funções simples de modo a repartir o problema e agrupar tarefas e a utilização de condição composta, condição – ação – senão, possibilitando assim aos professores que já tenham um programa de intervenção próprio, fazer os ajustamentos necessários para que os seus alunos evoluam mais rapidamente.

Além disso, os materiais disponibilizados permitirão aos professores do primeiro ciclo, interessados, explorar, de uma maneira fácil e intuitiva, este tema. Tenha-se sempre presente que um professor não deve apenas ser um transmissor de conhecimento, mas sim um formador de novas gerações que estimula a mudança e a promoção das aprendizagens significativas, contribuindo para um mundo mais justo e sustentável (Esteves et al., 2018). Esta abordagem, que poderá ser feita através de um ensino centrado nos alunos, segundo Silva (2004), permitir-lhes-á tornarem-se pesquisadores ativos da informação e construtores do seu próprio conhecimento através do recurso às novas tecnologias.

Além de os instrumentos de recolha de dados utilizados para avaliar o pensamento computacional e a autoeficácia dos alunos se terem revelado eficazes, existe também o benefício de serem disponibilizados na página www.programacao1ciclo.pt, para investigações futuras. Noutra vertente, propiciam aos professores fazer, por exemplo, um diagnóstico no início do ano letivo e um balanço do trabalho desenvolvido no final.

Neste trabalho não se verificou existir correlação entre o pensamento computacional e outras áreas curriculares, exploradas no primeiro ciclo, nomeadamente, a matemática, o português e o estudo do meio, ao contrário do que era espectável e do que indica a literatura.

Em termos de autoeficácia, também não se verificou nenhuma correlação entre esta e o pensamento computacional, mas sim que apresenta valores elevados e que tende a aumentar com a participação na iniciativa. Atendendo ao que refere Bandura (2001), “Unless people believe they can produce desired results and forestall detrimental ones by their actions, they have little incentive to act or to persevere in the face of difficulties”(p.10) estes dados poderão indicar que os alunos têm boas expectativas em

relação aos resultados e que, de uma forma otimista, superam as dificuldades através da persistência.

No âmbito deste trabalho foi publicado na Springer, volume 505, o artigo intitulado “Scratch 3 – Beginners Programming Course in 3rd Year of Primary School” (Anexo E). Houve ainda a apresentação do poster, visível no Anexo D, no âmbito do Regional HELIX 2018 International Conference on Innovation, Engineering and Entrepreneurship, dinamizado pela Universidade do Minho, na cidade de Guimarães.

1.2. Limitações e sugestões para investigações futuras

Nos estudos elaborados seria benéfico poder contar com um aumento do número de participantes, quer em termos de representatividade, quer em termos de grupos específicos, como seria o caso de grupos que se encontrem no 5.º ano de escolaridade, com diferentes tipos de participações na “Iniciativa IP1”, aumentando, assim, a faixa etária e a longevidade das implicações da medida.

Em futuras investigações, será necessário transpor os instrumentos de recolha de dados para uma aplicação digital, fator que irá facilitar bastante a análise dos dados, além de diminuir, drasticamente, os custos inerentes, visto que só o teste de pensamento computacional é composto por 16 páginas, que deverão ser impressas a cores. É óbvio que, posteriormente, terão de ser pensadas e ajustadas as metodologias e formas de aplicação dos instrumentos, visto já serem necessários outros meios de que, maioritariamente, as escolas do 1.º ciclo não dispõem, como por exemplo uma sala de informática ou um tablet por aluno. Ainda assim, esta transformação ocorrerá a curto prazo, para dar resposta às solicitações dos professores que estão envolvidos nesta iniciativa ou em outras que envolvem o pensamento computacional.

Após a transição dos instrumentos para uma aplicação digital, será solicitada autorização junto do MIMÉ para validar estes “novos” instrumentos e para que se possam registar e tratar os dados. Esta medida irá ampliar largamente o número de alunos envolvidos, aumentando o número de grupos e levando a investigação a todo o país. De salientar que o uso de questionários digitais e a sua divulgação online como estratégia de recolha de dados em estudos científicos já acontece, havendo a destacar diversas vantagens, como, por exemplo, a melhoria e a agilidade do processo de pesquisa, a possibilidade de o investigador recolher os dados de uma forma rápida e precisa, a existência de

participantes de diversas localizações geográficas sem custos elevados. De ter ainda em conta a imparcialidade e o anonimato, a não exposição dos participantes à influência do pesquisador, e a certeza de que os dados são inseridos automaticamente nas bases de dados. Não só se poupa imenso tempo ao investigador, como se eliminam as taxas de erro deste processo (Faleiros et al., 2016).

Uma das possíveis críticas que se podem assacar a este trabalho prende-se, primeiramente, com as características dos participantes, já que o grupo de controlo é mais pequeno em termos de indivíduos do sexo feminino do que o grupo de teste, fator que, por imposição das circunstâncias, não foi possível colmatar.

As limitações da presente investigação podem remeter-se para as variáveis que foram responsáveis pela quantificação dos resultados obtidos nas disciplinas de português, matemática e estudo do meio. Embora seja um facto que os alunos obtiveram no final do primeiro período (pré-teste) e no final do terceiro período (pós-teste) valores concretos, acontece que estes estão envolvidos em condicionantes/critérios alheios a esta investigação. Como, por este motivo, podem não resultar na aferição efetiva desses dados, futuramente, deve encontrar-se um instrumento que recolha, com rigor, os valores relativos a estas variáveis.

Para investigações futuras, seria bastante benéfico fazer ajustamentos ao programa de intervenção, de forma a envolver mais os conteúdos das disciplinas de português, matemática e estudo do meio, suscetíveis de trazer um maior benefício em termos de aprendizagens nestas áreas e na interligação entre as novas tecnologias e o meio envolvente. Outra investigação pertinente seria perceber qual o impacto que está a ter a nova matriz curricular do 2.º ciclo, onde surgem as TIC e onde um dos domínios incentiva a que o aluno conheça estratégias e ferramentas de apoio à criatividade, explore ideias, desenvolva o pensamento computacional e produza artefactos digitais criativos.

Por fim, referir que as sessões apresentadas no programa de intervenção serão, a curto prazo, ajustadas à nova versão do Scratch, a versão 3.6. Esta versão encontra-se com um visual mais atrativo e com diversas funções on-line; no entanto, nem todas as escolas têm os recursos adequados para acompanhar esta evolução e, por isso, continuarão disponíveis os recursos atuais, para os professores que optem por usar a versão para a qual as sessões foram idealizadas.

REFERÊNCIAS

Folha em branco

Referências

- Agrupamento de Escolas de Gouveia. (2018). *Projeto Educativo*.
<http://www.esgouveia.pt/images/CP1819/PE-18-21V1.pdf>
- Agrupamento de Escolas de Pinhel. (2018). *Projeto Educativo*.
<http://portal.aepinhel.org/index.php/component/attachments/download/59>
- Agrupamento de Escolas Pedro Álvares Cabral. (2017). *Projeto Educativo*.
<http://www.ae-pedroalvarescabral.net/index.php/documentos-gerais?download=35:pe1721>
- Akker, J. der, Keursten, P., & Plomp, T. (1992). The Integration of Computer Use in Education. *Pergamon Press*, 65–76.
- Almeida, M. E. B., & Valente, J. A. (2011). *Tecnologias e currículo: trajetórias convergentes ou divergentes?*
- Alves, N., Wangenheim, C. G. von, Hauck, J., Borgatto, A., & Andrade, D. (2019). *Uma Análise do Sequenciamento Pedagógico no Ensino de Computação na Educação Básica*. *Cbie*, 1. <https://doi.org/10.5753/cbie.sbie.2019.1>
- Amaral, J. da C. (1993). *Auto-Eficácia, Auto-Regulação e Desempenho na Realização de tarefas cognitivas [tese de mestrado]* [Lisboa: Instituto Superior de Psicologia Aplicada]. <https://repositorio.ispa.pt/handle/10400.12/302>
- Ambrósio, S. I. (1999). *Processo de conquista de autonomia em crianças de idade pré-escolar: Contributo para a análise das práticas educativas em contexto escolar [tese de mestrado]*. Instituto Superior de Psicologia Aplicada.
- ANACON. (2018). *Consumo de comunicações eletrónicas pelas crianças*.
<https://www.anacom.pt/render.jsp?contentId=1436288>
- Anderso, M. (2015). Technology Device Ownership. Em *Pew Research Center*.
https://www.pewresearch.org/wp-content/uploads/sites/9/2015/10/PI_2015-10-29_device-ownership_FINAL.pdf
- Araújo, L., Almeida, L., Spear-swerling, L., Velasquez, M. G., Lopes, J., & Oliveira, C. (2014). *Ensino da leitura no 1.º ciclo do ensino básico. Crenças, Conhecimentos e Formação dos Professores*. (Fundação F, Número 1).
<https://www.ffms.pt/FileDownload/71629ef8-a2c6-45d5-a925-e5d14b887c61/ensino-da-leitura-no-1o-ciclo-do-ensino-basico>
- Aureliano, V. C. O., & Tedesco, P. C. D. A. R. (2012). Avaliando o uso do Scratch como abordagem Alternativa para o processo de ensino-aprendizagem de programação.

- Anais do XX Workshop sobre Educação em Computação*, 10.
- Bandura, A. (2001). Social cognitive theory: An Agentic Perspective. *Annual Reviews Psychology*, 52, 1–26.
- Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? *ACM Inroads*, 2(1), 48–54. <https://doi.org/10.1145/1929887.1929905>
- Basawapatna, A., Koh, K. H., Repenning, A., Webb, D. C., & Marshall, K. S. (2011). Recognizing computational thinking patterns. *SIGCSE'11 - Proceedings of the 42nd ACM Technical Symposium on Computer Science Education*, 245–250. <https://doi.org/10.1145/1953163.1953241>
- Bear, G. G. (2010). *School discipline and self-discipline: A practical guide to promoting prosocial student behavior* (The Guilford Press (ed.)).
- Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers and Education*, 72, 145–157. <https://doi.org/10.1016/j.compedu.2013.10.020>
- Boucinha, R. M. (2017). *Aprendizagem do pensamento computacional e desenvolvimento do raciocínio [tese de doutoramento]*. Porto Alegre: Universidade Federal do Rio Grande do Sul.
- Bourguignon, C. (1994). *Comment intégrer l'ordinateur dans la classe de langue?* Paris: CNDP.
- Brackmann, C. P. (2017). *Desenvolvimento do pensamento computacional através de atividades desplugadas na educação básica [tese de doutoramento]*. Porto Alegre: Universidade Federal do Rio Grande do Sul.
- Brennan, K., & Resnick, M. (2012). *New frameworks for studying and assessing the development of computational thinking*. 1–25.
- Callear, D. (2000). *Teaching Programming: Some Lessons from Prolog*.
- Câmara Municipal de Pinhel. (2015). *Câmara Municipal de Pinhel*. <https://cm-pinhel.pt/>
- Carmo, B. S. do. (2013). *A robótica Educativa no Desenvolvimento do raciocínio matemático* [Universidade do Algarve]. <https://sapientia.ualg.pt/handle/10400.1/3625>
- Catarré, E., & Correia, L. M. (2019). Telemóveis e os Jovens: Uma Perspetiva Continuada sobre Utilização e Preocupações. Em *Informação sobre Radiação Eletromagnética em Comunicações Móveis*. FAQTOS.

- Charlton, P., & Luckin, R. (2012). *Time to reload? Computational thinking and computer science in schools. What researches says?* (U. of L. Lab, Knowledge - Institute of Education (ed.)).
- Chemers, M. M., Hu, L. T., & Garcia, B. F. (2001). Academic self-efficacy and first-year college student performance and adjustment. *Journal of Educational Psychology*, 93(1), 55–64. <https://doi.org/10.1037/0022-0663.93.1.55>
- Clark, J., Rogers, M., Spradling, C., & Pais, J. (2013). What, no canoes? Lessons learned while hosting a scratch summer camp. *Journal of Computing Sciences in Colleges*, 28(5), 204–210.
- Cochran-Smith, M. (2005). Teacher educators as researchers: multiple perspectives. *Teaching and Teacher Education*, 21(2), 219–225.
- Cohen, J. (1998). *Statistical power analysis for the behavioral sciences* (L. E. Associates (ed.); 2^a). <http://www.utstat.toronto.edu/~brunner/oldclass/378f16/readings/CohenPower.pdf>
- COMPUTING AT SCHOOL WORKING GROUP. (2012). *Computer Science: A curriculum for schools*. 3(September), 1–47.
- Costa, A. E. (2002). *Auto-eficácia e burnout. I*, 34–67.
- Coutinho, C. P. (2008). A qualidade da investigação educativa de natureza qualitativa: questões relativas à fidelidade e validade. *Educação Unisinos*, 12(1), 5–15. <https://doi.org/10.4013/5291>
- Cuban, L. (1993). Computers Meet Classroom: Classroom Wins. *Teachers College Record*, 15, 185–210.
- Direção Geral de Educação. (2015). *Iniciação à programação no 1º Ciclo do Ensino Básico - Linhas Orientadoras*. 1. http://www.erte.dge.mec.pt/sites/default/files/Projetos/Programacao/IP1CEB/linhas_orientadoras.pdf
- Drago, T. (2001). La Globalización es mucho más que pura economía. *Revista de Educación, Nº Extraordinario (Globalización y Educación)*, 83–89.
- Durak, H. Y., Yilmaz, F. G. K., & Bartin, R. Y. (2019). Computational thinking, programming self-efficacy, problem solving and experiences in the programming process conducted with robotic activities. *Contemporary Educational Technology*, 10(2), 173–197. <https://doi.org/10.30935/cet.554493>
- Esteves, A., Barbosa, A., Madeira, E., Barbosa, G., Oliveira, J., Cardoso, J., Coelho, L. S.,

- Neves, L., & Gonçalves, T. (2018). Global Schools. Propostas de integração curricular da Educação para o Desenvolvimento e Cidadania Global no 1.º e 2.º CEB. *Escola Superior de Educação do Instituto Politécnico de Viana do Castelo (ESE-IPVC)*. <http://www.sinergiased.org/index.php/revista/item/143>
- Everitt, B. S. (1975). Multivariate Analysis: The Need for Data, and other Problems. *British Journal of Psychiatry*, *126*(1), 237–240. <https://doi.org/https://doi.org/10.1192/bjp.126.3.237>
- Faleiros, F., Käßpler, C., Pontes, F. A. R., Silva, S. S. da C., Goes, F. dos S. N. de, & Cucick, C. D. (2016). Uso de questionário online e divulgação virtual como estratégia de coleta de dados em estudos científicos. *Texto & Contexto - Enfermagem*, *25*. <http://dx.doi.org/10.1590/0104-07072016003880014>
- Farias, A., Gomes, T., & Cabral, G. (2014). Uma Proposta Metodológica para o Ensino de Programação Baseado na relação entre Perfs Cognitivos, Padrões Pedagógicos e AutoRegulação dos Estudantes. *XXII Workshop sobre Educação em Computação - WEI. Anais do XXXIV Congresso da Sociedade Brasileira de Computação - CSBC.*, 1638–1646.
- Farias, H., Bonifácio, B., & Ferreira, R. (2015). Avaliando o Uso da Ferramenta Scratch para Ensino de Programação através de Análise Quantitativa e Qualitativa. *Anais do XXVI Simpósio Brasileiro de Informática na Educação (SBIE 2015)*, *1*(Sbie), 947. <https://doi.org/10.5753/cbie.sbie.2015.947>
- Feire, T., & Almeida, L. (2008). *Metodologia da Investigação em Psicologia e Educação* (Psiquilibrios (ed.); 5ª Edição).
- Figueiredo, A. D. (2001). Novos Media e Nova Aprendizagem in Novo Conhecimento – Nova Aprendizagem. *Fundação Calouste Gulbenkian*, 71–81.
- Floyd, F. J., & Widaman, K. F. (1995). Factor analysis in the development and refinement of clinical assessment instruments. *Psychological Assessment*, *7*(3), 286–299. <https://doi.org/10.1037/1040-3590.7.3.286>
- Fontes, C. (2018). *Indisciplina na Escolas*. <http://www.filorbis.pt/educar/indisciplina.htm>
- Forbes. (2012). *Why Estonia Has Started Teaching Its First-Graders To Code*. <https://www.forbes.com/sites/parmyolson/2012/09/06/why-estonia-has-started-teaching-its-first-graders-to-code/#c5fbb7b1aa3d>
- Fortin, M.-F. (2009). *Fundamentos e Etapas no Processo de Investigação* (Lusodidacta (ed.)).

