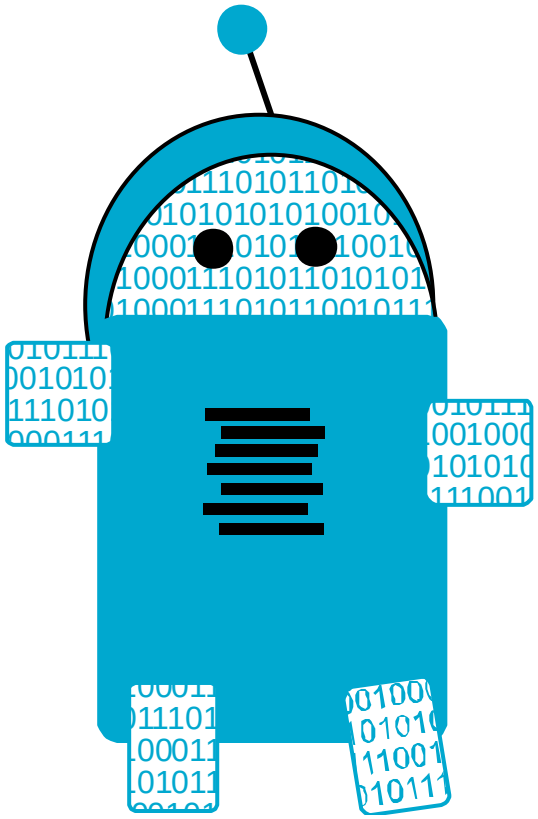
Iniciação à Programação no 1.º Ciclo do Ensino Básico

Linhas Orientadoras



Setembro de 2015

GOVERNO DE PORTUGAL | MINISTÉRIO DA EDUCAÇÃO E CIÊNCIA

Direção-Geral da Educação ■ Av. 24 de Julho, 140 - 1399-025 Lisboa - Portugal
Telef.: +(351) 21 393 45 00 - Fax: +(351) 21 393 46 95 ■ Email: dge@dge.mec.pt Internet: www.dge.mec.pt

Sumário

Direção-Geral da Educação ■ Av. 24 de Julho, 140 - 1399-025 Lisboa - Portugal
Telef.: +(351) 21 393 45 00 - Fax: +(351) 21 393 46 95 ■ Email: dge@dge.mec.pt Internet: www.dge.mec.pt

Introdução

Este documento fornece as linhas orientadoras para os professores que, no âmbito do projeto-piloto **Iniciação à Programação no 1.º Ciclo do Ensino Básico**, desenvolvam atividades com alunos do 3.º e do 4.º ano de escolaridade.

A iniciação à programação deverá ser entendida como mais uma ferramenta ao serviço e em articulação com as restantes áreas curriculares e não como mais uma área disciplinar. O trabalho a desenvolver deverá ser, tanto quanto possível, articulado com o professor titular da turma. Este poderá, por exemplo, trabalhar com os alunos na definição de projetos para alguns dos temas a abordar, na elaboração e correção de textos, ou na seleção de imagens e de sons a utilizar nos projetos.

A principal finalidade deste projeto-piloto é a de que os alunos não só aprendam a programar mas, ao mesmo tempo, aprendam programando. A programação, para além de desenvolver nos alunos a sua criatividade em ciências da computação, promove uma visão mais alargada dos diferentes usos do computador e contribui para o desenvolvimento do pensamento computacional.

Propõe-se que os alunos do 3.º e do 4.º ano efetuem um percurso de aprendizagem de modo evolutivo, podendo adaptar-se de acordo com as horas semanais destinadas às atividades e às características específicas dos alunos de cada escola/agrupamento. Assim, estas linhas orientadoras estão organizadas em duas fases, devendo os alunos iniciar as atividades com propostas correspondentes à primeira fase e só depois avançarem para os temas propostos na segunda.

O documento aborda conceitos relativos às Tecnologias da Informação e Comunicação, às Ciências da Computação e ao Pensamento Computacional e 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.

1. Tecnologias de Informação e Comunicação, Ciências da Computação e Pensamento Computacional

A utilização das Tecnologias de Informação e Comunicação (TIC) nas escolas têm vindo a aumentar com a disponibilização e facilidade de acesso de equipamentos e programas. Tais factos têm acrescentado, muitas vezes, uma mais-valia significativa para as aprendizagens dos alunos. Diversos projetos com TIC têm sido implementados por professores e alunos e muitos produtos têm sido construídos com recurso a diferentes aplicações informáticas, desenvolvidas especialmente para fins educativos, ou utilizando *software* de uso corrente, como os programas de processamento de texto, apresentações eletrónicas, folhas de cálculo, editores de imagem, de som e de vídeo, entre outros.

Utilizar uma linguagem de programação será certamente uma maneira de compreender e desenvolver o pensamento computacional, embora este seja mais do que programar computadores. As TIC e as Ciências da Computação (CC), áreas onde se insere este projeto, complementam-se em muitos aspetos. Enquanto o estudo das TIC permite conhecer e encontrar soluções informáticas, de *software* e de *hardware*, na resolução de problemas reais do nosso dia-a-dia, por exemplo, escrever um texto, procurar informação e comunicá-la de forma eficiente, nas CC o processo centra-se no desenvolvimento do pensamento computacional. Assim, as TIC permitem conhecer as ferramentas computacionais e perceber como se utilizam, enquanto as CC permitem perceber como aquelas se constroem e funcionam.

Miles Berry (2013), membro da agência inglesa Naace (anteriormente designada National Association of Advisers For Computers in Education), propõe uma correspondência entre termos usados nas TIC e os usados em CC (consultar Tabela 1). Segundo esta perspetiva, deixaremos de ter consumidores para falarmos antes em criadores, colaboradores em vez de comunicadores, responsabilidade em vez de segurança, compreensão em vez de habilidade.

TIC	CC
Users	*Makers*
Consumers	*Creators*
Communicators	*Collaborators*
Digitally literate	*Digitally critical*
Safe	*Responsible*
Skills	*Understanding*

Tabela 1- Comparação entre TIC e CC, segundo Berry (2013)

A utilização do computador deve, portanto, também ser vista como uma atividade que permite o desenvolvimento do pensamento computacional, através da possibilidade de resolver problemas do mundo real de forma criativa, não se centrando apenas na programação, mas principalmente nos aspetos de conceção, planificação e implementação, necessários ao desenvolvimento de um determinado projeto. Mais do que saber se um problema é fácil ou difícil, é importante encontrar uma solução, o que permite a utilização do pensamento computacional em muitas situações, incluindo as do nosso dia a dia.

Para Wing (2006), o pensamento computacional é construído a partir da análise das capacidades e limitações dos processos de tratamento de informação, quer estes sejam executados por computadores, quer sejam executados por humanos. Ao contrário do que se poderia pensar, o pensamento computacional não é exclusivo dos computadores, embora, muitas vezes, o associemos a eles quase instintivamente. Atualmente, a literacia informática e o pensamento computacional são também consideradas competências essenciais que os estudantes devem desenvolver (P21's Framework for 21st Century Learning, 2015), tal como anteriormente o foram a leitura e a escrita, bem como a realização de operações aritméticas. Nesta medida, estes ambientes de aprendizagem devem ser implementados, de forma a permitir aos alunos o seu desenvolvimento.

Ramos e Espadeiro (2014) referem que:

> *"O pensamento computacional tem recebido considerável interesse por parte da comunidade científica e educativa e resulta, em boa parte, da chamada de atenção de Jeannete Wing que, através do texto seminal "Computational Thinking", escrito em 2006 onde a autora reintroduziu o conceito e reclamou o seu uso e adoção por todos os cidadãos, incluindo jovens e crianças, como forma de proporcionar os conhecimentos e capacidades decorrentes das formas e recursos cognitivos próprios das ciências da computação e que, pela sua natureza transdisciplinar e universal, poderia ser útil a todos, recusando a ideia, até aí dada como adquirida, de que estas capacidades apenas seriam destinadas aos cientistas da computação." (p. 5)*

Para Brennan, Chung e Hawson (2011) existem conceitos relacionados com o pensamento computacional que são evidenciados quando o aluno programa:

Sequências

Sempre que executamos uma série de comandos em programação, eles são interpretados sequencialmente. A ordem pela qual aparecem é importante. Muitas

vezes basta trocar a ordem de dois elementos para obtermos resultados completamente diferentes.

Ciclos

A mesma sequência pode ser executada várias vezes. Depois de criar programas com sequências de comandos, os alunos reconhecerão padrões de repetição. A utilização de ciclos, sendo mais exigente em termos de pensamento computacional do que uma simples sequência, tornará os programas mais pequenos, mais legíveis e fáceis de compreender.