- França, R. (2015). *Um modelo para a aprendizagem do Pensamento Computacional aliado à autorregulação [tese de mestrado]*. Recife: Universidade Federal de Pernambuco.
- GAAIRES, G. de A. e A. da I. da R. do. (2007). *Relatório*.
- Garofalo, D. (2018). *Tecnologia na Educação: como enriquecer o currículo com a robótica*. <https://novaescola.org.br/conteudo/12586/tecnologia-na-educacao-como-enriquecer-o-curriculo-com-a-robotica>
- Gatti, B. A. (2004). Estudos quantitativos em educação. *Educação e Pesquisa*, 30(1), 11–30. <https://doi.org/10.1590/s1517-97022004000100002>
- Geraldes, W. B. (2014). Programar É Bom Para As Crianças? Uma Visão Crítica Sobre O Ensino De Programação Nas Escolas. *Texto Livre: Linguagem e Tecnologia*, 7(2), 105–117. <https://doi.org/10.17851/1983-3652.7.2.105-117>
- Gomes, A. (2000). *Ambiente de suporte à aprendizagem de conceitos básicos de programação [tese de mestrado]*. Universidade de Coimbra.
- Gomes, A., & Mendes, A. (1997). *Suporte à aprendizagem da programação com o ambiente SICAS*. 2(December 2000).
- Gomes, A., Mendes, A. J., & Marcelino, M. J. (2004). *Avaliação e evolução de um ambiente de suporte à aprendizagem de programação*. June 2014.
- González, M. R. (2016). *Codigofabetización y pensamiento computacional en educación primaria y secundaria: validación de un interumento y evaluación de programas [tese de doutoramento]*. Madrid: UNED.
- Graczyńska, E. (2010). ALICE as a tool for programming at schools. *Natural Science*, 02(02), 124–129. <https://doi.org/10.4236/ns.2010.22021>
- Grover, S., & Pea, R. D. (2013). Computational Thinking in K–12: A Review of the State of the Field. *Educational Researcher*, 42(1), 38–43. <https://stanford.edu/~shuchig/publications.html>
- Guimarães, V. da F., & Abbad, G. da S. (2015). Autoeficácia no uso do computador em situações de aprendizagem: uma análise da literatura internacional. *Revista Psicologia Organizações e Trabalho*, 15(2), 170–187. <https://doi.org/10.17652/rpot/2015.2.319>
- Gwénaelle, J., Usher, E. L., & Bressoux, P. (2011). Sources of Self-Efficacy: An Investigation of Elementary School Students in France. *Journal of Educational Psychology*, 103(3), 649–663. <https://doi.org/10.1037/a0024048>
- Hadinugrahaningsiha, T., Rahmawati, Y., & Ridwan, A. (2017). Developing 21st century

- skills in chemistry classrooms: Opportunities and challenges of STEAM integration. *AIP Conference Proceedings*, 1868. <https://doi.org/10.1063/1.4995107>
- Hadwin, A., Järvelä, S., & Miller, M. (2011). Self-Regulated, Co-Regulated, and Socially Shared Regulation of Learning. In B. Zimmerman, & D. Schunk (Eds.). *Handbook of Self-Regulation of Learning and Performance*, 65–84. <https://doi.org/10.4324/9780203839010.ch3>
- Hair, J. J. F., Black, W. C., & Sant’Anna, A. S. (2009). *Análise multivariada de dados (6a. ed.)*.
- Hayling, C. C., Cook, C., Gresham, F. M., State, T., & Kern, L. (2007). An Analysis of the Status and Stability of the Behaviors of Students with Emotional and Behavioral Difficulties. *Journal of Behavioral Education*, 17(1), 24–42. <https://doi.org/10.1007/s10864-007-9059-5>
- Hedges, L. V. (1981). Distribution Theory for Glass’s Estimator of Effect Size and Related Estimators. *Journal of Educational Statistics*, 6 (2), 107–128. <https://www.jstor.org/stable/1164588>
- Horta, M. J., Mendonça, F., & Nascimento, R. (2012). *Metas Curriculares Tecnologias de Informação e Comunicação 7º e 8º ano*.
- Hoy, A. W., & Spero, R. B. (2005). Changes in teacher efficacy during the early years of teaching: A comparison of four measures. *Teaching and Teacher Education*. *Teaching and Teacher Education*, 21, 343–356. <https://doi.org/10.1016/j.tate.2005.01.007>
- Isbell, C. L., Stein, L. A., Cutler, R., Forbes, J., Fraser, L., Fraser, L., Impagliazzo, J., Russ, S., & Thomas, R. (2009). (Re) Defining Computing Curricula by (Re) Defining Computing. *SIGCSE Bulletin*, 41(4), 195–207.
- Jardim, J. (2003). *O Método da Animação: Manual Para o Formador (AVE (ed.))*.
- Jenkins, T. (2002). On the Difficulty of Learning to Program. *3rd Annual LTSN-ICS Conference, Loughborough University*, 53–58.
- João, S. M. (2003). *Programa de Tecnologias da Informação e da Comunicação 9º e 10º anos*. Ministério da Educação - Direcção-Geral da Inovação e Desenvolvimento Curricular.
- Kurshan, B. (2016). Thawing from a Long Winter in Computer Science Education. *Forbes*, 1–2.
- Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., Malyn-Smith, J., & Werner, L. (2011). Computational thinking for youth in practice. *ACM Inroads*, 2(1),

- 32–37. <https://doi.org/10.1145/1929887.1929902>
- Luiz, S. D. S. S., & Dantas, V. F. (2019). *Pensamento Computacional e Gênero: Avaliando Competências em Séries do Ensino Fundamental II*. 415–423. <https://doi.org/10.5753/ctrl.2019.8913>
- Lusa. (2017). Projeto vai dar formação em programação «Scratch» a 5.000 alunos e 500 professores. *Diário de Notícias*. <http://www.dn.pt/lusa/interior/projeto-vai-dar-formacao-em-programacao-scratch-a-5000-alunos-e-500-professores-8536562.html>
- 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*, 41, 51–61. <https://doi.org/10.1016/j.chb.2014.09.012>
- Major, S. (2011). *Avaliação de Aptidões Sociais e Problemas de Comportamento em Idade Pré- Escolar: Retrato das Crianças Portuguesas* [Faculdade de Psicologia e de Ciências da Educação - Universidade de Coimbra]. <http://hdl.handle.net/10316/17774>
- Maloney, J., Peppier, K., Kafai, Y. B., Resnick, M., & Rusk, N. (2008). Programming by choice: Urban youth learning programming with scratch. *SIGCSE'08 - Proceedings of the 39th ACM Technical Symposium on Computer Science Education, January*, 367–371. <https://doi.org/10.1145/1352135.1352260>
- Marques, R. (2001). *História Concisa da Pedagogia* (Plátano (ed.)).
- Martínez, I., & Soria, M. (2006). Autoeficacia en el trabajo: el poder de creer que tú puedes. *Estudios financieros. Revista de trabajo y seguridad social. Comentarios, casos prácticos. Recursos humanos*, 279, 175–202.
- Maya, I., Pearson, J. N., Tapia, T., Wherfel, Q. M., & Reese, G. (2015). Supporting all learners in school-wide computational thinking: A cross-case qualitative analysis. *Computers & Education*, 82, 263–279. <https://doi.org/https://doi.org/10.1016/j.compedu.2014.11.022>
- Melo, C. I. B. de. (2017). Relações de gênero na matemática: o processo histórico-social de afastamento das mulheres e algumas bravas transgressoras. *Revista Ártemis*, 24(1), 189–200. <https://doi.org/10.22478/ufpb.1807-8214.2017v24n1.34424>
- Ministério da Educação e Ciência. (2013). *Despacho n.º 5048-B/2013*. https://www.portaldascolas.pt/imageserver/plumtree/portal/matnet/Despacho_5048B_2013.pdf
- Moreira, J. M. (2004). *Questionários: Teoria e prática* (Almedina (ed.)).

- National Research Council. (2010). *Report of a Workshop on the scope and nature of Computational Thinking*. www.nap.edu
- Neves, S. P., & Faria, L. (2007). Auto-eficácia acadêmica e atribuições causais em Português e Matemática. *Análise Psicológica*, 4 (XXV), 35–652. <http://www.scielo.mec.pt/pdf/aps/v25n4/v25n4a09.pdf>
- Ng, E., & Bereiter, C. (1991). Three Levels of Goal Orientation in Learning. *Journal of the Learning Sciences*. *The Journal of the Learning Sciences*, 1, 243–271. <https://doi.org/https://doi.org/10.1080/10508406.1991.9671972>
- Nunes, D. J. (2011). Ciência da Computação na Educação Básica. *Jornal da Ciência, setembro*.
- Obama, B. (2013). Don't Just Play on Your Phone, Program It. *The White House Blog*. <https://obamawhitehouse.archives.gov/blog/2013/12/09/don-t-just-play-your-phone-program-it>
- Oliveira, P. F. D. de. (2018). *O uso das tecnologias digitais numa escola com 3º ciclo do ensino básico-efeitos sobre a autorregulação da aprendizagem*. Instituto de Educação: Universidade de Lisboa.
- Pajares, F. (2006). Self-efficacy during childhood and adolescence: Implications for teachers and parents. Em Age Publishing (Ed.), *Self-efficacy beliefs of adolescents* (pp. 339–367). https://pdfs.semanticscholar.org/bb2f/53f4ba6dd197ce64673f880869286eb4eb4ebd.pdf?_ga=2.202034712.882163845.1569151729-1089427227.1569151729
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas* (Basic Books (ed.)).
- Papert, S. (1996). Looking at Technology Through School-Colored Spectacles. *MIT Media Lab*.
- Pattis, R. E. (1990). A philosophy and example of CS-1 programming projects. *SIGCSE Bulletin*, 34–39.
- Pedro, N., Soares, F., Matos, J. F., & Santos, M. (2008). *Utilização de Plataformas de Gestão de Aprendizagem em Contexto Escolar*. Relatório do Estudo Nacional do Ministério da Educação.
- Pereira, A. (2008). *Guia prático de utilização: análise de dados para ciências sociais e psicologia*. (7ª). Edições Sílabo.
- Pereira, C. C., & L. Celis, N. (2018). Autoeficácia e Estilos de Aprendizagem: um estudo comparativo entre os alunos que frequentam o ensino artístico especializado da

- música e os alunos que frequentam o ensino regular. *Convergências - Revista de Investigação e Ensino das Artes*, VOL XI (22). <http://convergencias.ipcb.pt>
- Pereira, S., & Pereira, L. (2011). Políticas tecnológicas educativas em Portugal: do Projecto Minerva à Iniciativa e-Escolinha. Em U. do M. C. de E. de C. e Sociedade (Ed.), *Congresso Nacional «Literacia, Media e Cidadania»* (pp. 158–168).
- Phillips, P. (2008). *Computational Thinking: A Problem-Solving Tool for Every Classroom*. <http://education.sdsc.edu/resources/CompThinking.pdf>
- Piva Jr, D., & Freitas, R. L. (2010). Estratégias para melhorar os processos de abstração na disciplina de Algoritmos. *Simpósio Brasileiro de Informática na Educação*, 21. <http://www.br-ie.org/pub/index.php/sbie/article/view/1464>
- Ponte, J. P. da. (1986). *O computador - Um instrumento da Educação*. Texto Editora.
- Ponte, J. P. da. (1994). *O Projecto MINERVA Introduzindo as NTI na Educação em Portugal*. Relatório do Projecto MINERVA: Ministério da Educação.
- Ponte, J. P. da. (2000). Tecnologias de Informação e Comunicação na Formação de Professores: Que desafios? *Revista Iberoamericana de Educação*, nº 24, 63–90.
- Ponte, J. P. da. (2002). Investigar a nossa própria prática In GTI (Org). Em *Refletir e investigar sobre a prática profissional* (pp. 5–28). APM.
- Prensky, M. (2006). *Don't Bother Me Mom - I'm Learning! - How computer and video games are preparing your kids for 21st century success and how you can help!* (Paragon House (ed.)).
- Ramos, José Luís. (1999). A Integração do Computador na escola e no Currículo: Problemas e Perspectivas. *Inovação*, 12, 89–106.
- Ramos, José Luis, & Espadeiro, R. G. (2016). *Pensamento Computacional na Escola e práticas de avaliação das aprendizagens. Uma revisão sistemática da literatura*. http://dspace.uevora.pt/rdpc/bitstream/10174/14227/1/challenges_2015br.pdf
- Ramos, José Luís, & 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 - ISSN 1646-933X*, 7(2), 4–25.
- Ramos, José Luís, & Espadeiro, R. G. (2016). *Estudos de Avaliação 1º Ciclo Iniciação à programação*. DGE – Direção Geral de Educação.
- Ramos, José Luís, Espadeiro, R. G., Carvalho, J. L., Gancho do Maio, V., & Matos, J. M. (2009). *Iniciativa Escola, Professores e Computadores Portáteis: Estudos de Avaliação*. CRIE, MEC/DGIDC.

- Raymundo, V. P. (2009). Construção e validação de instrumentos um desafio para a Psicolinguística. *Letras de Hoje*, 44(3), 86–93.
- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. (2009). Scratch: Programming for all. *Communications of the ACM*, 52(11), 60–67. <https://doi.org/10.1145/1592761.1592779>
- Rich, K. M., Strickland, C., Binkowski, T. A., Moran, C., & Franklin, D. (2018). K-8 learning trajectories derived from research literature: sequence, repetition, conditionals. *ACM Inroads*, 9(1), 182–190. <https://doi.org/10.1145/3183508>
- Sá, J. (1997). *Estratégias de desenvolvimento do pensamento científico em crianças do 1º ciclo do ensino básico* [Universidade do Minho]. <http://hdl.handle.net/1822/8165>
- Sáez-López, J. M., & Sevillano-García, M. L. (2017). Sensors, programming and devices in Art Education sessions. One case in the context of primary education / Sensores, programación y dispositivos en sesiones de Educación Artística. Un caso en el contexto de Educación Primaria. *Cultura y Educación*, 29(2), 350–384. <https://doi.org/https://doi.org/10.1080/11356405.2017.1305075>
- Sandholtz; Haymore Judith; And Others. (1997). *Teaching with Technology: Creating Student-Centered Classrooms* (NY: Teachers College Press (ed.)).
- Schunk, D. H., & Zimmerman, B. J. (2007). Influencing children's self-efficacy and self-regulation of reading and writing through modeling. *Reading and Writing Quarterly*, 23, 7–25.
- Schwarzer, R., & Scholz, U. (2000). Cross-cultural assessment of coping resources: The general perceived self-efficacy scale. *Health Psychology and Culture*.
- Selby, C. C., & Woollard, J. (2013). *Computational Thinking: The Developing Definition*. <https://eprints.soton.ac.uk/356481/>
- Serafini, G. (2011). Teaching Programming at Primary Schools: Visions, Experiences, and Long-Term Research Prospects. Em Springer (Ed.), *Proceedings of the 5th international conference on Informatics in Schools: situation, Evolution and Perspectives* (pp. 143–154).
- Shailaja, J., & Sridaran, R. (2015). Computational Thinking the Intellectual Thinking for the 21st century. *International Journal of Advanced Networking & Applications, May 2015 Special Issue*, 39–46.
- Shapiro, S. S. ., & Wilk, M. B. (1965). An analysis of variance test for normality (complete

- samples). *Biometrika*, 52(3), 591–611.
- Silva, Á. A. T. Da. (2004). *Ensinar e aprender com as TIC —um estudo sobre as atitudes, formação, condições de equipamento e utilização nas escolas do 1º ciclo do ensino básico do concelho de Cabeceiras de Basto* [Universidade do Minho]. <http://repositorium.sdum.uminho.pt/bitstream/1822/3285/1/TESE - Ensinar e Aprender com as TIC.pdf>
- Silva, I., Marques, L., Mata, L., & Rosa, M. (2016). Orientações Curriculares para a Educação Pré-Escolar. *Ministério da Educação/Direção-Geral da Educação*.
- Slomski, V. G., & Martins, G. de A. (2008). O conceito de professor investigador: os saberes e as competências necessárias à docência reflexiva na área contábil. *Revista Universo Contábil*, 4(4), 6–21.
- Sobral, J. B. M., Kittel, R., Hoppen, A., Mito, A. E., & Cruz, F. A. S. (2015). *Computação no Ensino Fundamental na Escola Pública*. <https://repositorio.ufsc.br/handle/123456789/134696>
- Teodoro, V. D., & Freitas, J. C. de. (1992). *Educação e computadores. Série - Desenvolvimento dos Sistemas Educativos*. GEP/Educação.
- Trindade, R., & Cosme, A. (2010). *Educar e Aprender na Escola: Questões, desafios e respostas pedagógicas*. <https://doi.org/302576/09>
- Türker, P. M., & Pala, F. K. (2020). The Effect of Algorithm Education on Students' Computer Programming Self-Efficacy Perceptions and Computational Thinking Skills. *International Journal of Computer Science Education in Schools*, 3(3), 19–32. <https://doi.org/10.21585/ijcses.v3i3.69>
- Valente, J. A. (2016). Integração do Pensamento Computacional no Currículo da Educação Básica: Diferentes Estratégias Usadas e Questões de Formação de Professores e Avaliação do Aluno. *Revista e-Curriculum*, 14(3), 864–897. <https://revistas.pucsp.br/index.php/curriculum/article/view/29051>
- Valle, A., Cabanach, R. G., Núñez, J. C., González-Pienda, J., Rodríguez, S., & Piñeiro, I. (2003). Multiple goals, motivation and academic learning. *British Journal of Educational Psychology*, 73(1), 71–87. <https://doi.org/10.1348/000709903762869923>
- Vieira, D. A. (2012). *Transição do Ensino Superior para o Trabalho: o poder da auto-eficácia e dos objetivos profissionais*. (E. Politema (ed.)). <http://hdl.handle.net/10400.22/668>
- Wing, J. M. (2006). *Computational Thinking* (C. of the A.-S. managed Systems (ed.);

-
- Vol. 49, Número 3, pp. 33–35). <https://www.cs.cmu.edu/~15110-s13/Wing06-ct.pdf>
- Wing, J. M. (2014). *Computational thinking benefits society*. *Social issues in computing*. Academic Press. <http://socialissues.cs.toronto.edu/2014/01/computational-thinking/>
- Zapalowski, V. (2011). *Análise quantitativa e comparativa de linguagens de programação*. Universidade Federal do Rio Grande do Sul.
- Zapata-ros, M. (2015). Pensamiento computacional : Una nueva alfabetización digital
Computational Thinking : A New Digital Literacy. *Revista de Educación a Distancia*, 46(4). <https://doi.org/10.6018/red/46/4>
- Zhou, N. (2011). Computational Thinking In Pre-Service Teacher Education : Integrating CT. *Technology And Teacher Education Annual*. *Associationn for the Advancement of Computing in Education*, 62–63.

ANEXOS

Folha em branco

Anexo A: Autorização de Marcos Román González para a utilização e adaptação do seu teste de pensamento computacional



António Marques <prof.informatica.marques@gmail.com>

Solicitud de autorización para usar la prueba

MARCOS ROMAN GONZALEZ <mroman@edu.uned.es>
Para: António Marques <prof.informatica.marques@gmail.com>

8 de janeiro de 2018 às 12:09

Ok, tienes mi autorización.

¿Necesitas algo más por mi parte?

Saludos,

MARCOS ROMÁN GONZÁLEZ

Profesor e Investigador

Secretario Académico del Departamento de Métodos de Investigación y Diagnóstico en Educación I (MIDE I)

Facultad de Educación de la Universidad Nacional de Educación a Distancia (UNED)

C/ Juan del Rosal, nº14, Despacho 2.18. C.P. 28040. MADRID

Teléfono: 91 398 90 37

Web: <http://goo.gl/oox5Qn>

De: António Marques <prof.informatica.marques@gmail.com>

Enviado: lunes, 8 de enero de 2018 13:02

Para: MARCOS ROMAN GONZALEZ

Asunto: RE: Solicitud de autorización para usar la prueba

[Citação ocultada]

[Citação ocultada]

Anexo B: Autorização do MIME

Dados do Inquérito

Número de registo:

0636100001

Designação:

A iniciativa "Iniciação à Programação no 1.º Ciclo do Ensino Básico" promove o pensamento computacional? Análise da implementação da medida a alunos do 3.º ano.

Descrição:

A iniciativa "Iniciação à Programação no 1.º Ciclo do Ensino Básico" está direcionada para os alunos do 3.º e do 4.º anos de escolaridade e tem como principal finalidade que "os alunos não só aprendam a programar mas, ao mesmo tempo, aprendam programando" (DGE, 2015, p. 3) promovendo o ensino em termos de ciências da computação e, assim, alargar os seus horizontes e a forma como veem o computador. Nesse sentido, foram construídos instrumentos para estudar o impacto desta iniciativa em geral, bem como a impacto de um programa que foi contruído também no âmbito desta Tese, por uma equipa de cinco professores de informática.

O Teste de Pensamento Computacional, foi desenvolvido, originalmente, por Roman (2015), em Espanha, e cujos estudos de adaptação à população portuguesa foram levados a cabo no âmbito da Tese deste Doutoramento. Outros países, como o Brasil, têm também desenvolvido processos de investigação e adaptação do instrumento (Boucinha, 2017; Brackmann, 2017).

Em suma, como se trata de uma iniciativa recente espera-se que os resultados do estudo a realizar possam levar ao conhecimento detalhado dos contributos que a mesma está a fomentar no ensino/aprendizagem dos alunos envolvidos.