Eventos

Acontecimentos que desencadeiam uma determinada ação.

Condições

A programação nem sempre é linear. De acordo com determinadas condições e mediante a utilização de estruturas de decisão, o programa poderá tomar diferentes rumos.

Operadores

Os alunos utilizarão operadores para realizarem operações matemáticas e/ou lógicas.

Dados/Variáveis

Em programação será necessário armazenar, recuperar e atualizar valores que serão guardados em variáveis.

Execução em paralelo

Quando executamos um programa, muitas vezes várias ações iniciam-se em paralelo. Será necessário compreender este conceito e programar de modo a que os eventos aconteçam quando necessário e previsto.

Estes autores identificam também práticas associadas ao ato de programar que podem ser importantes no desenvolvimento de competências essenciais para o séc. XXI:

Ação iterativa e incremental

Um projeto de programação é desenvolvido por etapas. Apenas quando parte do projeto funciona corretamente, se avança para o desenvolvimento das etapas seguintes, que, muitas vezes, também poderão ser testadas isoladamente.

Teste e depuração

Depois de concluir um programa (ou uma etapa), é necessário testar e certificar-se que tudo funciona como estava previsto. Muitas vezes, nesta fase, e na partilha de

projetos, são encontrados erros que tinham passado despercebidos ao longo do processo de construção do programa e que deverão ser corrigidos.

Decomposição e abstração

Problemas complexos podem ser divididos em problemas mais simples. Por exemplo, se pretendermos desenhar vários quadrados, é possível criar um programa para desenhar um. Desenhar os restantes será simples, reformulando ou repetindo o programa anterior. Mas se tivermos que desenhar vários polígonos regulares, podemos tentar desenhar um quadrado, um pentágono e um hexágono regulares e depois generalizar, obtendo um programa que permita desenhar um qualquer polígono regular, indicando o número de lados.

Reutilização e reformulação

Os projetos podem ser construídos partindo de outros, sendo apenas reformulados ou adaptados. Pode ainda haver partes do projeto que possam ser programadas como funções ou procedimentos que se repetem ao longo da programação, construindo projetos grandes a partir de partes mais pequenas.

Os autores mencionados anteriormente identificam ainda perspetivas relacionadas com o pensamento computacional, que ajudam a que o programador se veja de uma maneira diferente ou veja o mundo de modo diferente. As perspetivas que a seguir se enumeram não serão exclusivas do pensamento computacional, mas, através da programação e da partilha dos produtos obtidos, é possível ajudar a desenvolvê-las e a estimulá-las:

Expressar

A partilha dos projetos desenvolvidos no grupo/turma e depois na rede ou em eventos onde estejam presentes outros elementos da comunidade escolar permite que o aluno possa, não só mostrar como se expressou ao construir o programa, mas também demonstrar e defender as suas ideias. É fundamental que os alunos percebam que através do computador e da programação podem comunicar, mostrando as suas ideias e criações. Isto vai ajudar a reconhecer-se e a ver-se a ele próprio como um construtor e não um mero consumidor.

Colaborar

Os alunos devem reconhecer a vantagem de criar com e para os outros. Podem, por exemplo, explorar programas anteriormente construídos como forma de

inspiração, percebendo que as suas criações servirão também de fonte de inspiração para outros. Isto ajudará ainda ao desenvolvimento do espírito crítico, à compreensão do seu funcionamento e a forma como foi feito. É importante que analisem o trabalho criado entre pares.

Questionar

Os alunos que desenvolvem os seus programas também se questionam sobre a forma como outros terão sido desenvolvidos, como foram solucionados determinados problemas, levando-os a questionar como funcionam diversas situações do mundo real.