De salientar que este estudo vem colmatar uma lacuna já referenciada por José Ramos e Rui Espadeiro, nomeadamente "...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..." (José Luis Ramos & Espadeiro, 2016, p. 19)

Referências

Boucinha, R.M.: Aprendizagem do pensamento computacional e desenvolvimento do raciocínio. Tese de doutoramento, Universidade Federal do Rio Grande do Sul (2017)

Brackmann, C.P.: Desenvolvimento do pensamento computacional através de atividades desplugadas na educação básica. Tese de doutoramento, Universidade Federal do Rio Grande do Sul (2017)

Educação, D.G.: Iniciação à programação no 1.º ciclo do ensino básico. Linhas orientadoras gerais. http://www.erte.dge.mec.pt/sites/default/files/Projetos/Programacao/IP1CEB/linhas_orientadoras.pdf. Accessed 24 Jan 2017

Joet, G., Usher, E., Bressoux, P.: Sources of self-efficacy: an investigation of elementary school students in France. *J. Educ. Psychol.* 103(3), 649–663 (2011)

Ramos, J.L., Espadeiro, R.G.: Iniciação à Programação no 1.º Ciclo do Ensino Básico. Estudos de Avaliação do projeto-piloto. DGE – Direção Geral de Educação, Portugal (2016)

Román, M.: Test de Pensamiento Computacional: principios de diseño, validación de contenido y análisis de ítems. In: Murga Menoyo, M.A. (eds.) *Perspectivas y Avances de la investigación*, pp. 279–302. UNED, Madrid (2015)

Objectivos:

Objetivos Principais

1. Avaliar o impacto da Iniciativa nos conhecimentos dos alunos em termos de princípios e conceitos fundamentais das Ciências da Computação;
2. Avaliar o impacto da iniciativa em termos de mudanças nas competências e conhecimentos, bem como o impacto nas aprendizagens;
3. Analisar a eficácia de um programa que pretende promover o pensamento computacional e desenvolver competências para resolução de problemas, formas divergentes de pensar, criatividade, comunicação e trabalho em grupo.

Periodicidade:

Pontual

Data do início do período de recolha de dados:

10-09-2018

Data do fim do período de recolha de dados:

28-06-2019

Universo:

Cerca de 300 alunos

Unidade de observação:

Alunos do 3º e 4º anos de Agrupamentos de Escolas da Zona Centro

Método de recolha de dados:

Questionários

Inquérito registado no Sistema Estatístico Nacional:

Não

Inquérito aplicado pela entidade:

Sim

Dados adicionais

Estado:

Aprovado

Avaliação:

Exmo.(a) Senhor(a) António Soares Marques
Venho por este meio informar que o pedido de realização de inquérito em meio escolar é autorizado uma vez que, submetido a análise, cumpre os requisitos, devendo atender-se às observações aduzidas.
Com os melhores cumprimentos
José Vítor Pedroso
Diretor-Geral
DGE

Anexo C: Autorização dos Encarregados de Educação

Exmo(a). Sr(a).

Encarregada de Educação

O meu nome é António Marques e sou docente no Agrupamento de Escolas de Pinhel além de ser estudante do Doutoramento em Educação lecionado pela Faculdade de Ciências Sociais e Humanas da Universidade da Beira Interior. De momento estou a desenvolver uma Investigação cujo tema é: ***“Iniciação à Programação no 1.º Ciclo do Ensino Básico”: Análise da implementação da medida a alunos do 3.ºano***, visando os seguintes objetivos:

1. Avaliar os impactos da Iniciativa no pensamento computacional dos alunos;
2. Avaliar os impactos da Iniciativa nos conhecimentos dos alunos em termos de princípios e conceitos fundamentais das Ciências da Computação;
3. Identificar e disseminar práticas educativas inovadoras na aplicação da iniciativa.

A referida investigação tem como orientadora a professora Doutora Sandra Carina Guimarães.

Para tal, gostaria de **solicitar a sua autorização** e colaboração para recolher dados relativos ao seu educando e seu processo de ensino aprendizagem mais concretamente no que envolva as aulas de iniciação à programação que decorrem da promoção da medida “IP1 - Iniciação à Programação no 1.º Ciclo do Ensino Básico” promovida pela Direção-Geral da Educação.

Saliento ainda que o estudo não envolve qualquer alteração no tratamento estabelecido ao longo das aulas e que a recolha de dados será feita mediante a análise dos registos bibliográficos dos alunos e dos resultados obtidos na realização de diferentes tarefas que constam nos planos de aula.

Certo que a sua autorização me irá ajudar a desenvolver este estudo, agradeço antecipadamente a sua colaboração e disponibilidade.

PEDIDO DE AUTORIZAÇÃO AO ENCARREGADO DE EDUCAÇÃO

Eu _____, Encarregado(a) de Educação do aluno (a)

_____, autorizo/não autorizo a recolha de dados no âmbito

do estudo de investigação ***“Iniciação à Programação no 1.º Ciclo do Ensino Básico”:***

Análise da implementação da medida a alunos do 3.ºano.

Anexo D: Poster apresentado no Regional Helix 2018 Conference

REGIONAL HELIX
2018 CONFERENCE
June 27th - 29th
Guimarães
PORTUGAL

International Conference on
Innovation, Engineering and Entrepreneurship

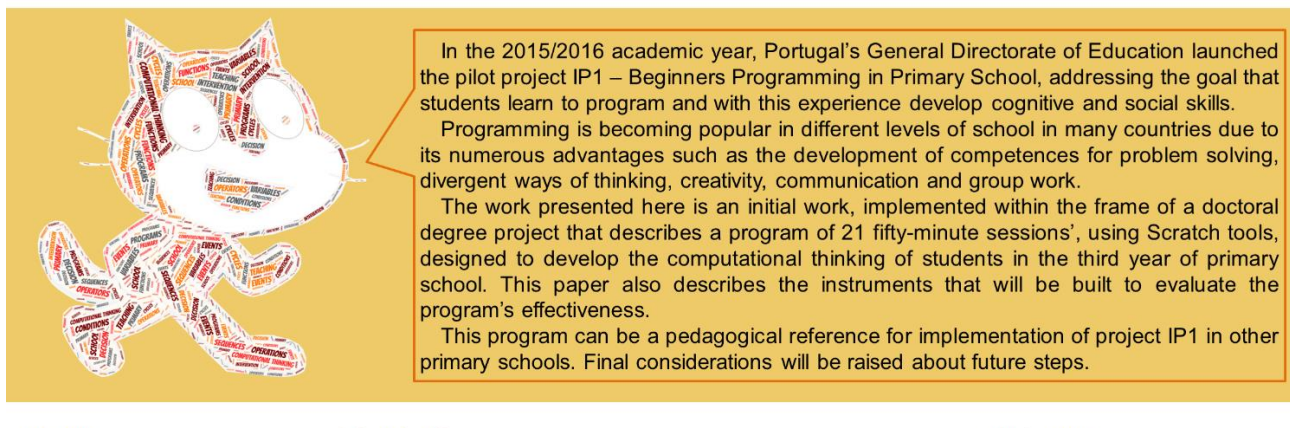
Scratch 3 – Beginners Programming Course in 3rd Year of Primary School

In the 2015/2016 academic year, Portugal's General Directorate of Education launched the pilot project IP1 – Beginners Programming in Primary School, addressing the goal that students learn to program and with this experience develop cognitive and social skills.

Programming is becoming popular in different levels of school in many countries due to its numerous advantages such as the development of competences for problem solving, divergent ways of thinking, creativity, communication and group work.

The work presented here is an initial work, implemented within the frame of a doctoral degree project that describes a program of 21 fifty-minute sessions', using Scratch tools, designed to develop the computational thinking of students in the third year of primary school. This paper also describes the instruments that will be built to evaluate the program's effectiveness.

This program can be a pedagogical reference for implementation of project IP1 in other primary schools. Final considerations will be raised about future steps.



1 Introduction

Nowadays, any service or product that we consume interrelates with, whether directly or indirectly, some process or decision determined by a computer chip. Almost everyone, irrespective of their age, needs some basic technological knowledge. Some authors [1, 2] maintain that computational thinking is literacy for the 21st century. With this technology so very present, a new learning profile is appearing, the so called "digital natives", a person who is accustomed to working on various tasks simultaneously, with easy and fast access to information, interconnected via networks and based on immediate and frequent communications [3]. Since a very high percentage of professions require digital competences [4] people should not act only as consumers, they also need to be creators. Complex education reports point in this direction [5]. The contribution of technology to educational content is not only necessary but also an unmissable reality for our schools.

DePette Paper [6] having introduced a programming language for educational purposes and transferable to other spheres of life in 1980, it was only in 2006 that Jeanette Wing popularized the term "computational thinking" and set out the necessary theoretical framework. This concept "involves solving problems, designing systems, and understanding human behaviour, by drawing on the concepts fundamental to computer science" [7].

Multiple definitions and updates were to follow in search of a consensus that is still lacking in the field [8]. One of the most consensual definitions of CT - computational thinking, include the "creative, critical and strategic human capacity to know how to use the fundamentals of computing in the most diverse areas of knowledge, with the purpose of identifying and resolving problems, whether individually or collaboratively, through clear steps, so that either a person or a machine may execute them efficiently" [9].

For the majority of people, programming and coding are perceived as complex actions based on what are considered to be advanced levels of technological knowledge and thus accessible only those with certain abilities. However, a number of authors defend an opposing idea. Paper [6], for example, advocates that programming languages should have different levels, from the simple to the complex, and thus be accessible to all.

The increasing number of programming languages, such as Scratch [10] or Alice [11], free and user friendly, easier to grasp and begin using, raises the interest of programming among pre-university level. In recent years, primary and secondary schools, mainly all in Anglo-Saxon countries, have embarked on initiatives striving to introduce CT. Worldwide, school curricula have been updated in this field to include students at younger ages via an enormous variety of educational initiatives [12, 13]. Valente [14] systematized these as follows: (a) activities without resource of technologies; (b) programming in Scratch; (c) pedagogical robotics; (d) producing digital narratives; (e) designing games and (f) using of simulators. However, even though these technologies are diverse and susceptible to multiple applications, these interventions frequently emerge as sporadic or they only target an elite group or they lack overall interconnection.

In Portugal the most recent initiatives launched by the Ministry of Education and Science in this field include the pilot project "Beginners Programming in Primary School and the inventory and support for Programming and Robotics Clubs existing nationwide" [15]. However, the literature states that there are only occasional studies on such interventions in K12 school contexts (pre-university) [8] and that these studies are especially rare in primary school [16].

In Brazil, Campos and his colleagues [17] carried out one of the first attempts to measure improvements to CT, through pre- and post-testing. The results did not return any statistically significant improvement but the activities were described as gratifying. In Spain, Roman [18] developed a CT test with 28 items with four response choices for a 45 min test for school children aged between 12 and 13. Fabre and his colleagues [19] developed a six-session program in 26 primary school establishments in the Netherlands and received positive feedback from teachers and students but there was no metric for participants' levels of satisfaction or statistical data on changes in CT performance standards.

Specifically in Portugal, there have been diverse studies carried out to evaluate the IP1 program in order to boost the effectiveness of future proposals [15]. The conclusions convey a very positive general evaluation by school headmasters, teachers and school initiative coordinators with regard to the level of attainment of the project goals and also highlighted the importance of the guidelines drawn up, which helped reinforce the continuity of the program. However, Ramos and Espalino [15] remind us of the need to evaluate the impact of these programs in terms of the changes in student competences and knowledge.

In accordance with this framework, this study seeks to describe a program fostering CT and to analyse its effectiveness.

2 Program Scratch 3

The program presented here stems from the pilot project "Beginners Programming in Primary School" [20], run by the GDE - General Directorate of Education, as an intervention proposed for a field that is integrated into the education program of a number of schools through recourse to the Scratch programming application (3rd and 4th years).

The literature details various ways of applying Scratch to different levels of teaching [21] and, in terms of the challenges set by the pilot project, each teacher and school established their own interventions. What occurred in practice was that in many schools, the program featured some random activities, detached from the theoretical framework that would have endowed them with meaning or coherence. Another issue was that there were also many teachers engaged who lacked any sort of guide for orienting their work, and consequently were unable to evaluate the effectiveness of their interventions.

In order to avoid these problems, a group of five computer teachers (one is author of this paper), who are developing the pilot project in 3rd year of primary school, designed originally this program which intends to guide the work of teaching programming in partnership with teachers from other subject areas (see Table 1). The program to foster computational thinking among participants students follow literature review, guidelines handed down by the Ministry of Education (ME) for the program and the curricular targets to be attained by third year students. Specific goals (e.g. activities planning, symbolic representation) and soft skills (e.g. group work, communication) proposed by ME are also integrated. These 21 sessions, 50 min each, through use of the Scratch tools (e.g. characters, stages, scenarios, movements, sound, sensors, etc.) allow students to develop fundamental concepts in the field of programming such as sequences, parallel operations, events, decision conditions/structures, cycles, operators, data/variables, and simple functions, among others.

According to Calicut [22] when teaching programming is relevant to: (a) slow pace teaching to facilitate the concepts integration; (b) teach one by one topic to avoid confusions; (c) review previous topics and (d) sort the subjects by degree of difficulty. These principles guided this specific program.

The training is planned around exercises with increasing levels of difficulty (most of the times with the introduction of new blocks of code, which are explored at the

4 Final Considerations

Computational thinking in education is, to some extent, an inevitable trend. There are various studies demonstrating how students with training and education in programming are better prepared for future daily professional tasks and challenges [10]. Based on their study findings, Sáez-López and Sevilla-García [21] recommend that those responsible for educational policies integrate them into primary school teaching programs and into subject areas of art, music and technology.

However, such practices are not always reflective and intentional, and very often the teaching of programming languages teaching only a very reduced number of students or only happens on an occasional basis. This, as proposed by the ME, is an opportunity for schools to develop structural CT programs. However, according to literature (e.g. [6]), it is also important to assess these programs, in areas like computational thinking skills and their impact on learning, important evaluation factors. Therefore, this paper is only one first step, within the frame of a doctoral degree project doctoral, presenting one program and their process of evaluation. In the near future authors will be obtained the authorizations from the respective School Board and parents of the children participated (3rd and 4th years of primary school) in two studies: evaluation of program and psychometric study of instruments. After gathering all data, statistical analysis of the instruments and the program will be carried out according to the goals defined. The results and discussion will continue to be just a small contribution towards building knowledge in this area. More research is clearly required to establish how to teach children use programming tools and how they influence their development (e.g. problem-solving and computational thinking abilities).

3 Evaluation of Program Effectiveness

Several instruments will be used to evaluate the effectiveness of the program.

The *Computational Thinking Test*, originally developed by Roman [18] for application in Spain involves measuring the level of CT development with a focus on the fundamental concepts of programming such as sequences and repetitions. The adaptation and validation of the Portuguese version will be done within the scope of this work (with both experts and prior pilot studies). In Brazil, there have also been psychometric studies applying this test [22, 24]. The test contains 28 exercises: four questions about simple sequences; four questions about the cycle of repetition; the number of times; four questions about the cycle and repetition through to a determined condition; four questions about simple conditions; (condition - action); four questions about conditions compound (condition - action - exception); four questions about the cycle of repetition as a condition for verification; and four questions about simple functions so as to divide a problem by grouping tasks. The level of complexity and difficulty of the exercises rises throughout the test and always presents four response options (A, B, C and D), of which only one is correct. The test lasts for 45 min and begins with three practice exercises. Carrying out this test also explores the capacity to formulate and resolve problems either directly or in phased approaches.

Self-efficacy in Computational Thinking is measured through a specific scale developed by the authors in accordance with the literature (e.g. [25]). The scale evaluates participant perceptions in regard to their competences in the computer science field (e.g. "I know how to combinatorially represent sequences of actions for activities") with responses made according to a five-point Likert scale (1 - totally disagree to 5 - totally agree). Validation for the portuguese population still comply with all the requirements for its appropriate application.

In addition to the demographic data (e.g. academic qualifications of parents) as well as a technology user information (e.g. "Do you use a computer daily?") were also collected.

The data gathering process also extends to the participants' academic records in three subjects: Portuguese, Mathematics and Environmental Studies.

We would further add that evaluating the program's efficiency spans two phases: pre (MI) and post (M2) application into two groups of students.

Session	Activity	Sequences	Repetition cycles	Conditional operations	Communication and symbolic representation	Random logical	Random operators	Variables	User interface elements
1	Scratch Presentation	X	X						
2	Customize avatar	X	X	X					
3	Insert stages	X	X						
4	Conditional statements	X	X	X	X				
5	Message and sounds	X	X	X	X				
6	Effects	X	X	X					
7	Sensors	X	X	X	X				
8	Scenes II	X	X	X	X				
9	Customize stages	X	X	X	X				
10	Visibility (eyes)	X	X		X				
11	Customize stages and actors	X	X						
12	Synchronized communication	X	X	X					
13	Action overlay	X	X	X					
14	Change of stage	X	X	X	X				
15	Stylized sound change of stage and actors	X	X	X					
16	Random numbers	X	X	X	X	X			
17	Variables	X	X	X	X	X	X		
18	Action and message interaction	X	X	X	X	X	X		
19	Customizing the course and using multiple variables	X	X	X	X	X	X		
20	Direct interaction with the user	X	X	X	X	X	X	X	
21	Assessment	X	X	X	X	X	X	X	X

References

- Heiter, F., Manóvil, L., Furey, T.: A review of models for introducing computational thinking, computer science and computing in K-12 education. In: Frontiers in Education Conference (FIE), pp. 1-9, IEE, LN (2016)
- Ross, S., Halpern, J., Kay, T.: An exploration of the role of visual programming tools in the development of young children's computational thinking. *Educ. J. e-Learning* 18, 297-309 (2017)
- Premsky, M.: Digital natives, digital immigrants. *Horizon* 9(5), 1-6 (2001)
- European Commission: Skills for Jobs in Europe: measuring progress and moving ahead. https://ec.europa.eu/eurostat/tgm/table.do?tab=table&init=1&language=en&code=sdg_4_5_10, Accessed 12 Dec 2017
- Johnson, L., Becker, S., Estrada, V., Freeman, A.: NMC Horizon Report: 2014 K-12 Edition. Austin, Texas: The New Media Consortium, <http://www.nmc.org/2014/07/14/2014-horizon-report-k-12/>, Accessed 7 Feb 2018
- Piper, S.: Mindstorms: Children, Computers, and Powerful Ideas. Basic Books, New York (1993)
- Wing, J.: Computational Thinking. *Commun. ACM* 49(3), 33-36 (2006)
- Gross, S., Paik, R.: Computational thinking in K-12: review of the state of the field. *Educ. Res.* 42(1), 38-43 (2013)
- Karamba, B.: Thinking from a long winter in computer science education. *Forbes*, pp. 1-2 (2019)
- Ronick, M., Makony, J., Moore Hernandez, A., Busk, N., Eastwood, E., Brennan, K., Miller, A., Buchanan, E., Silver, J., Silverman, B., Katz, Y.: Scratch programming for all. *Commun. ACM* 52(11), 86-87 (2009)
- Graczyk, A.: ALICE as a tool for programming at schools. *Natural Sci.* 2(2), 124-129 (2010)
- Lyse, S.Y., Koh, J.H.L.: Review on teaching and learning of computational thinking through programming. What is next for K-12? *Comput. Hum. Behav.* 41, 53-61 (2014)
- Bern, M.L., Flannery, L., Karzoff, E.R., Sullivan, A.: Computational thinking and tinkering: exploration of an early childhood robotics curriculum. *Comput. Educ.* 72, 145-157 (2014)
- Valente, J.A.: Integração do Pensamento Computacional no Currículo da Educação Básica: Diferentes Estratégias Usadas e Questões de Formação de Professores e Avaliação do Aluno. *Revista e-Curriculum*, vol. 14(3), <https://revista.unesp.br/doi/pdf/10.11606/1807-0191/revista/29651>, Accessed 12 Dec 2017
- Ramos, L.L., Espalino, R.G.: Iniciação à Programação no 1.º Ciclo do Ensino Básico. Estudos de Avaliação do Projeto. DGE - Direção Geral de Educação, Portugal (2016)
- Mays, L., Pearson, J.N., Tapscott, T., Wheeler, Q.M., Reese, G.: Supporting all learners in school-wide computational thinking: a cross-case qualitative analysis. *Comput. Educ.* 82, 263-279 (2015)
- Campos, G., Cavaleiro, S., Foss, L., Peres, A., Pinto, C.F., Aguiar, M., Du Bois, A., Ribeiro, R.: Organização de informações via pensamento computacional: relato de atividade aplicada ao ensino fundamental. V Congresso Brasileiro de Informática na Educação (CBIE), 30th Workshop de Informática na Escola (WIFE), UFPA, Brazil (2014)
- Román, M.: Test de Pensamento Computacional: princípios de diseño, validación de contenido y análisis de ítems. In: María Menoyo, M.A. (eds.) *Prospección y Avances de la Investigación*, pp. 279-302. (NB3), Madrid (2015)
- Fabre, H., Winkens, M., Doornbos, R., van der Ven, J.S., de Vries, K.: Teaching computational thinking to primary school students via unplugged programming lessons. *J. Europ. Teach. Educ.* 26, 12-24 (2017)
- Espalino, R.G.: Iniciação à Programação no 1.º ciclo do ensino básico. *Livros orientados para professores*. Accessed 25 Jan 2017
- Sáez-López, J.M., Sevilla-García, M.L.: Sensors, programming and devices in 2nd Education sessions. One case in the context of primary education. *Cultura y Educación* 29 (2), 138-154 (2017)
- Calicut, D.: Teaching programming: some lessons from probing. In: Proceedings of the IETN-NC in Annual Conference, Houton, Wain, Edinburgh, UK (2009)
- Bouchette, R.M.: Aplicações do pensamento computacional e desenvolvimento do raciocínio. Tese de doutoramento, Universidade Federal do Rio Grande do Sul (2017)
- Buchanan, C.P.: Desenvolvimento do pensamento computacional através de atividades desafiadoras na educação básica. Tese de doutoramento, Universidade Federal do Rio Grande do Sul (2017)
- Joel, G., López, E., Branson, P.: Sources of self-efficacy in investigation of elementary school students in France. *J. Educ. Psychol.* 103(3), 649-663 (2011)