2. Objetivos

Com o projeto **Iniciação à Programação no 1.º Ciclo do Ensino Básico** pretendemos que os alunos sejam capazes de:

- entender e aplicar princípios e conceitos fundamentais das Ciências da Computação;
- descrever e representar simbolicamente sequências de ações de atividades do quotidiano;
- planificar sequências de instruções que permitam a realização de uma dada tarefa;
- utilizar diferentes tipos de dados (textos, números, entre outros);
- criar sequências de instruções que envolvam seleção e repetição;
- analisar algoritmos, identificando o seu resultado;
- reconhecer que um algoritmo pode ser reutilizado em diferentes situações;
- identificar um problema e decompô-lo em subproblemas;
- planificar e criar um projeto de forma estruturada;
- identificar e corrigir erros existentes na programação de um projeto;
- otimizar a programação da solução encontrada;
- resolver problemas, criar histórias animadas e construir jogos com recurso ao desenvolvimento de programas informáticos;
- usar as Tecnologias de Informação e Comunicação de forma responsável, competente, segura e criativa;
- desenvolver competências nas diferentes áreas das componentes do currículo, bem como nas áreas transversais, por exemplo, no âmbito da Educação para a Cidadania, em articulação com o professor titular da turma, sempre que o mesmo não seja o responsável pela implementação deste projeto;
- apresentar um projeto desenvolvido pelo seu grupo e partilhá-lo com outros;
- analisar e comentar projetos desenvolvidos pelos pares.

Este projeto-piloto tem a implementação de um ano de escolaridade. Contudo, apresentamos na Tabela 2, uma proposta de conteúdos a abordar e respetivos descritores de desempenho, a desenvolver em duas fases do projeto. Numa fase inicial sugerimos uma abordagem introdutória a cada um dos conteúdos e, numa segunda fase, um aprofundamento dos mesmos.

CONTEÚDOS	1ª FASE	2ª FASE
Algoritmos	Os alunos devem: - reconhecer que um algoritmo é um conjunto de instruções concretas, com uma determinada sequência, que permitem alcançar um objetivo; - reconhecer que um algoritmo pode ser representado de forma simples e pode descrever, por exemplo, as atividades que fazemos no dia a dia; - reconhecer que os computadores precisam de instruções mais concretas que os humanos e uma alteração no algoritmo culminará numa mudança, percetível ou não, no resultado do programa.	Os alunos devem: - compreender que vários algoritmos podem produzir os mesmos resultados, sendo uns mais eficientes que outros; - ser capazes de decompor um problema em partes; - compreender que os programas podem ser otimizados e que dois programas podem ter o mesmo efeito tendo programações diferentes.
Programação	Os alunos devem: - ler e interpretar programas já existentes, compreendendo o funcionamento dos comandos envolvidos e verbalizando a finalidade do programa; - ser capazes de criar programas que envolvam: - animações de personagens; - relato de histórias; - criação artística; - jogos interativos; - entre outros.	Os alunos devem: - ser capazes de compreender e utilizar conceitos fundamentais na programação: - sequências; - execução em paralelo; - eventos; - condições/estruturas de decisão; - ciclos; - operadores; - dados/variáveis; - criar programas interativos, com a utilização de teclas e/ou rato para fornecer informação.
Segurança da Informação	Os alunos devem: - compreender que devem manter a sua informação pessoal privada; - ser incentivados a, sempre que possível, criar os seus próprios textos, imagens e sons para os seus projetos; - compreender a necessidade de registar os créditos do material utilizado que não seja da sua autoria.	Os alunos devem: - manter a sua informação pessoal privada; - compreender a necessidade de registar os créditos do material utilizado que não seja da sua autoria; - reconhecer os comentários aos seus projetos como contributos para o seu desenvolvimento; - comentar produtos de forma responsável e construtiva.
Partilha	Os alunos devem: - compreender a importância de apresentar e partilhar os produtos desenvolvidos em grupo, com os seus colegas e numa comunidade em linha; - conhecer e explorar outros projetos em linha.	Os alunos devem: - apresentar e partilhar os produtos desenvolvidos em grupo aos seus colegas e partilhá-los numa comunidade em linha; - compreender a necessidade de indicar as instruções e o modo de funcionamento dos produtos desenvolvidos.

Tabela 2 – Conteúdos e descritores

3. Metodologias propostas de acordo com os principais objetivos

A metodologia utilizada nas sessões de trabalho com alunos dependerá, em larga medida, das condições materiais e humanas existentes no agrupamento/escola. Importa, no entanto, não perder de vista os objetivos gerais do projeto. Depois de uma adaptação ao *software* utilizado, os alunos deverão, tão cedo quanto possível, realizar os seus próprios projetos, apelando à criatividade e à imaginação. Deverá também ser privilegiada a articulação com os conteúdos que os alunos estejam a estudar e o envolvimento do professor titular de turma no acompanhamento dos projetos elaborados, caso não seja o responsável pelo desenvolvimento deste projeto, de forma a contribuir com a articulação referida. Assim, consideramos adequado:

1) **Que sejam desenvolvidos projetos em articulação com as diferentes componentes do currículo.**

Os trabalhos devem ser desenvolvidos integrando, tanto quanto possível, os temas e conteúdos das diferentes componentes do currículo do 1.º Ciclo do Ensino Básico, bem como das áreas transversais.