Authors:
António Marques 1, Carina Guimarães 1, and Ana Salgado 2
Universidade da Beira Interior 1, Escola Superior de Saúde do Porto 2
Portugal 1, Portugal 2

145

Anexo E: Artigo Scratch 3 – Beginners Programming Course in 3rd Year of Primary School

Scratch 3 – Beginners Programming Course in 3rd Year of Primary School

António Marques¹, Carina Guimarães¹, Ana Salgado²

¹ Universidade da Beira Interior, Estrada do Sineiro, 6200-209, Covilhã, Portugal

² Escola Superior de Saúde do Porto – P.Porto, Rua Dr. António Bernardino de

Almeida, 400, 4200-072, Porto, Portugal

antonio.marques@ubi.pt

scmg@ubi.pt

ais@ess.ipp.pt

Abstract. In the 2015/2016 academic year, Portugal's General Directorate of Education launched the pilot project IP1 – Beginners Programming in Primary School, addressing the goal that students learn to program and with this experience develop cognitive and social skills.

Programming is becoming popular in different levels of school in many countries due to its numerous advantages such as the development of competences for problem solving, divergent ways of thinking, creativity, communication and group work.

The work presented here is an initial work, implemented within the frame of a doctoral degree project that describes a program of 21 fifty-minute sessions', using Scratch tools, designed to develop the computational thinking of students in the third year of primary school. This paper also describes the instruments that will be built to evaluate the program's effectiveness.

This program can be a pedagogical reference for implementation of project IP1 in other primary schools. Final considerations will be raised about future steps.

Keywords: Computational Thinking, Primary School, Intervention Programs.

1 Introduction

Nowadays, any service or product that we consume interrelates with, whether directly or indirectly, some process or decision determined by a computer chip. Almost everyone, irrespective of their age, needs some basic technological knowledge. Some authors [1, 2] maintain that computational thinking is literacy for the 21st century. With this technology so very present, a new learning profile is appearing, the so called “digital native”, a person who is accustomed to working on various tasks simultaneously, with easy and fast access to information, interconnected via networks and based on immediate and frequent gratifications [3]. Since a very high percentage of professions require digital competences [4] people should not act only as consumers, they also need to be creators. Countless education reports point in this direction [5]. The contribution of technology to educational context is not only necessary but also an unmistakable reality for our schools.

Despite Papert [6] having introduced a programming language for educational purposes and transferable to other spheres of life in 1980, it was only in 2006 that Jeannette Wing popularize the term “*computational thinking*” and set out the necessary theoretical framework. This concept “involves solving problems, designing systems, and understanding human behaviour, by drawing on the concepts fundamental to computer science” [7].

Multiple definitions and updates were to follow in search of a consensus that is still lacking in the field [8]. One the most consensual definition of CT – computational thinking, include the “creative, critical and strategic human capacity to know how to use the fundamentals of computing in the most diverse areas of knowledge, with the purpose of identifying and resolving problems, whether individually or collaboratively, through clear steps, so that either a person or a machine may execute them efficiently” [9].

For the majority of people, programming and coding are perceived as complex actions based on what are considered to be advanced levels of technological knowledge and thus accessible only those with certain abilities. However, a number of authors defend an opposing idea. Papert [6], for example, advocates that programming languages should have different levels, from the simple to the complex, and thus be accessible to all.

The increasing number of programming languages, such as *Scratch* [10] or *Alice* [11], free and user friendly, easier to grasp and begin using, raises the interest of programming among pre university level. In recent years, primary and secondary schools, mainly all in Anglo-Saxon countries, have embarked on initiatives striving to introduce CT. Worldwide, school curricula have been updated in this field to include students at younger ages via an enormous variety of educational interventions [12, 13]. Valente [14] systematized these as follows: a) activities without resource of technologies; b) programming in Scratch; c) pedagogical robotics; d) producing digital narratives; e) designing games and f) using of simulators. However, even though these technologies are diverse and susceptible to multiple applications, these interventions frequently emerge as sporadic or they only target an elite group or they lack overall intentionality.

In Portugal the most recent initiatives launched by the Ministry of Education and Science in this field include the pilot project “*Beginners Programming in Primary School* and the inventory and support for *Programming and Robotics Clubs* existing nationwide” [15]. However, the literature states that there are only occasional studies on such interventions in K12 school contexts (pre-university) [8] and that these studies are especially rare in primary school [16].

In Brazil, Campos and his colleagues [17] carried out one of the first attempts to measure improvements to CT, through pre- and post-testing. The results did not return any statistically significant improvement but the activities were described as gratifying. In Spain, Roman [18] developed a CT test with 28 items with four response choices for a 45 minutes test for school children aged between 12 and 13. Faber and his colleagues [19] developed a six-session program in 26 primary school establishments in the Netherlands and received positive feedback from teachers and students but there was no metric for participants’ levels of satisfaction or statistical data on changes in CT performance standards.

Specifically in Portugal, there have been diverse studies carried out to evaluate the IP1 program in order to boost the effectiveness of future proposals [15]. The conclusions convey a very positive general evaluation by school headmasters, teachers and school initiative coordinators with regard to the level of attainment of the project goals and also highlighted the importance of the guidelines drawn up, which helped reinforce the continuity of the program. However, Ramos and Espadeiro [15] remind us of the need to evaluate the impact of these programs in terms of the changes in student competences and knowledge.

In accordance with this framework, this study seeks to describe a program fostering CT and to analyse its effectiveness.

2 Program_Scratch 3

The program presented here stems from the pilot project “Beginners Programming in Primary School” [20], run by the GDE - General Directorate of Education, as an intervention proposal for a field that is integrated into the education program of a number of schools through recourse to the Scratch programming application (3.rd and 4.rd years). The literature details various ways of applying Scratch to different levels of teaching [21] and, in terms of the challenges set by the pilot project, each teacher and school established their own interventions. What occurred in practice was that in many schools, the program featured some random activities, detached from the theoretical framework that would have endowed them with meaning or coherence. Another issue was that there were also many teachers engaged who lacked any sort of guide for orienting their work, and consequently were unable to evaluate the effectiveness of their interventions.

In order to avoid these problems, a group of five computer teachers (one is author of this paper), who are developing the pilot project in 3.rd year of primary school, designed originally this program which intends to guide the work of teaching programming in partnership with teachers from other subject areas (see Table 1.). The program to foster computational thinking among participants students follow literature review, guidelines handed down by the Ministry of Education (ME) for the program and the curricular targets to be attained by third year students. Specific goals (e.g. activities planning, symbolic representation) and soft skills (e.g. group work, communication) proposed by ME are also integrated. These 21 sessions, 50 minutes each, through use of the Scratch tools (e.g. characters, stages, scenarios, movements, sound, sensors, etc.) allow students to develop fundamental concepts in the field of programming such as sequences, parallel operations, events, decision conditions/structures, cycles, operators, data/variables, and simple functions, among others.

According to Callear [22] when teaching programming is relevant to: a) slow pace teaching to facilitate the concepts integration; b) teach one by one topic to avoid confusions; c) review previous topics and d) sort the subjects by degree of difficulty. These principles guided this specific program.

The training is planned around exercises with increasing levels of difficulty (most of the times with the introduction of new blocks of code, which are explored at the beginning of each session) and in their majority, ended with options that students have

to analyse, often by breaking down the problem into smaller sections in order to arrive their best decision. This allows students to consolidate the knowledge that they have thus far acquired and to develop problem-solving strategies.

3 Evaluation of program effectiveness

Several instruments will be use to evaluate the effectiveness of the program.

The *Computational Thinking Test*, originally developed by Roman [18] for application in Spain involves measuring the level of CT development with a focus on the fundamental concepts of programming such as sequences and repetitions. The adaptation and validation of the Portuguese version will be done within the scope of this work (with both experts and prior pilot studies). In Brazil, there have also been psychometric studies applying this test [23, 24]. The test contains 28 exercises: four questions about simple sequences; four questions about the cycle of repetition, by the number of times; four questions about the cycle and repetition through to a determined condition; four questions about simple conditions, (condition – action); four questions about conditions composed (condition – action – exception); four questions about the cycle of repetition as a condition for verification; and four questions about simple functions so as to divide a problem by grouping tasks. The level of complexity and difficulty of the exercises rises throughout the test and always presents four response options (A, B, C and D), of which only one is correct. The test lasts for 45 minutes and begins with three practice exercises. Carrying out this test also explores the capacity to formulate and resolve problems either directly or in phased approaches.

Self-efficacy in Computational Thinking is measured through a specific scale developed by the authors in accordance with the literature [e.g. 25]. The scale evaluates participant perceptions in regard to their competences in the computer science field (e.g. “*I know how to symbolically represent sequences of actions for activities*”) with responses made according to a five-point Likert scale (1 – totally disagree to 5 – totally agree). Validation for the portuguese population shall comply with all the requirements for its appropriate application.

In addition, *sociodemographic data* (e.g. *academic qualifications of parents*) as well as a *technology user information* (e.g. “*Do you use a computer daily?*”) were also collected. The data gathering process also extends to the participants’ academic records in three subjects: Portuguese, Mathematics and Environmental Studies.

We would further add that evaluating the program’s efficiency spans two phases: pre (M1) and post (M2) application into two groups of students.

Table 1. Program General Plan

Session	Activity	Programming concepts								
		Sequences	User passive interaction	Repetition	Conditional statements	Communication and	Boolean logical	Random	Variables	User direct interaction
1	Scratch Presentation	X	X							
2	Customize actors	X	X	X						
3	Insert stages	X	X							
4	Conditional statements	X	X		X					
5	Messages and sounds	X	X	X	X	X				
6	Effects	X	X	X						
7	Sensors	X	X	X	X	X				
8	Sensors II	X	X	X	X	X				
9	Customize stages	X	X	X	X	X				
10	Visibility (yes/no)	X	X				X			
11	Customize stages and actors	X	X							
12	Synchronized communication	X	X			X				
13	Actions overlap	X	X	X		X				
14	Change of stages	X	X	X	X	X				
15	Synchronized change of stages and actors	X	X			X				
16	Random numbers	X	X	X	X	X	X	X		
17	Variables	X	X	X	X	X	X	X	X	
18	Actors and stages interaction	X	X	X	X	X	X	X	X	
19	Customizing the cursor and using multiple variables	X	X	X		X		X	X	
20	Direct interaction with the user	X	X	X	X	X	X	X	X	X
21	Assessment	X	X	X	X	X	X	X	X	X

4 Final Considerations

Computational thinking in education is, to some extent, an inevitable trend. There are various studies demonstrating how students with training and education in programming are better prepared for future daily professional tasks and challenges [10]. Based on their study findings, Sáez-López and Sevillano-García [21] recommend that those responsible for education policies integrate them into primary school teaching programs and into subject areas of art, music and technology.

However, such practices are not always reflective and intentional, and very often the teaching of programming languages teaching only a very reduced number of students or only happens on an occasional basis. The IP1, proposed by the ME, is an opportunity for schools to develop structured CT programs. However, according to literature [e.g. 16], is also important to assess these programs, in areas like computational thinking skills and their impact on learning. important evaluate this programs. Therefore, this paper is only one first step, within the frame of a doctoral degree project doctoral, presenting one program and their process of evaluation. In the near future authors will be obtained the authorizations from the respective School Board and parents of the children participated (3.rd and 4.rd years of primary school) in two studies: evaluation of program and psychometric study of instruments. After gathering all data, statistical analysis of the instruments and the program will be carried out according to the goals defined. The results and discussion will continue to be just a small contribution towards building knowledge in this area. More research is clearly required to establish how to teach children use programming tools and how they influence their development (e.g. problem-solving and computational thinking abilities).

References

1. Heintz, F., Mannila, L., Farnqvist, T.: A Review of Models for Introducing Computational Thinking, Computer Science and Computing in K–12 Education. In: *Frontiers in Education Conference (FIE)*, pp. 1–9, Erie, US (2016).
2. Rose S., Habgood J., Jay, T. : An Exploration of the Role of Visual Programming Tools in the Development of Young Children’s Computational Thinking. *The Electronic Journal of e-Learning* 15, 297-309 (2017).
3. Prensky, M.: Digital natives, digital immigrants. *On the Horizon* 9(5), 1-6 (2001).
4. European Commission. *E-Skills for Jobs in Europe: measuring progress and moving ahead*, <http://eskills-monitor2013.eu/results/>, last accessed 2017/12/12.
5. Johnson, L., Becker, S., Estrada, V., Freeman, A.: *NMC Horizon Report: 2014 K-12 Edition*. Austin, Texas: The New Media Consortium, <https://www.learntechlib.org/p/147472/>, last accessed 2018/02/7.
6. Papert, S.: *Mindstorms: Children, computers, and powerful ideas*. Basic Books, New York (1980).
7. Wing, J.: Computational thinking. *Communications of the ACM* 49(3), 33–36 (2006).
8. Grover, S., Pea, R.: Computational Thinking in K-12: A Review of the State of the Field. *Educational Researcher* 42(1), 38–43 (2013)
9. Kurshan, B.: Thawing from a Long Winter in Computer Science Education. *Forbes*, pp. 1-2, (2016).
10. Resnick, M., Maloney, J., Monroy-Hernandez, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., Kafai, Y.: *Scratch: Programming for All*. *Communications of ACM* 52(11), 60–67 (2009).

11. Graczyńska, E.: ALICE as a tool for programming at schools. *Natural Science* 2(2), 124-129 (2010).
12. Lye, S.Y., Koh, J.H.L.: Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior* 41(0), 51 – 61(2014).
13. Bers, M. I., Flannery, L., Kazakoff, E. R., Sullivan, A.: Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education* 72, 145–157 (2014).
14. Valente, J.: A. Integração do Pensamento Computacional no Currículo da Educação Básica: Diferentes Estratégias Usadas e Questões de Formação de Professores e Avaliação do Aluno. *Revista e-Curriculum*, vol. 14 (3), <https://revistas.pucsp.br/index.php/curriculum/article/view/29051>, last accessed 2017/12/12.
15. Ramos, J. L., Espadeiro, R.G.: Iniciação à Programação no 1.º Ciclo do Ensino Básico. Estudos de Avaliação do projeto-piloto. DGE – Direção Geral de Educação, Portugal (2016).
16. Maya, I., Pearson, J. N., Tapia, T., Wherfel, Q. M., Reese, G.: Supporting all learners in school-wide computational thinking: A cross-case qualitative analysis. *Computers & Education* 82, 263– 279 (2015).
17. Campos, G., Cavalheiro, S., Foss, L., Pernas, A., Piana, C.F., Aguiar, M., Du Bois, A., Reiser, R.: Organização de informações via pensamento computacional: relato de atividade aplicada no ensino fundamental. 3.º Congresso Brasileiro de Informática na Educação (CBIE). 20.º Workshop de informática na escola (WIE), UFGD, Brasil (2014).
18. Román, M.: Test de Pensamiento Computacional: principios de diseño, validación de contenido y análisis de ítems. In Murga Menoyo, M. A. *Perspectivas y Avances de la investigación*, pp. 279-302. UNED, Madrid (2015).
19. Faber, H., Wierdsma, M., Doornbos, R., Ven, J. S., Van Der; Vette, K.: Teaching Computational Thinking to Primary School Students via Unplugged Programming Lessons. *Journal of the European Teacher Education Network* 12, 13-24 (2017).
20. Educação, D.G. Iniciação à programação no 1. ciclo do ensino básico. Linhas orientadoras gerais, http://www.erte.dge.mec.pt/sites/default/files/Projetos/Programacao/IPICEB/linhas_orientadoras.pdf, last accessed 2017/01/24.
21. Sáez-López, J.M., Sevillano-García, M.L.: Sensors, programming and devices in Art Education sessions. One case in the context of primary education. *Cultura y Educación* 29(2), 350-384 (2017).
22. Callear, D.: Teaching Programming: Some Lessons from Prolog. In *The Proceedings of the LTSN-ICS 1st Annual Conference*, Heriot Watt, Edinburg, UK (2000).
23. Boucinha, R. M.: Aprendizagem do pensamento computacional e desenvolvimento do raciocínio. Tese de doutoramento, Universidade Federal do Rio Grande do Sul (2017).
24. Brackmann, C. P.: Desenvolvimento do pensamento computacional através de atividades desplugadas na educação básica. Tese de doutoramento, Universidade Federal do Rio Grande do Sul (2017).
25. Joet, G., Usher E., Bressoux, P.: Sources of Self-Efficacy: An Investigation of Elementary School Students in France. *Journal of Educational Psychology*, Vol. 103 (3), 649–663 (2011).