2) **Que os alunos trabalhem em grupo na construção dos seus próprios projetos.**

Os alunos devem ter liberdade para escolher os temas a abordar e os aspetos a incluir nos seus projetos. Nesse sentido, o trabalho em grupo poderá ser fundamental para que entre os elementos do grupo haja uma primeira discussão e tentativa de resolução de problemas que eventualmente surjam. Os grupos devem ter preferencialmente dois alunos, mas podem incluir mais elementos, aconselhando-se neste caso a agruparem-se em subgrupos com uma divisão clara do trabalho atribuído a cada um dos subgrupos. Cada aluno ou grupo de alunos poderá contribuir nas vertentes em que se sentir mais vocacionado (desenhando, gravando sons, programando, etc.), de modo a todos os elementos contribuírem para o produto final.

3) Que seja incentivada a criatividade e a diversidade de projetos.

Embora numa primeira fase de adaptação dos alunos à(s) linguagem(s) de programação utilizada(s) sejam propostas tarefas exemplificativas que permitam aprender a trabalhar com o *software*, será importante que cada grupo defina os projetos em que deseja trabalhar dando liberdade para os alunos serem criativos e colocarem em prática as suas ideias, mesmo que pareçam, à partida, ir além do que tecnicamente já foi abordado nas sessões. Os alunos devem, no entanto, perceber as limitações do *software* utilizado, cabendo ao professor a tarefa de os aconselhar de modo a que os projetos sejam passíveis de finalização em tempo útil. No caso de projetos muito ambiciosos, os alunos devem ser incentivados a criar versões mais simples que serão posteriormente completadas/melhoradas, dando origem a novas versões dos projetos.

Os alunos devem ainda ser incentivados a registar em papel as suas ideias antes de passar à parte da programação. Deverão planificar, escrever diálogos, regras de jogo, definir personagens, entre outros, antes de passarem à fase de programação. Depois de planificado o projeto, devem testá-lo, pedindo a ajuda de colegas, refletir sobre eventuais críticas e comentários, apresentar e partilhar o projeto e, se sentirem necessidade, melhorá-lo, desenvolvendo novas versões.

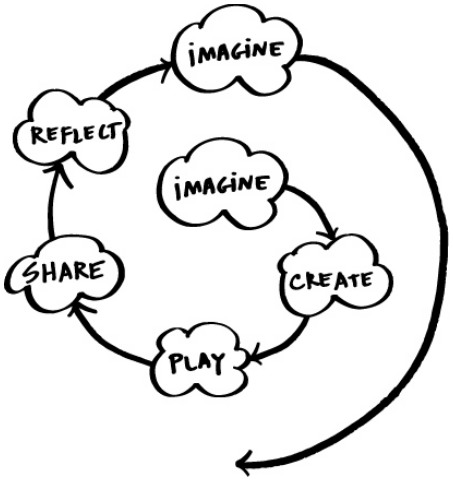


Figura 1 - Espiral de pensamento criativo proposta por Resnick (2007)

4) Que os alunos partilhem os projetos elaborados.

Os alunos devem ter oportunidade de partilhar os seus projetos, explicando o modo como os elaboraram e as ideias subjacentes à sua conceção. Ao longo das sessões, os alunos devem tornar públicos os seus projetos, apresentando-os a outros.

4. Avaliação

Embora o projeto **Iniciação à Programação no 1.º Ciclo do Ensino Básico** possa ser implementado numa grande diversidade de cenários, é importante que sejam pensados à partida os mecanismos de avaliação dos alunos de modo a respeitar as opções globais definidas para cada agrupamento/escola em que seja implementado. Sugerimos que se privilegie uma avaliação baseada em evidências recolhidas ao longo das sessões em que os alunos desenvolvem os seus projetos. Para tal podem ser utilizadas grelhas de observação, tomadas notas, bem como recolhidos registos ao longo das sessões que permitam ficar com evidências do processo de criação dos projetos por parte dos alunos. Os registos áudio/foto/vídeo deverão ser, obviamente, autorizados pelos encarregados de educação a quem deve ser comunicada a finalidade e o âmbito em que serão utilizados. Outros elementos a ter em linha de conta serão os produtos elaborados pelos alunos e o modo como os apresentam e documentam. Deverá ficar claro, no entanto, que, mais que o produto, interessa o processo e o modo como cada grupo trabalhou para chegar a esse produto.