Anexo F: Programa de Intervenção

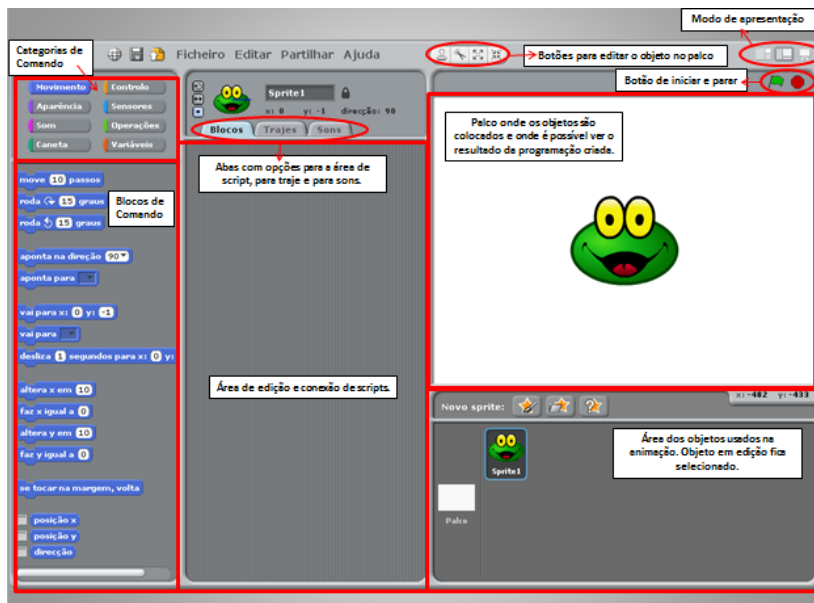


1.º Ciclo

Iniciação à Programação

Exercício 1 – Apresentação do Scratch

Objetivo: Conceitos gerais, exploração do ambiente de trabalho e programação da alteração de trajes.

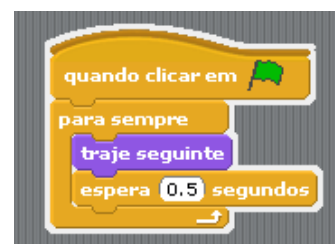


Passos a realizar:

1. Abra o Scratch e apague o gato que aparece no início.
2. De seguida personalize o fundo (palco) ao seu gosto.
3. Vá à biblioteca de atores e selecione o morcego.
4. Importe os trajes como mostra a seguinte figura:



5. Programe agora o morcego com o seguinte código:
6. Caso pretenda faça o mesmo para outras personagens.



Bom trabalho!!



1.º Ciclo

Iniciação à Programação

Exercício 2 – Letras animadas

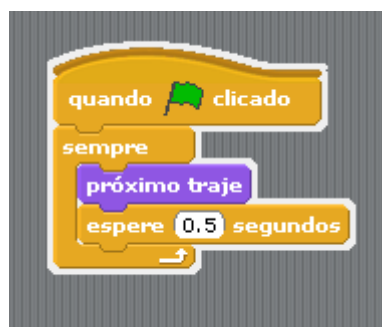
Objetivo: Inserir diferentes letras criando novos trajes (diferentes cores). Programação da alteração de trajes.

Passos a realizar:

1. Abra o Scratch.
2. De seguida vá à biblioteca de atores e selecione um novo *sprite* da **Biblioteca, Letras, OutLine.**



3. Selecione a primeira letra do seu nome.
4. Vá agora aos **trajes**, selecione a letra escolhida e duplique-a várias vezes (botão Copiar). Pinte-a de outras cores (botão Editar).
5. Programe a letra com o seguinte código:



6. Repita todo o processo para as restantes letras do seu nome.
7. Personalize o palco a seu gosto.
8. Experimente agora com a palavra **DIVERTIDO**.

Bom trabalho!!



1.º Ciclo

Iniciação à Programação

Exercício 3 – Caminhar no deserto

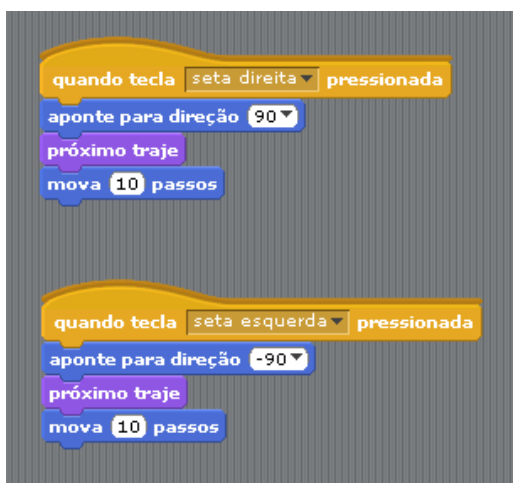
Objetivo: Personalização do Palco, inserir personagem com trajes, programar alteração de trajes e movimento.

Passos a realizar:

1. Abra o Scratch e apague o gato que aparece no início.
2. De seguida personalize o fundo da tela importando da livreria **Nature, desert**.
3. De seguida vá à biblioteca de atores e selecione um novo *sprite* da **Biblioteca, People**.



4. Vá agora aos **trajes** e importe a segunda posição da personagem.
5. Programe a personagem com o seguinte código:



6. Altere o tipo de rotação da personagem.
7. Se quiser, faça o mesmo processo com outra personagem.



Bom trabalho!!



1.º Ciclo

Iniciação à Programação

Exercício 4 – Aquário

Objetivo: Personalização do Palco, inserir personagem com trajes, programar alteração de trajes e movimento.

Passos a realizar:

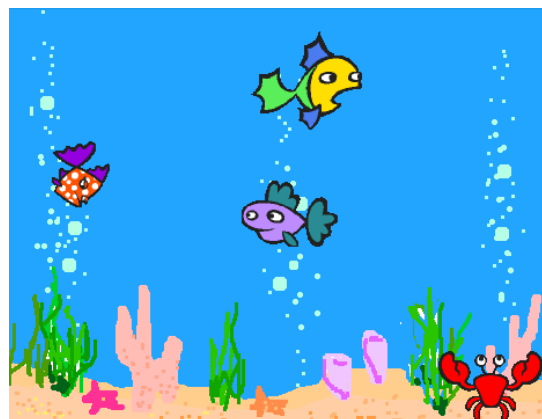
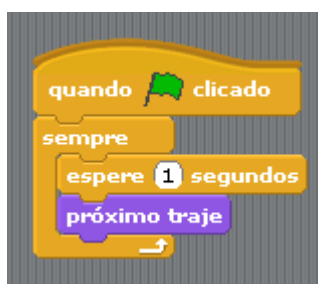
1. Abra o Scratch e apague o gato que aparece no início.

2. Insira os seguintes atores:

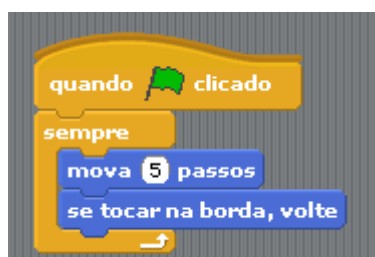


3. Coloque o seguinte palco e coloque os atores como mostra a figura:

4. Programe o caranguejo com o seguinte código:



5. Programe cada peixe com o seguinte código:



6. Faça as alterações necessárias ao peixe laranja para ele funcionar corretamente.

7. Insira novos personagens a seu gosto.

Bom trabalho!!



1.º Ciclo

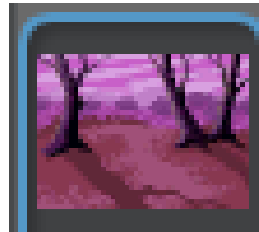
Iniciação à Programação

Exercício 5 – A bruxa

Objetivo: Introdução de pensamentos/falas e efeitos de som e cores.

Passos a realizar:

1. Abra o Scratch e apague o gato que aparece no início.
2. Coloque o seguinte fundo/palco.
3. Insira os seguintes atores:

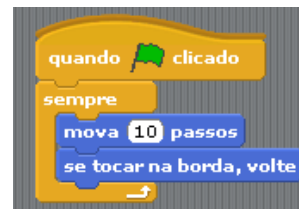


4. Coloque os trajes no morcego.
5. Programe com os seguintes códigos:

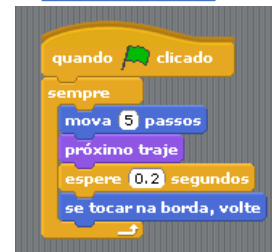
Senhora



Bruxa



Morcego



6. Selecione o palco e importe um som para a animação.



7. Programe o palco com o seguinte código.



Bom trabalho!!



1.º Ciclo

Iniciação à Programação

Exercício 6 – Dança

Objetivo: Introdução de pensamentos/falas e efeitos de som e cores.

Passos a realizar:

1. Abra o Scratch e apague o gato que aparece no início.
2. Coloque o seguinte palco e os seguintes atores:



3. Programe a apresentadora da seguinte forma:

```

quando clicado
diga Olá!! Bem vindos por 2 segundos
diga Querem ver este senhor a dançar? por 2 segundos
diga Então carreguem na tecla SPACE para começar por 3 segundos
  
```

4. Selecione o ator “dançarino” e importe os vários trajes dessa personagem.
5. Programe o dançarino com o seguinte código:

```

quando tecla space pressionada
repita 100
  toque o tambor 56 por 0.1
  próximo traje
mude para o traje dan5
  
```

6. Selecione o palco e programe-o da seguinte forma:

```

quando tecla space pressionada
repita 400
  mude o efeito cor por 25
  
```

Bom trabalho!!



1.º Ciclo

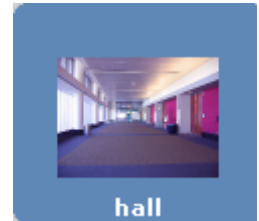
Iniciação à Programação

Exercício 7 – Formas geométricas

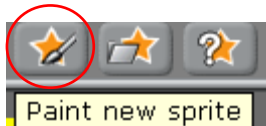
Objetivo: Criação de atores e programação da interação entre os diferentes atores.

Passos a realizar:

1. Abra o Scratch e apague o gato que aparece no início.
2. Personalize o fundo da tela importando da livraria **Indoor, hall**.



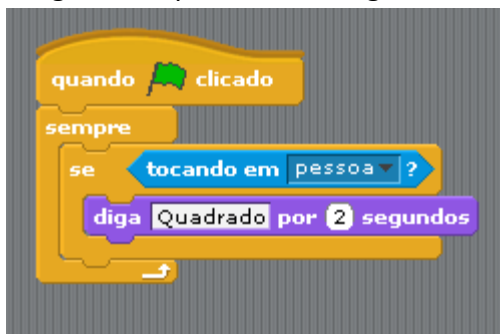
3. Crie 3 atores (Triângulo; Quadrado; Retângulo)



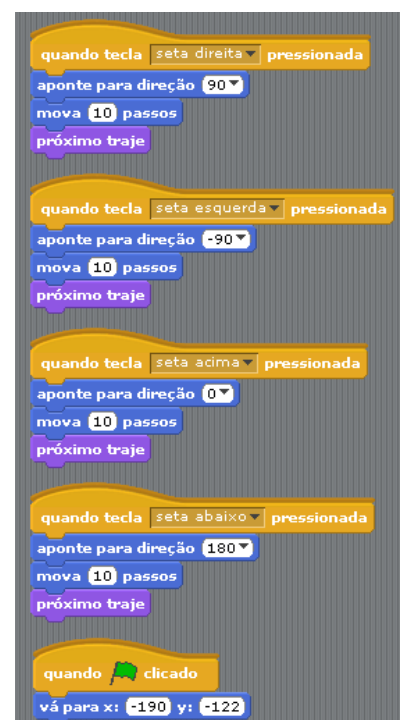
4. Vá à biblioteca de atores e selecione um novo *sprite* da **Biblioteca, People**.



5. Vá agora aos **trajes** e importe as restantes posições do ator.
6. Programe o quadrado da seguinte forma:



7. Faça o mesmo processo com o triângulo e o retângulo.
8. Programe o rapaz da seguinte forma:
9. Altere o tipo de rotação da personagem.



Bom trabalho!!



1.º Ciclo

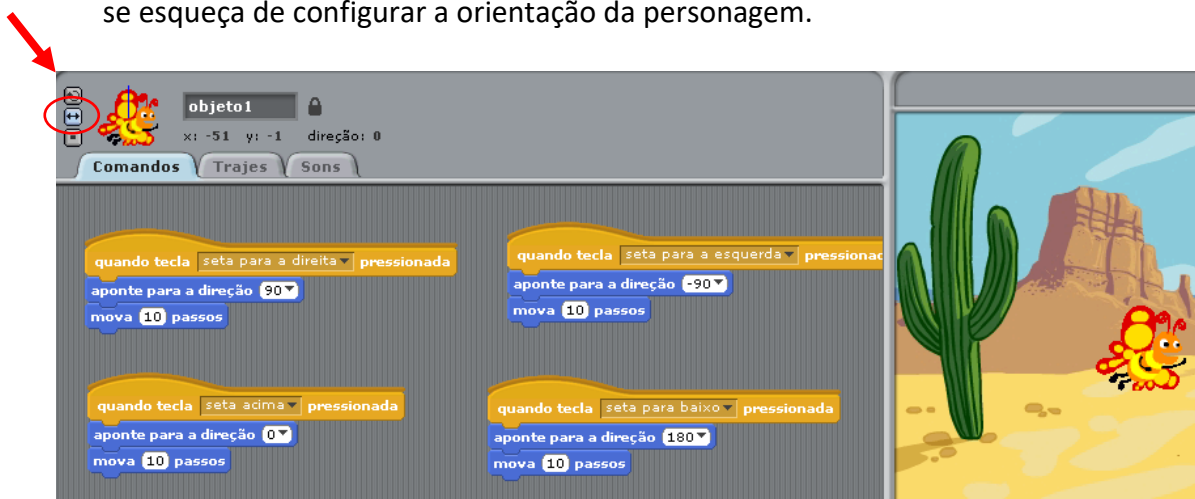
Iniciação à Programação

Exercício 8 – Cores em Inglês

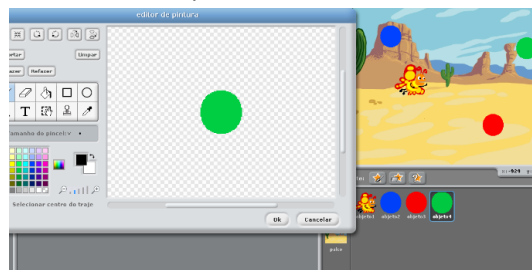
Objetivo: Criação de atores e programação da interação entre os diferentes atores.

Passos a realizar:

1. Abra o Scratch e apague o gato que aparece no início.
2. Personalize o fundo da tela e insira uma personagem, por exemplo uma borboleta.
3. Programe a borboleta para que ela se movimente com o auxílio das setas. Não se esqueça de configurar a orientação da personagem.



4. Desenhe mais 3 personagens (círculos de diferentes cores).



5. De seguida, programe a borboleta para quando tocar nos círculos apareça um “balão” com a designação da respetiva cor em inglês, durante 1 segundo e volte à posição inicial.

Bom trabalho!!



1.º Ciclo

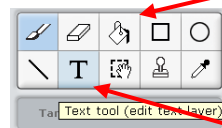
Iniciação à Programação

Exercício 9 – Animais

Objetivo: Personalização do palco e programação de diferentes sons e comportamentos para os diferentes atores.

Passos a realizar:

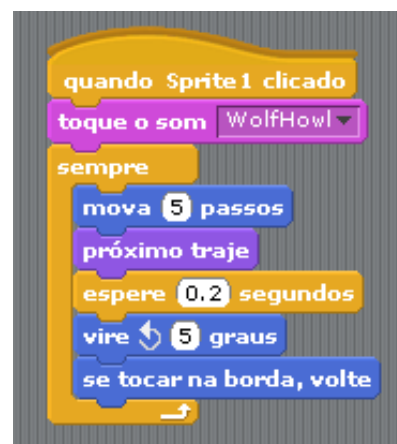
1. Abra o Scratch e apague o gato que aparece no início.
2. Mude o nome do fundo da tela1 para Início e carregue no botão **editar** para escrever o texto “Selecione um animal” e colocar uma cor de fundo (o tipo de letra e cor de fundo é à sua escolha).



Alterar cor de

Inserir Texto

3. Coloque as personagens morcego, cão e elefante nesse fundo.
4. Selecione o morcego e importe o som WolfHowl.
5. Importe os 2 trajes do morcego.
6. Programe o morcego com o seguinte código:
7. Programe os outros animais da forma que quiser. Não se esqueça dos sons.



Bom trabalho!!



1.º Ciclo

Iniciação à Programação

Exercício 10 – Carros

Objetivo: Personalização do palco e atores com diferentes formas/estilos. Programação de atores com diferentes comportamentos (andar, aparecer, deslizar, desaparecer).

Passos a realizar:

1. Abra o Scratch e apague o gato que aparece no início.
2. Crie um palco parecido ao seguinte (pode melhorá-lo) e adicione os atores carros (1.º o carro verde depois o carro azul):



3. Programe os carros de forma a deslizar até desaparecer.

Carro Verde	Carro Azul
<pre> quando clicado apareça vá para x: 134 y: -89 deslize em 1 segundos para x: -234 y: -97 desapareça </pre>	<pre> quando clicado apareça vá para x: -162 y: -113 deslize em 1 segundos para x: 278 y: -114 desapareça </pre>

4. Pode adicionar e programar novos atores como um avião ou um pássaro por exemplo.

Bom trabalho!!



1.º Ciclo

Iniciação à Programação

Exercício 11 – O meu carro

Objetivo: Personalização do palco e atores com diferentes formas/estilos. Programação de atores com diferentes comportamentos (andar, aparecer, deslizar, desaparecer).

Passos a realizar:

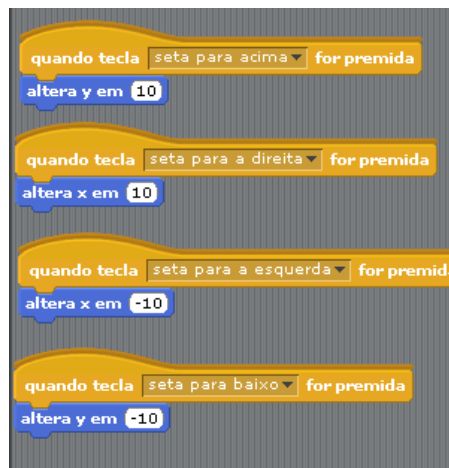
1. Abra o Scratch e apague o gato que aparece no início.
2. Crie um palco parecido ao seguinte (pode melhorá-lo):



3. Vá à biblioteca de atores e selecione um novo *sprite* da **Biblioteca**, **transportes** e personalize-o.



4. Programe o carro da seguinte forma:



Bom trabalho!!



1.º Ciclo

Iniciação à Programação

Exercício 12 – Jogo de basquetebol

Objetivo: Programação de diálogos devidamente coordenados no tempo.

Passos a realizar:

1. Abra o Scratch e apague o gato que aparece no início.
2. Importe o seguinte palco e personagens.
3. Personalize as personagens (mude a cor do cabelo, da roupa e dos ténis da menina da direita e troque a direção da menina da esquerda).



4. Programe as meninas com o seguinte código:

```

quando clicado
diga Já estou cansada de jogar, vamos parar? por 2 segundos
anuncie primeira msg e espere

quando receber segunda msg
diga Então jogamos mais dez minutos. por 2 segundos
  
```

Menina da

```

quando receber primeira msg
diga Eu estou ótima para continuar, vamos jogar mais por 2 segundos
anuncie segunda msg
  
```

Menina da

5. Pode continuar o diálogo das duas meninas como quiser.

Bom trabalho!!



1.º Ciclo

Iniciação à Programação

Exercício 13 – Pista de Dança

Objetivo: Programação de diálogos devidamente coordenados no tempo.

Passos a realizar:

1. Abra o Scratch e apague o gato que aparece no início.
2. Importe o seguinte palco e personagens.



3. Selecione o ator “dançarino” e importe os vários trajes desse personagem.
4. Programe as personagens com o seguinte código:

```

quando clicado
diga Olá!! Bem vindos por 2 segundos
diga Querem ver este senhor a dançar? por 2 segundos
diga Então carreguem na tecla SPACE para começar por 3 segundos
  
```

Apresentadora

```

quando tecla space pressionada
repita 400
  muda o efeito cor por 25
  muda para o traje dan1
  
```

Dançarino

5. Selecione o palco e programe-o da seguinte forma:
6. Caso pretenda acrescentar dançarinos.

Bom trabalho!!



1.º Ciclo

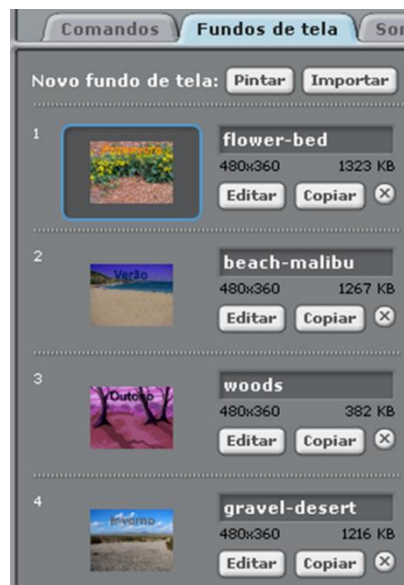
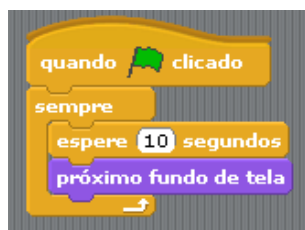
Iniciação à Programação

Exercício 14 – Estações do ano

Objetivo: Programação do comportamento dos atores com alteração de cenários ao longo da animação.

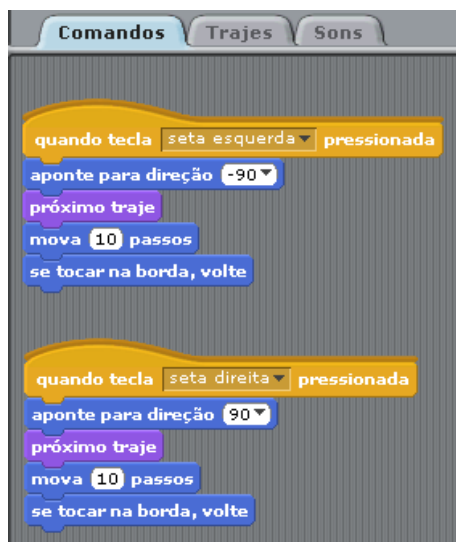
Passos a realizar:

1. Abra o Scratch e apague o gato que aparece no início.
2. Importe os seguintes fundos de tela.
3. Personalize cada fundo escrevendo o nome da estação do ano correspondente.
4. Programe o palco da seguinte forma:



5. Vá à biblioteca de atores e selecione um novo sprite da **Biblioteca, pessoas** e introduza todos os seus trajés.

6. Programe a personagem com o seguinte código:



Bom trabalho!!



1.º Ciclo

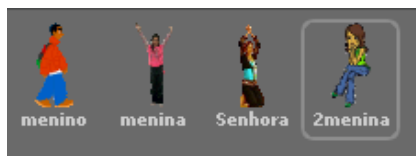
Iniciação à Programação

Exercício 15 – Descobrir o mundo

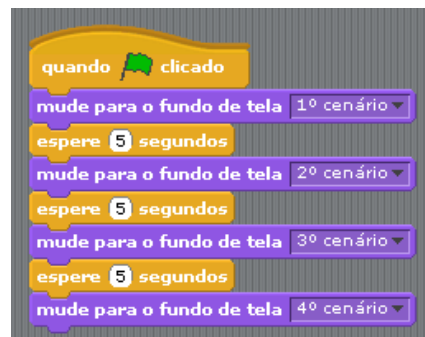
Objetivo: Programação do comportamento dos atores com alteração de cenários ao longo da animação.

Passos a realizar:

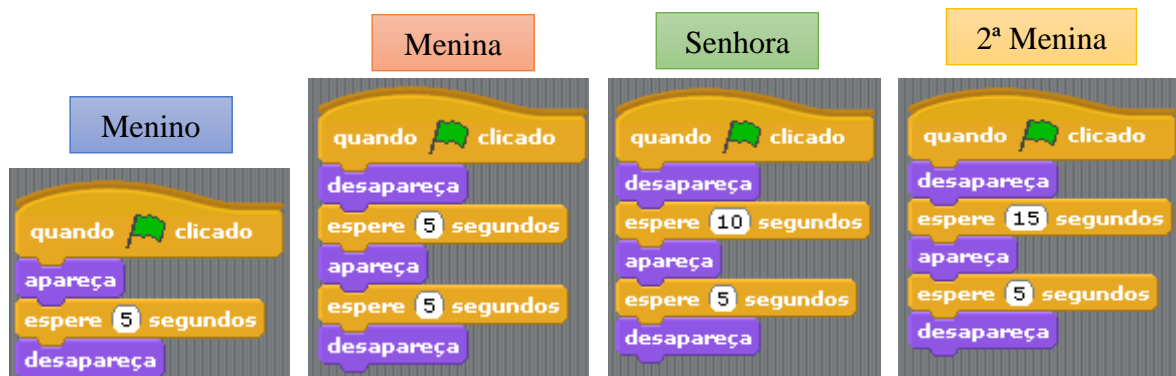
1. Abra o Scratch e apague o gato que aparece no início.
2. Importe os seguintes fundos de tela e altere o nome de cada um deles.
3. Insira os seguintes atores e mude-lhes o nome:



4. Programe o palco da seguinte forma:



5. Programe também as personagens com o seguinte código:



6. Posicione os atores a seu gosto.

Bom trabalho!!



1.º Ciclo

Iniciação à Programação

Exercício 16 – Apanhar borboletas

Objetivo: Conceito de número aleatório conjugado com a programação do movimento e a interação entre atores (ação/reação).

Passos a realizar:

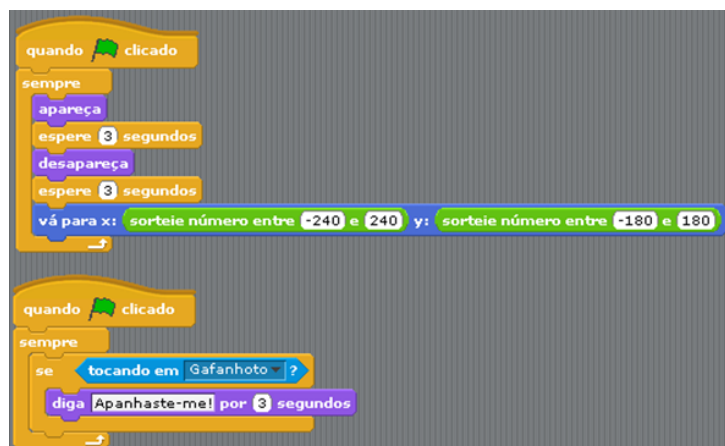
1. Abra o Scratch e apague o gato que aparece no início.
2. De seguida importe o seguinte fundo de tela e atores como mostra a figura.



3. Programe o Gafanhoto para se deslocar com as setas do teclado:



4. Programe a Borboleta para se mostrar e esconder em locais diferentes da tela e quando for apanhada pelo Gafanhoto diz “Apanhaste-me!”.



Bom trabalho!!



1.º Ciclo

Iniciação à Programação

Exercício 17 – Apanhar ratos

Objetivo: Introdução de variáveis associadas a um comportamento.

Passos a realizar:

1. Abra o Scratch e apague o gato que aparece no início.
2. Importe os seguintes atores:



Nota: Deverá editar os ratos para lhe alterar a cor.

3. Crie a variável “Pontos”.
4. Coloque o seguinte palco e coloque os atores como mostra a figura:
5. Programe os atores da seguinte forma:



Em todos os ratos

Gato

```

quando clicado
sempre
  vá para x: sorteie número entre -200 e 200 y: sorteie número entre -150 e 150
  espere 3 segundos

quando clicado
apareça
sempre
  se tocando em Gato?
  desapareça
  mude Pontos por 1
  espere 10 segundos
  apareça
  
```

```

quando tecla seta acima pressionada
  aponte para direção 0
  mova 5 passos

quando clicado
  mude Pontos para 0

quando tecla seta abaixo pressionada
  aponte para direção 180
  mova 5 passos

quando tecla seta direita pressionada
  aponte para direção 90
  mova 5 passos

quando tecla seta esquerda pressionada
  aponte para direção -90
  mova 5 passos
  
```

Bom trabalho!!



1.º Ciclo

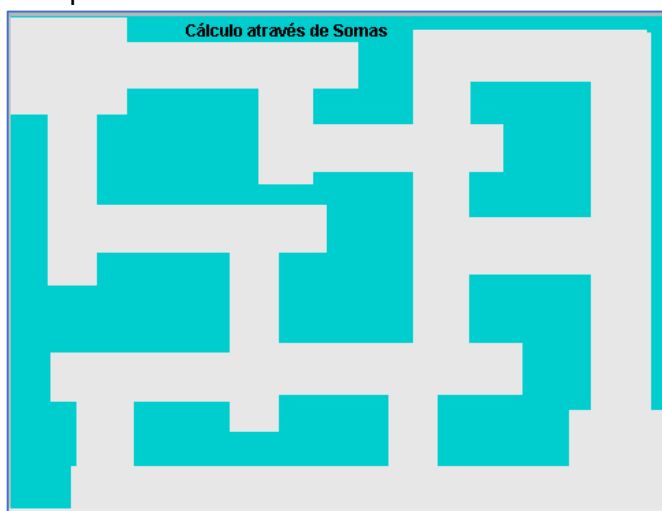
Iniciação à Programação

Exercício 18 – Labirinto

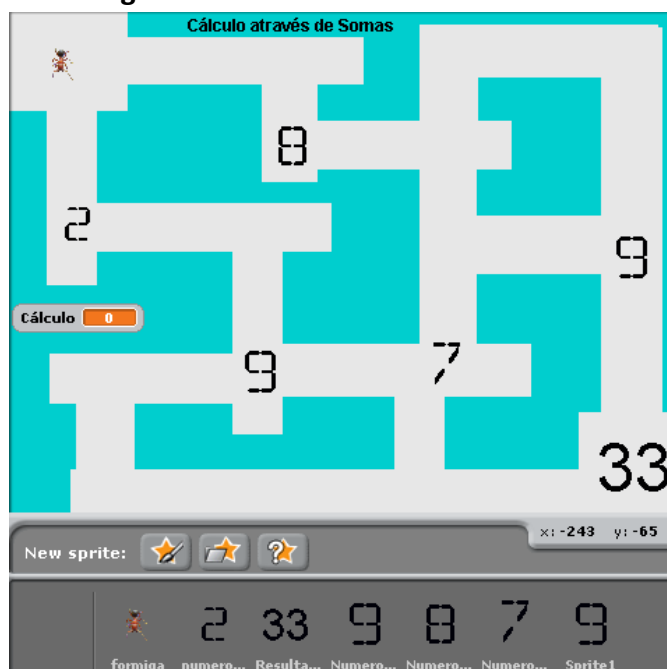
Objetivo: Programação de comportamentos através de condições que estimulem o cálculo mental (Figura 1 - exemplo em que o ator (formiga) terá de encontrar o caminho correto perfazendo a soma de 33.)

Passos a realizar:

1. Abra o Scratch e apague o gato que aparece no início.
2. Crie o seguinte palco:



3. Crie a variável “Cálculo” e insira os diferentes atores. Os números de 0 a 9 estão na pasta **Letters-Digital** e o número 33 deve ser desenhado.

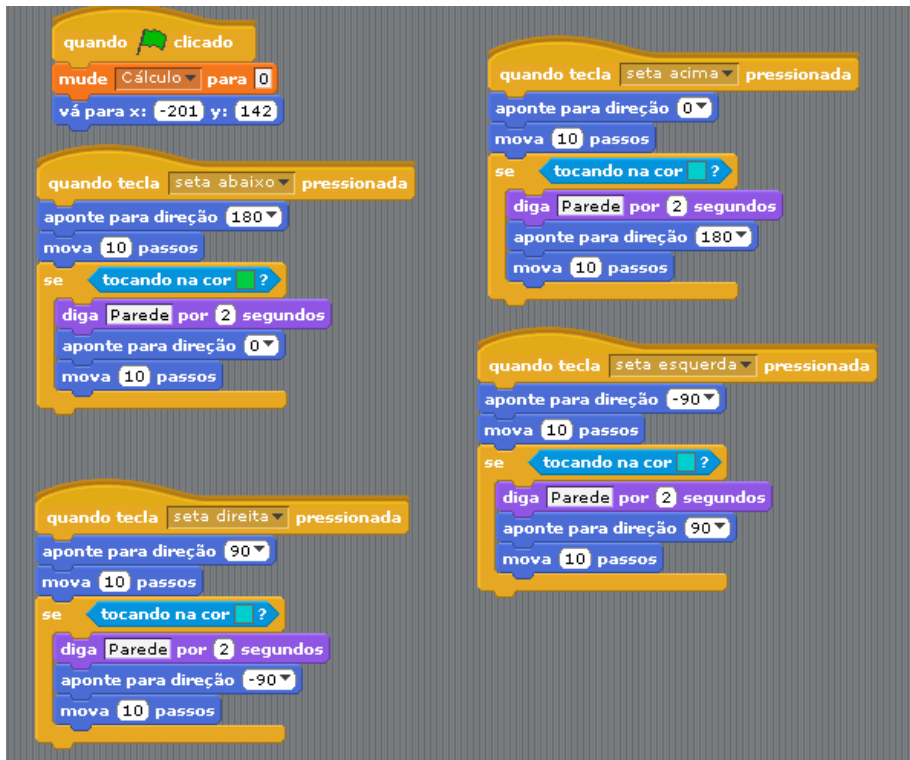




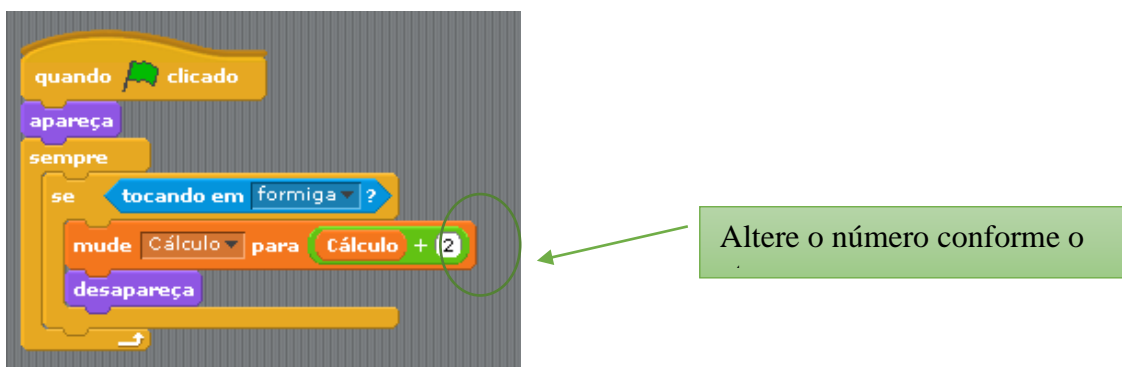
1.º Ciclo

Iniciação à Programação

4. Programe a formiga da seguinte forma:



5. Programe cada um dos números da seguinte forma:



6. Programe o número 33 da seguinte forma:



Bom trabalho!!




1.º Ciclo

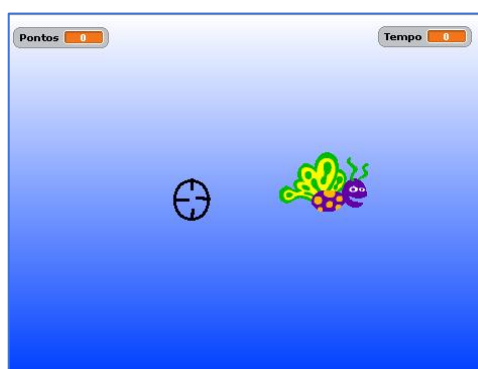
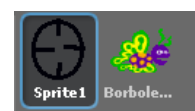
Iniciação à Programação

Exercício 19 – Tiro ao alvo

Objetivo: Introdução de funções para otimização de código. Personalização/programação do cursor.

Passos a realizar:

1. Abra o Scratch e apague o gato que aparece no início.
2. Crie um ator para o alvo e insira o ator borboleta com os seus 2 trajes.
3. Crie a variável Pontos e a variável Tempo. 
4. Crie o seguinte palco e coloque os atores como mostra a figura:



5. Programe da seguinte forma:

Alvo

```

quando clicar em [bandeira]
para sempre
  vai para x: x do rato y: y do rato
quando clicar em [bandeira]
  espera 10 segundos
  diz junta Fez com junta Pontos com Pontos durante 5 segundos
  termina o programa
        
```

Palco

```

quando clicar em [bandeira]
  faz Tempo igual a 0
  faz Pontos igual a 0
  para sempre
    espera 1 segundos
    altera Tempo em 1
        
```

Borboleta

```

quando clicar em [bandeira]
para sempre
  traje seguinte
  desliza 1 segundos para x: sorteia número entre -230 e 230 y: sorteia número entre -170 e 170
quando clicar em Borboleta
  altera Pontos em 1
        
```

Bom trabalho!!



1.º Ciclo

Iniciação à Programação

Exercício 20 – Questionário

Objetivo: Personalização de cenários e atores que, através de funções, podem explorar diferentes conteúdos.

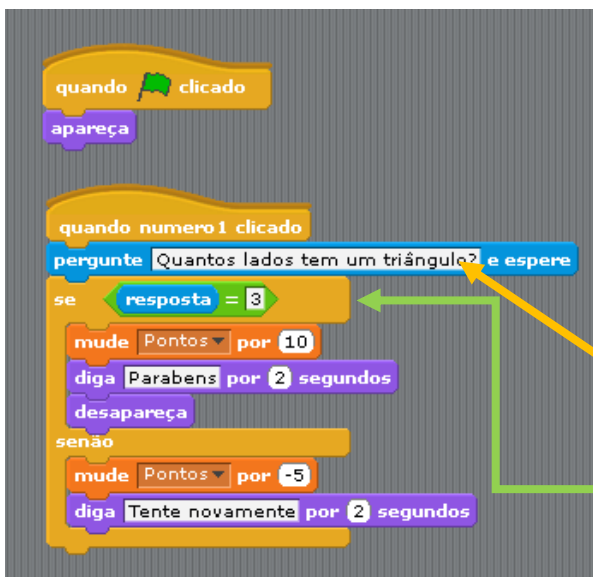
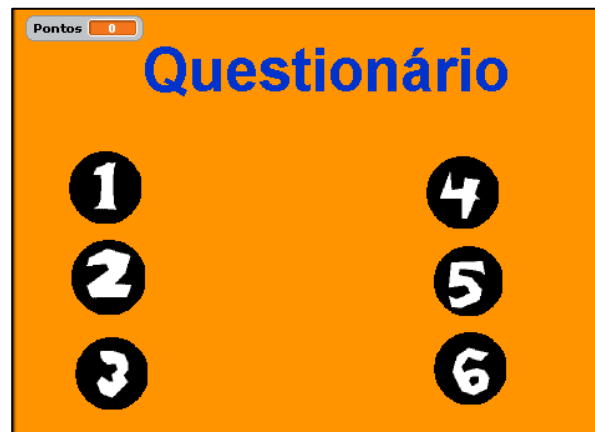
Passos a realizar:

1. Abra o Scratch e apague o gato que aparece no início.

2. Construa o seguinte cenário:

Notas: Os números são atores. A variável chama-se Pontos.