Em anexo reproduzem-se grelhas propostas por Brenann, Balsch, e Chung (2014), traduzidas por Ramos e Espadeiro (2015), que podem servir de exemplo/inspiração e serem adaptadas à realidade de cada escola/projeto.

Não consideramos apropriada a utilização de testes práticos ou teóricos bem como fichas escritas individuais como forma de avaliação dos conhecimentos no âmbito deste projeto, uma vez que mais do que a memorização de comandos e outras formas que estes testes poderiam medir se pretende aferir também a criatividade, o empenho e a compreensão de um modo geral dos conceitos subjacentes ao pensamento computacional.

Referências

Berry, M. (2013). *The computing curriculum beyond 2014*. Disponível em: http://www.slideshare.net/mgberry/the-computing-curriculum-beyond-2014. Acedido em abril de 2015.

Brennan, K., Balch, C. & Chung, M. (2014) *Creative computing.* Harvard Grauate School of Education. Disponível em: http://scratched.gse.harvard.edu/guide/files/CreativeComputing20141015.pdf. Acedido em agosto de 2015.

Brennan, K., Chung, M., & Hawson, J. (2011). *Computação criativa – uma introdução ao pensamento computacional baseada no conceito de design*. (Tradução de Teresa Marques), Disponível em http://projectos.ese.ips.pt/cctic/wp-content/uploads/2011/10/Guia-Curricular-ScratchMIT-EduScratchLPpdf.pdf. Acedido em março de 2015.

Cobo, C. (2014). *Experiencia del caso inglés en la integración de tic y la definición de estándares de habilidades TIC para docentes* (1997-2013). Universidad de Oxford. Disponível em: https://www.acade.edu/9038842/. Acedido em abril de 2015.

Computing at School (2102). *Computer Science: A Curriculum for Schools*. Disponível em http://www.computingatschool.org.uk/data/uploads/ComputingCurric.pdf. Consultado em abril de 2015.

P21's Framework for 21st Century Learning. (2015). Disponível em: http://www.p21.org/storage/documents/docs/P21_Framework_Definitions_New_Logo_2015.pdf. Acedido em agosto de 2015.

Ramos, J.L. & Espadeiro, R.G. (2015) *Pensamento computacional na escola e práticas de avaliação das aprendizagens. Uma revisão sistemática da literatura*. Disponível em: http://dspace.uevora.pt/rdpc/bitstream/10174/14227/1/challenges%202015br.pdf. Acedido em agosto 2015.

Ramos, J.L. & Espadeiro, R.G. (2014) Os futuros professores e os professores do futuro. Os desafios da introdução ao pensamento computacional na escola, no currículo e na aprendizagem. *Revista de Educação, Formação e Tecnologia*., 7(2), 4-25, Disponível em: http://eft.educom.pt/index.php/eft. Acedido em agosto 2015.

Resnick, M. (2007). Sowing the Seeds for a More Creative Society. *Learning and Leading with Technology*. Disponível em http://web.media.mit.edu/~mres/papers/Learning-Leading-final.pdf. Acedido em abril de 2015.

Resnick, M. (2014). *Transformar 2014 - Debate com Mitchel Resnick*. Disponível em https://www.youtube.com/watch?t=395&v=A95XkiJjcaM. Acedido em abril de 2015.

Wing, J. M. (2006). *Computational thinking*. CACM, 49(3), 33-35. Disponível em: http://www.cs.cmu.edu/~wing/publications/Wing06.pdf. Acedido em abril 2015.