3. Programe o número 1 com o seguinte código:



Programa os outros números alterando apenas a pergunta e a resposta

4. Programe o palco com o seguinte código.

5. Faça as adaptações necessárias para poder usar funções e otimizar o código.

Bom trabalho!!



1.º Ciclo

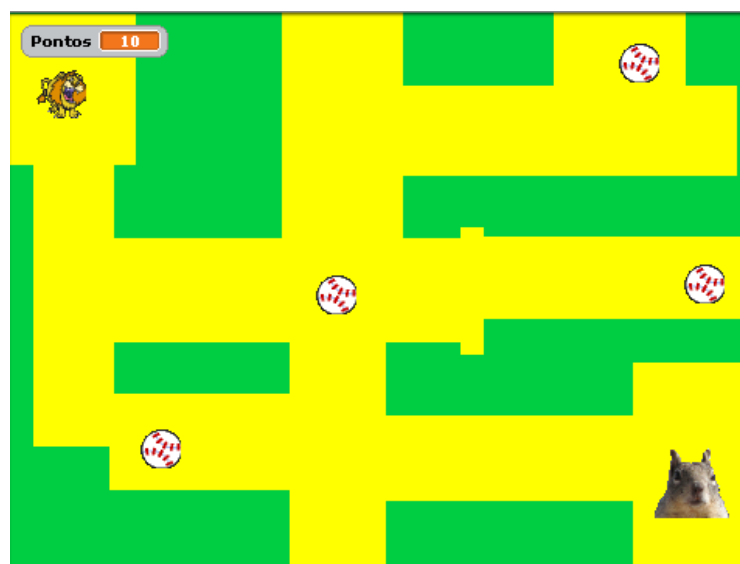
Iniciação à Programação

Exercício 21 – Avaliação

Objetivo: Exploração/Avaliação dos conteúdos explorados.

Passos a realizar:

1. Abra o Scratch e apague o gato que aparece no início.
2. Construa o seguinte cenário e coloque os atores como mostra a figura.



3. Programe os objetos da forma que achar mais adequado. **(Use a imaginação!)**

Sugestões:

- Colocar as bolas a aparecer em posições aleatórias;
- Personalizar as bolas;
- Perder pontos quando tocar nas laterais;
- Ganhar pontos quando apanhar uma bola;
- Usar funções e otimizar o código;
- Acrescentar uma variável para o tempo;
- Colocar mensagens nas personagens.

Bom trabalho!!

Anexo G: Protocolo de Investigação (Questionário, Teste de Autoeficácia para a área de programação e Teste e Pensamento Computacional)



PROJETO DE INVESTIGAÇÃO

“Iniciação à Programação no 1.º Ciclo do Ensino Básico” Análise da implementação da medida a alunos do 3.º ano.

Este projeto de investigação está a ser desenvolvido no âmbito do Doutoramento em Educação na Universidade da Beira Interior (UBI) e tem como objetivo principal avaliar o impacto da Iniciativa no pensamento computacional dos alunos.

Toda a informação recolhida será tratada de forma anónima e confidencial e só a equipa de investigação terá acesso aos dados.

Por favor, responde sinceramente às questões apresentadas. **Obrigado!**

QUESTIONÁRIO

DADOS SOCIODEMOGRÁFICOS:

Ano de Escolaridade:

3.º Ano

4.º Ano

Género:

Masculino

Feminino

Idade: _____ anos

Atividades extracurriculares que frequentas ou já frequentaste: _____

EXPERIÊNCIAS E UTILIZAÇÃO DE TECNOLOGIAS:

Tens computador em casa?

Sim

Não

Tens tablet em casa?

Sim

Não

Quantas vezes por semana utilizas o computador ou tablet em casa?

- Nunca
- 1 ou 2 vezes
- 3 ou 4 vezes
- Frequentemente (5 ou mais vezes)

Quantas vezes por semana utilizas o computador ou tablet, na escola?

- Nunca
- 1 ou 2 vezes
- 3 ou 4 vezes
- Frequentemente (5 ou mais vezes)

Quais são os teus jogos de computador preferidos?

Para que utilizas o computador ou tablet?

- Não uso o computador ou tablet
- Para lazer (jogar, ouvir musica, etc...)
- Apenas para trabalhos escolares ou consultar a escola virtual
- Para trabalhos escolares e lazer

Sabes usar o telemóvel?

- Não sei usar o telemóvel
- Com alguma ajuda
- Sem ajuda

Que uso fazes do telemóvel?

TESTE DE AUTOEFICÁCIA PARA A ÁREA DA PROGRAMAÇÃO

1. **Sou capaz de ligar e desligar corretamente um computador.**
 - Nada verdade
 - Pouco verdade
 - Bastante verdade
 - Totalmente verdade
2. **Sou capaz de identificar os componentes de um computador (ex. rato, teclado, monitor, digitalizador, webcam).**
 - Nada verdade
 - Pouco verdade
 - Bastante verdade
 - Totalmente verdade
3. **Consigo escrever corretamente no computador (ex. todo o tipo de acentuação e pontuação).**
 - Nada verdade
 - Pouco verdade
 - Bastante verdade
 - Totalmente verdade
4. **Sou capaz de organizar a informação no computador (ex. criar pastas, copiar/colar ficheiros, abrir/guardar documentos).**
 - Nada verdade
 - Pouco verdade
 - Bastante verdade
 - Totalmente verdade
5. **Sou capaz, no computador, de identificar e representar sequências de ações de atividades do quotidiano (passos para a realização de uma tarefa como, por exemplo, fazer uma pesquisa sobre um tema específico).**
 - Nada verdade
 - Pouco verdade
 - Bastante verdade
 - Totalmente verdade

-
6. Consigo identificar e utilizar diferentes tipos de dados (texto e números).
- Nada verdade
 - Pouco verdade
 - Bastante verdade
 - Totalmente verdade
7. Sou capaz de distinguir operadores (>, <, =, +, -, *, /, não, ou, e).
- Nada verdade
 - Pouco verdade
 - Bastante verdade
 - Totalmente verdade
8. Num jogo de computador ou tablet, sou capaz de dizer qual o resultado de um conjunto de instruções dadas a uma determinada personagem (ex. as teclas/passos necessários para a personagem chegar a um determinado local).
- Nada verdade
 - Pouco verdade
 - Bastante verdade
 - Totalmente verdade
9. Sou capaz de aplicar estratégias que aprendi num jogo num outro em que esteja a jogar pela primeira vez. (Consigo ver muitas semelhanças entre os jogos)
- Nada verdade
 - Pouco verdade
 - Bastante verdade
 - Totalmente verdade
10. Sou capaz de programar uma personagem num jogo de computador ou tablet para desenhar uma forma geométrica (ex. um quadrado).
- Nada verdade
 - Pouco verdade
 - Bastante verdade
 - Totalmente verdade

11. Eu sou capaz de dividir um problema em partes quando estou a usar um computador ou tablet (ex. quando não consigo fazer um puzzle faço por partes para depois juntar tudo).

- Nada verdade
- Pouco verdade
- Bastante verdade
- Totalmente verdade

12. Consigo identificar e corrigir erros num programa/jogo no computador ou tablet (ex. ajudar um colega a criar as suas instruções para atingir um determinado objetivo).

- Nada verdade
- Pouco verdade
- Bastante verdade
- Totalmente verdade

13. Sou capaz de otimizar um programa ou jogo no computador ou tablet (ex. fazer o mesmo programa, mas com menos instruções ou encontrar uma estratégia para passar um determinado nível mais rapidamente).

- Nada verdade
- Pouco verdade
- Bastante verdade
- Totalmente verdade

14. Consigo planificar e criar um projeto no computador ou tablet (ex. fazer um jogo, uma animação, uma história, etc.).

- Nada verdade
- Pouco verdade
- Bastante verdade
- Totalmente verdade

TESTE DE PENSAMENTO COMPUTACIONAL

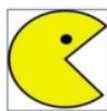
INSTRUÇÕES

O teste é composto por 28 perguntas, distribuídas por 16 páginas com 2 perguntas em cada uma.

Todas as perguntas têm 4 opções de resposta (A, B, C ou D) **das quais apenas uma é a correta.**

A partir do momento que começas o teste, tens 45 minutos para fazeres o melhor que conseguires. Não é necessário que respondas a todas as perguntas.

Antes de começares o teste, vamos ver 3 exemplos para que te familiarizes com o tipo de perguntas que irás encontrar, nas quais vão aparecer as personagens que agora te apresentamos.



Pac-Man



Fantasma



Artista

BOA SORTE

Exemplo I

Neste primeiro exemplo perguntam-te quais os passos que levam o Pac-Man até ao Fantasma pelo caminho assinalado.

A opção correta neste exemplo é a B. Assinala a resposta correta, na tabela que está por baixo da pergunta.

<p>Quais os passos que levam o Pac-Man até ao Fantasma pelo caminho assinalado?</p>	<p>Opção A</p>
	<p>Opção B</p>
	<p>Opção C</p>
	<p>Opção D</p>

Exemplo I

 A

 B

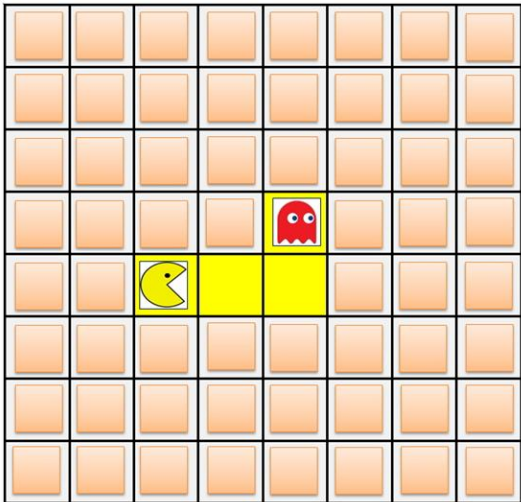
 C

 D

Exemplo II

Neste segundo exemplo perguntam-te, novamente, quais os passos que levam o Pac-Man até ao Fantasma pelo caminho assinalado. Mas neste caso as opções de resposta, em vez de serem setas, são blocos/instruções que encaixam uns nos outros.

A opção correta neste exemplo é a C. Assinala a resposta correta, na tabela que está por baixo da pergunta.

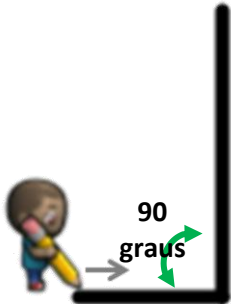
<p>Quais os passos que levam o Pac-Man até ao Fantasma pelo caminho assinalado?</p> 	<p>Opção A</p> <p>mover para a frente virar à esquerda ⤵ mover para a frente mover para a frente</p>	<p>Opção B</p> <p>mover para a frente virar à direita ⤵ mover para a frente mover para a frente</p>
	<p>Opção C</p> <p>mover para a frente ✓ mover para a frente virar à esquerda ⤵ mover para a frente</p>	<p>Opção D</p> <p>mover para a frente mover para a frente virar à direita ⤵ mover para a frente</p>

Exemplo I A B C D

Exemplo III

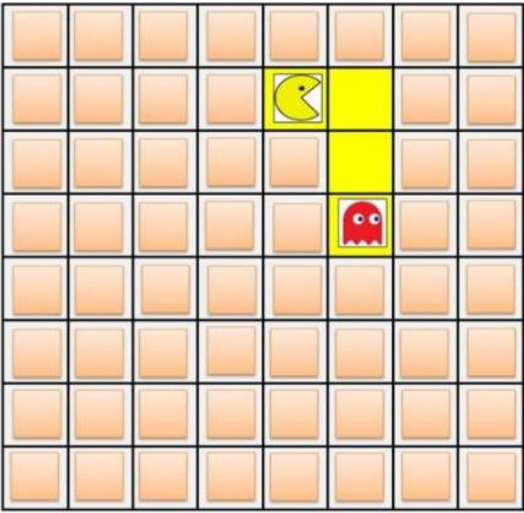




Neste terceiro exemplo perguntam-te quais os passos que deve seguir o Artista para desenhar a figura que aparece no quadro. Ou seja, como o lápis se deve mover para que a figura seja desenhada.

A opção correta neste exemplo é a A. Assinala a resposta correta, na tabela que está por baixo da pergunta.

<p>Quais os passos que o Artista deve executar para desenhar a figura? O lado menor mede 50 píxeis e o maior mede 100 píxeis.</p> 	<p>Opção A</p> <p>mover para a frente por 50 píxeis ✓ virar à esquerda por 90 graus mover para a frente por 100 píxeis</p>	<p>Opção B</p> <p>mover para a frente por 50 píxeis virar à direita por 90 graus mover para a frente por 100 píxeis</p>
	<p>Opção C</p> <p>mover para a frente por 100 píxeis virar à esquerda por 90 graus mover para a frente por 50 píxeis</p>	<p>Opção D</p> <p>mover para a frente por 100 píxeis virar à direita por 90 graus mover para a frente por 50 píxeis</p>

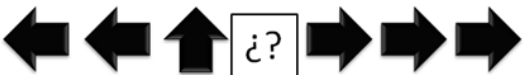
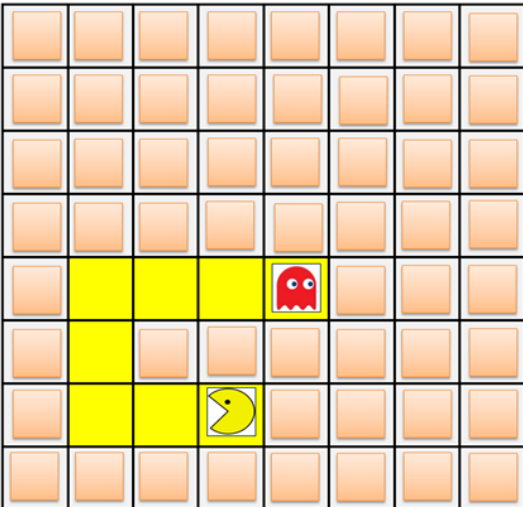




Exemplo I A B C D

Questão 1

<p>Quais os passos que levam o Pac-Man até ao Fantasma pelo caminho assinalado?</p> 	<p>Opção A</p> <p style="text-align: center;">  </p> <hr/> <p>Opção B</p> <p style="text-align: center;">  </p> <hr/> <p>Opção C</p> <p style="text-align: center;">  </p> <hr/> <p>Opção D</p> <p style="text-align: center;">  </p>
---	--

Questão 1
 A
 B
 C
 D

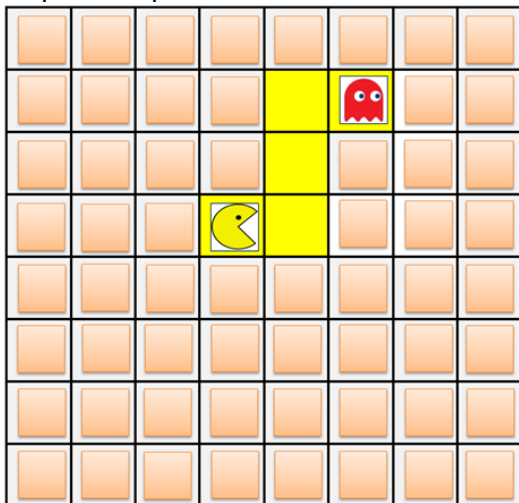
Questão 2

<p>Que passo falta à sequência para levar o Pac-Man até ao Fantasma pelo caminho assinalado?</p> <p style="text-align: center;">  </p> 	<p>Opção A</p> <p style="text-align: center;">  </p> <hr/> <p>Opção B</p> <p style="text-align: center;">  </p> <hr/> <p>Opção C</p> <p style="text-align: center;">  </p> <hr/> <p>Opção D</p> <p style="text-align: center;">  </p>
---	---

Questão 2
 A
 B
 C
 D

Questão 3

Para levar o Pac-Man até ao Fantasma pelo caminho selecionado, qual o passo da sequência que está **errado**?

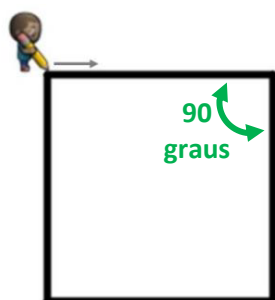


- mover para a frente → Passo A
- virar à esquerda ↻ → Passo B
- mover para a frente → Passo C
- virar à esquerda ↻ → Passo D
- mover para a frente

Questão 3 A B C D

Questão 4

Quais os passos que o Artista deve executar para desenhar o quadrado? Cada um dos lados do quadrado mede 100 píxeis.



Opção A

- mover para a frente por 100 píxeis
- virar à direita por 90 graus
- mover para a frente por 100 píxeis
- virar à esquerda por 90 graus
- mover para a frente por 100 píxeis
- virar à direita por 90 graus
- mover para a frente por 100 píxeis

Opção B

- mover para a frente por 25 píxeis
- virar à direita por 90 graus
- mover para a frente por 25 píxeis
- virar à esquerda por 90 graus
- mover para a frente por 25 píxeis
- virar à direita por 90 graus
- mover para a frente por 25 píxeis

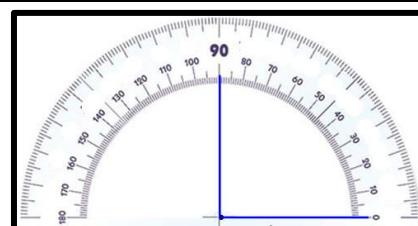
Opção C

- mover para a frente por 50 píxeis
- virar à direita por 90 graus
- mover para a frente por 50 píxeis
- virar à direita por 90 graus
- mover para a frente por 50 píxeis
- virar à direita por 90 graus
- mover para a frente por 50 píxeis

Opção D

- mover para a frente por 100 píxeis
- virar à direita por 90 graus
- mover para a frente por 100 píxeis
- virar à direita por 90 graus
- mover para a frente por 100 píxeis
- virar à direita por 90 graus
- mover para a frente por 100 píxeis

Questão 4 A B C D



Questão 5

Qual a sequência que leva o Pac-Man até ao Fantasma pelo caminho assinalado?

Número de vezes que repete um determinado passo (neste exemplo 8 passos em frente)

Opção A

Opção B

Opção C

Opção D

Questão 5 A B C D

Questão 6

Quantas vezes se deve repetir a seguinte sequência para levar o Pac-Man até ao Fantasma pelo caminho assinalado?

Opção A

X 2

Opção B

X 1

Opção C

X 4

Opção D

X 3

Questão 6 A B C D

Questão 7

Para que o Artista desenhe **uma vez** o seguinte retângulo (50 pixéis de largura e 100 pixéis de altura), qual o passo da seguinte sequência que está **incorreto**?



Passo A

```

repetir 4 vezes
fazer
  mover para a frente por 50 pixéis
  virar à esquerda por 90 graus
  mover para a frente por 100 pixéis
  virar à esquerda por 90 graus
  
```

Passo B

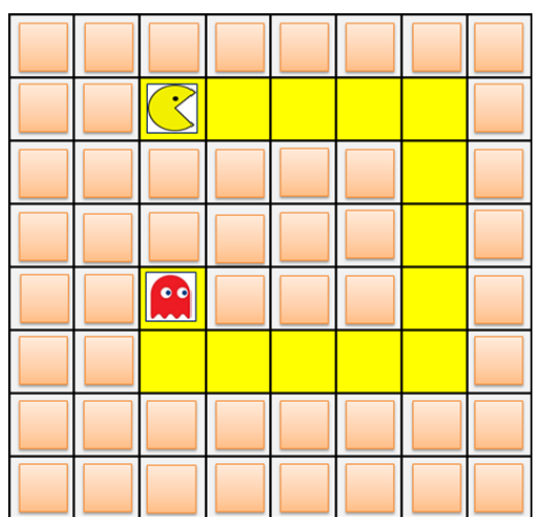
Passo C

Passo D

Questão 7 A B C D

Questão 8

Qual a sequência que leva o Pac-Man até ao Fantasma pelo caminho assinalado?



Opção A

```

repetir 4 vezes
fazer
  repetir 3 vezes
  fazer mover para a frente
  virar à direita
  mover para a frente
  
```

Opção B

```

repetir 3 vezes
fazer
  repetir 4 vezes
  fazer mover para a frente
  virar à direita
  mover para a frente
  
```

Opção C

```

repetir 3 vezes
fazer
  repetir 4 vezes
  fazer mover para a frente
  virar à direita
  mover para a frente
  
```

Opção D

```

repetir 4 vezes
fazer mover para a frente
repetir 4 vezes
fazer virar à direita
mover para a frente
  
```

Questão 8 A B C D

Questão 9

Qual a sequência que leva o Pac-Man até ao Fantasma pelo caminho assinalado?

<p>Opção A</p>	<p>Opção B</p>
<p>Opção C</p>	<p>Opção D</p>

Questão 9 A B C D

Questão 10

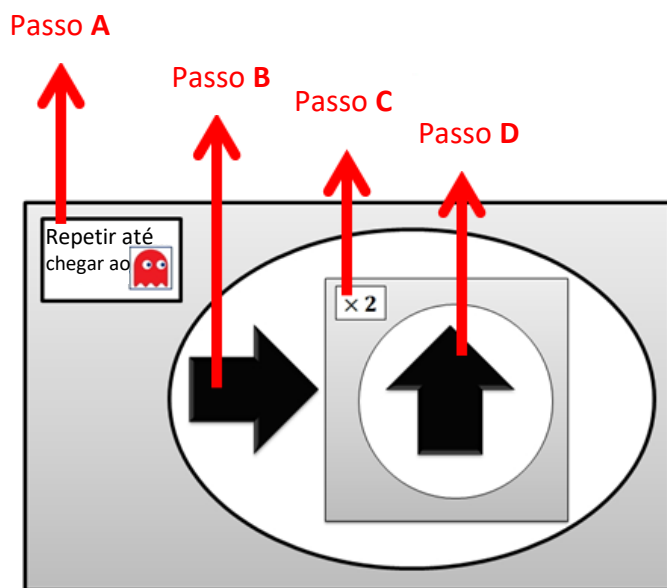
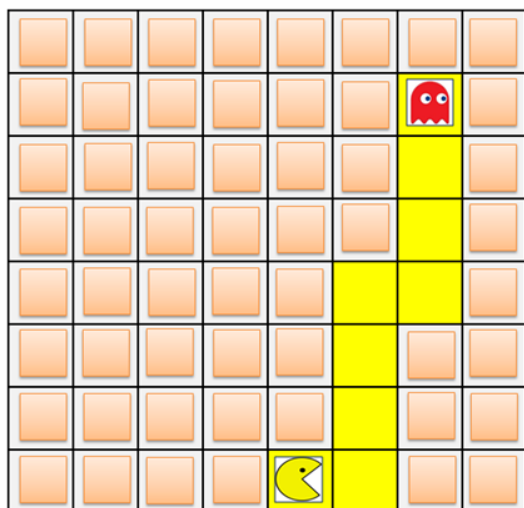
Qual o passo que falta na seguinte sequência para que o Pac-Man vá até ao Fantasma pelo caminho assinalado?

<p>Opção A</p>	<p>Opção B</p>
<p>Opção C</p>	<p>Opção D</p> <p>Não falta nenhum passo</p>

Questão 10 A B C D

Questão 11

Para que o Pac-Man chegue até ao Fantasma pelo caminho assinalado, qual o passo da seguinte sequência que está **incorreto**?



Questão 11 A B C D

Questão 12

Qual a sequência que faz com que o Artista desenhe a escada até à flor? Cada degrau sobe 30 píxeis.



Opção A

```

repetir até à flor
fazer
  repetir 4 vezes
  fazer
    mover para a frente por 30 píxeis
    virar à direita por 90 graus
  saltar para a frente por 30 píxeis
  
```

Opção B

```

repetir até à flor
fazer
  repetir 4 vezes
  fazer
    mover para a frente por 120 píxeis
    virar à direita por 90 graus
  saltar para a frente por 30 píxeis
  
```

Opção C

```

repetir até à flor
fazer
  repetir 4 vezes
  fazer
    mover para a frente por 30 píxeis
    virar à direita por 90 graus
  saltar para a frente por 210 píxeis
  
```

Opção D

```

repetir até à flor
fazer
  repetir 7 vezes
  fazer
    mover para a frente por 30 píxeis
    virar à direita por 90 graus
  saltar para a frente por 30 píxeis
  
```

Questão 12 A B C D

Questão 13

Qual a sequência que leva o Pac-Man até ao Fantasma pelo caminho assinalado?

Opção A

Opção B

Opção C

Opção D

Questão 13 A B C D

Questão 14

Qual a sequência que leva o Pac-Man até ao Fantasma pelo caminho assinalado?

Opção A

Opção B

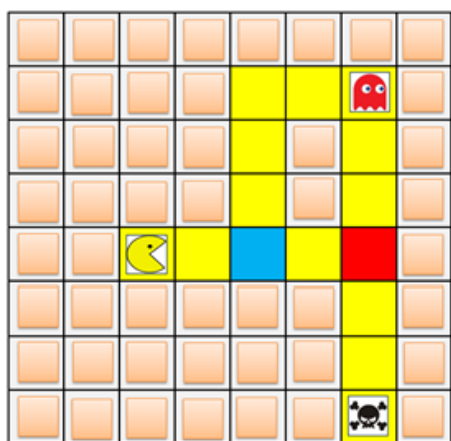
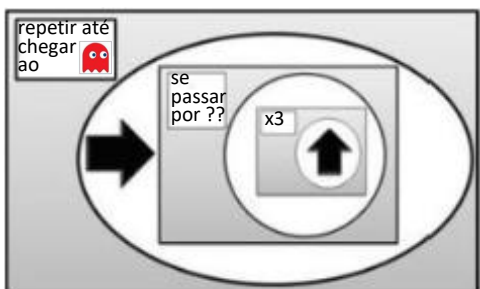
Opção C

Opção D

Questão 14 A B C D

Questão 15

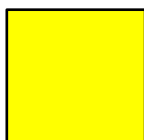
O que falta na seguinte sequência para que o Pac-Man vá até ao Fantasma pelo caminho assinalado?



Opção A



Opção B



Opção C



Opção D

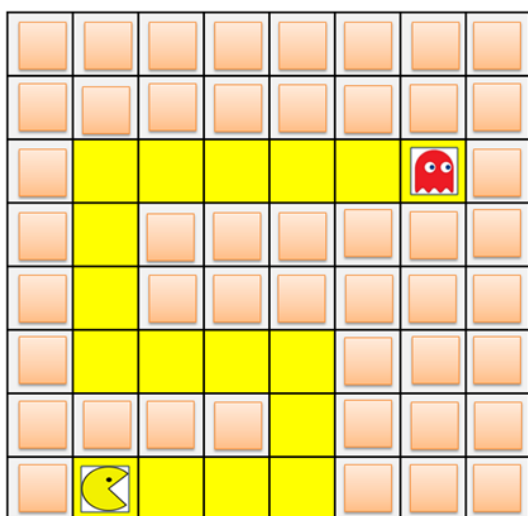
Tanto a opção A como a opção C estão corretas

Questão 15

- A B C D

Questão 16

Para que o Pac-Man chegue até ao Fantasma pelo caminho assinalado, qual o passo da seguinte sequência que está **incorreto**?



```

repetir até [Fantasma]
  fazer
    mover para a frente
    se há caminho para a esquerda → Passo A
    fazer virar à esquerda → Passo B
    se há caminho para a direita → Passo C
    fazer mover para a frente → Passo D
  
```

Questão 16

- A B C D

Questão 17

<p>Qual a sequência que leva o Pac-Man até ao Fantasma pelo caminho assinalado?</p>	<p>Opção A</p> <pre> repetir até fazer se há caminho em frente fazer mover para a frente se não virar à esquerda </pre>	<p>Opção B</p> <pre> repetir até fazer se há caminho em frente fazer mover para a frente se não virar à direita </pre>
	<p>Opção C</p> <pre> repetir até fazer se há caminho para a direita fazer virar à direita se não mover para a frente </pre>	<p>Opção D</p> <pre> repetir até fazer se há caminho para a esquerda fazer virar à esquerda se não mover para a frente </pre>

Questão 17 A B C D

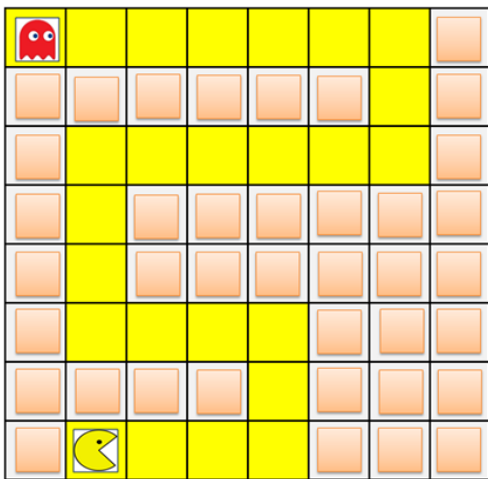
Questão 18

<p>Qual a sequência que leva o Pac-Man até ao Fantasma pelo caminho assinalado?</p>	<p>Opção A</p> <pre> repetir até fazer se há caminho em frente fazer mover para a frente se não virar à esquerda </pre>	<p>Opção B</p> <pre> repetir até fazer se há caminho em frente fazer mover para a frente se não virar à direita </pre>
	<p>Opção C</p> <pre> repetir até fazer se há caminho para a direita fazer virar à direita se não mover para a frente </pre>	<p>Opção D</p> <pre> repetir até fazer se há caminho para a esquerda fazer virar à esquerda se não mover para a frente </pre>

Questão 18 A B C D

Questão 19

Para que o Pac-Man chegue até ao Fantasma pelo caminho assinalado, qual o passo da seguinte sequência que está **incorreto**?



```

repetir até
  fazer
    se há caminho em frente
      fazer mover para a frente
    se não
      se há caminho para a direita
        fazer virar à esquerda
      se não
        virar à direita
  
```

→ Passo A (mover para a frente)
 → Passo B (se há caminho para a direita)
 → Passo C (virar à esquerda)
 → Passo D (virar à direita)

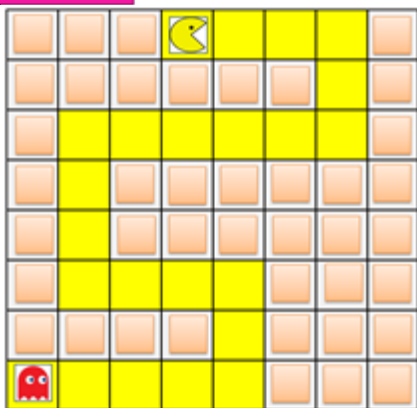
Questão 19 A B C D

Questão 20

Qual o passo que falta na seguinte sequência para que o Pac-Man vá até ao Fantasma pelo caminho assinalado?

```

repetir até
  fazer
    se há caminho em frente
      fazer mover para a frente
    se não
      se há caminho para a direita
        fazer virar à direita
      se não
        ??????
  
```



Opção A

```
mover para a frente
```

Opção B

```
virar à direita
```

Opção C

```
virar à esquerda
```

Opção D

Não falta nenhum passo

Questão 20 A B C D

Questão 21

Qual a sequência que leva o Pac-Man até aos Morangos pelo caminho assinalado e faz com que ele coma o número de Morangos indicado?



Opção A

```

enquanto houver caminho em frente
fazer mover para a frente
repetir 3 vezes
fazer comer 1 morango
    
```

Opção B

```

enquanto houver caminho em frente
fazer mover para a frente
repetir 4 vezes
fazer comer 1 morango
    
```

Opção C

```

enquanto houver caminho em frente
fazer mover para a frente
repetir 5 vezes
fazer comer 1 morango
    
```

Opção D

```

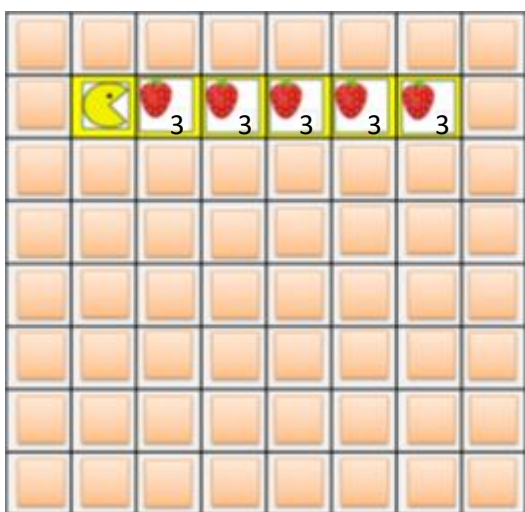
enquanto houver caminho em frente
fazer mover para a frente
repetir 3 vezes
fazer comer 1 morango
    
```

Questão 21

- A B C D

Questão 22

Qual a sequência que leva o Pac-Man até aos Morangos pelo caminho assinalado e faz com que ele coma o número de Morangos indicado?



Opção A

```

enquanto houver caminho em frente
fazer repetir 5 vezes
fazer mover para a frente
repetir 3 vezes
fazer comer 1 morango
    
```

Opção B

```

enquanto houver caminho em frente
fazer mover para a frente
repetir 3 vezes
fazer comer 1 morango
    
```

Opção C

```

enquanto houver caminho em frente
fazer repetir 3 vezes
fazer mover para a frente
repetir 5 vezes
fazer comer 1 morango
    
```

Opção D

```

enquanto houver caminho em frente
fazer mover para a frente
repetir 3 vezes
fazer comer 1 morango
    
```

Questão 22

- A B C D

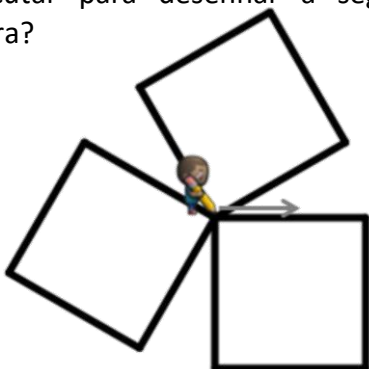
Questão 23

Se tivermos o seguinte conjunto de passos a que chamamos de “A minha função”, e que desenha um quadrado com 100 pixéis de lado:

```

Função
A minha função
  repetir 4 vez
  fazer
    mover para a frente por 100 pixéis
    virar à direita por 90 graus
  
```

Qual a sequência que o Artista deve executar para desenharmos a seguinte figura?



Opção A

```

repetir 3 vez
fazer
  A minha função
  virar à direita por 120 graus

```

Opção B

```

repetir 3 vez
fazer
  A minha função
  virar à direita por 120 graus

```

Opção C

```

repetir 4 vez
fazer
  A minha função
  virar à direita por 90 graus

```

Opção D

```

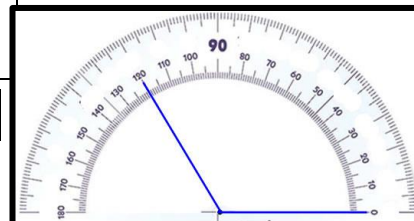
repetir 4 vez
fazer
  A minha função
  virar à direita por 90 graus

```

Questão 25

- A B C D

Questão 26



Se tivermos o seguinte conjunto de passos a que chamamos de “A minha função” e que desenha um triângulo com 50 pixéis de lado:

```

Função
A minha função
  repetir 3 vez
  fazer
    mover para a frente por 50 pixéis
    virar à esquerda por 120 graus
  
```

O que falta na seguinte sequência para que o Artista desenhe a seguinte figura?

```

repetir ?? vez
fazer
  A minha função
  saltar para a frente por 50 pixéis

```



Opção A

15

Opção B

5

Opção C

4

Opção D

3

Questão 26

- A B C D

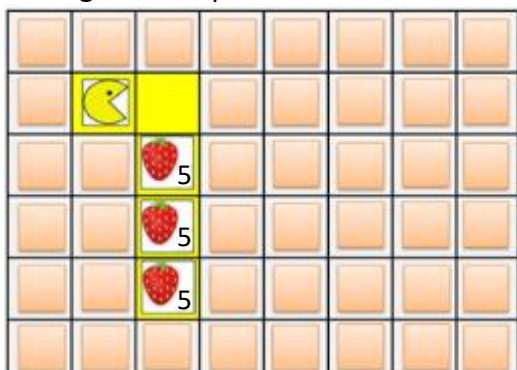
Questão 27

Se tivermos o seguinte conjunto de passos a que chamámos de “comer 5”:

```

Função
  comer 5
  repetir 5 vez
  fazer
    comer 1 morango
  
```

Qual a sequência que leva o Pac-Man até aos Morangos pelo caminho assinalado e faz com que ele coma o número de Morangos correspondentes?



Opção A

```

mover para a frente
virar à direita
repetir 3 vezes
fazer
  mover para a frente
  comer 5
  
```

Opção B

```

mover para a frente
virar à direita
repetir 3 vezes
fazer
  comer 5
  mover para a frente
  
```

Opção C

```

mover para a frente
virar à direita
repetir 5 vezes
fazer
  mover para a frente
  comer 5
  
```

Opção D

```

mover para a frente
virar à direita
repetir 5 vezes
fazer
  comer 5
  mover para a frente
  
```

Questão 27

- A B C D

Questão 28

Se tivermos o seguinte conjunto de passos a que chamámos de “andar e comer 4”:

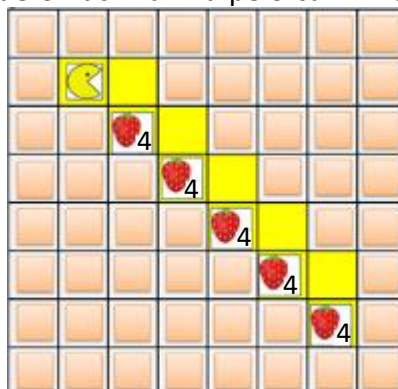
```

Função
  andar e comer 4
  mover para a frente
  virar à direita
  mover para a frente
  repetir 4 vezes
  fazer
    comer 1 morango
  virar à esquerda
  
```

O que falta na seguinte sequência para que o Pac-Man vá pelo caminho assinalado comendo todos os Morangos?

```

repetir ??? vezes
fazer
  andar e comer 4
  
```



Opção A

3

Opção B

4

Opção C

5

Opção D

6

Questão 28

- A B C D