Anexo

Exemplo de grelhas de avaliação propostas por Brenann (2014), traduzidas por Ramos (2015)

EXPERIMENTAR E INTERAGIR	BAIXO	MÉDIO	ALTO - ELEVADO
Descreve a construção do projeto, passo a passo.	Fornece uma descrição elementar da construção do projeto, mas não detalha aspetos específicos do mesmo.	Faz uma descrição genérica do projeto, de forma ordenada.	Fornece detalhes sobre as diferentes componentes dum projeto específico e descreve o modo como foram desenvolvidos, de forma ordenada.
Enumera exemplos que experimentou ao longo da elaboração do projeto?	Não apresenta exemplos específicos do que experimentou.	Deixa transparecer de forma genérica que experimentou outras coisas no projeto.	Fornece exemplos específicos de outras coisas que foi experimentando no projeto.
Apresenta as revisões que foram sendo feitas.	Afirma não ter feito revisões, ou afirma ter feito algumas mas não exemplifica.	Descreve uma revisão específica que fez ao projeto.	Descreve aspetos específicos de coisas que acrescentou ao projeto e justifica.
Descreve as diferentes abordagens que experimentou no projeto, ou quando tentou fazer algo novo.	Não revela evidências de ter experimentado algo novo.	Fornece um exemplo de algo novo que experimentou no projeto.	Descreve com detalhe coisas novas que experimentou no projeto.

TESTAR E CORRIGIR	BAIXO	MÉDIO	ALTO - ELEVADO
Descreve o que aconteceu com o projeto de diferente em relação ao pretendido.	Não descreve o que resultou diferente em relação ao pretendido.	Descreve o que correu mal no projeto, mas não o que pretendia fazer.	Dá um exemplo detalhado do que aconteceu e o que pretendia, quando executa o programa.
Descreve de que forma foi feita a leitura do código para encontrar a causa do problema.	Não descreve um problema.	Descreve como faz a leitura mas não apresenta um exemplo específico de encontrar um problema no código.	Descreve como faz a leitura e apresenta um exemplo específico de encontrar um problema no código
Descreve que alterações fez e como as testou para verificar os resultados.	Não descreve que problemas teve ou a solução	Fornece um exemplo genérico sobre as alterações feitas e os testes feitos para verificar o funcionamento.	Fornece um exemplo específico sobre as alterações feitas e os testes feitos para verificar o funcionamento.
Descreve outras formas de resolver o problema.	Não apresenta uma forma para encontrar uma solução para o problema.	Apresenta uma forma genérica para encontrar uma solução para o problema.	Apresenta um exemplo específico de como encontrar uma solução para o problema.

REUTILIZAR E RECOMBINAR	BAIXO	MÉDIO	ALTO
Descreve se encontrou inspiração em outros projetos e na leitura do código disponível.	Descreve como desenvolveu as ideias ou em que projetos se inspirou.	Fornece uma descrição geral de um projeto que o inspirou.	Dá um exemplo específico do projeto que o/a inspirou.
Descreve como selecionou uma parte de outro projeto e como a adaptou ao seu.	Não descreve como adaptou as ideias, *scripts* ou recursos de outros projetos.	Identifica *scripts*, ideias ou recursos que adaptou de outros projetos.	Fornece exemplos específicos de *scripts*, ideias ou recursos que ele adaptou de outros projetos e como.
Como refere/cita as pessoas cujo trabalho inspirou o seu.	Não identifica as fontes e os autores em que se inspirou para a realização do seu projeto.	Identifica as fontes e os autores em que se inspirou para a realização do seu projeto.	Documenta no projeto as fontes e os autores que o inspiraram.

ABSTRAIR E MODULARIZAR	BAIXO	MÉDIO	ALTO
Como foi decidido que *sprites* eram necessários para o projeto e onde eram utilizados.	Não descreve que *sprites* foram selecionados.	Fornece uma descrição geral da decisão de escolher certos *sprites.*	Dá uma explicação detalhada acerca de como selecionou os *sprites* em função do objetivo do projeto.
Como foi decidido que *scripts* eram necessários para o projeto e onde eram utilizados.	Não descreve que *scripts* foram criados.	Fornece uma descrição geral da decisão de criar certos *scripts*.	Dá uma explicação detalhada acerca de como criou os *scripts* em função do objetivo do projeto.
Como foram organizados os *scripts* de forma a terem significado para o estudante e para os outros.	Não descreve como os *scripts* foram organizados.	Fornece uma descrição geral da forma como foram organizados *scripts*.	Dá uma explicação detalhada acerca de como organizou os *scripts* e porquê.

AUTORIA:

Miguel Figueiredo

João Torres

REVISÃO:

António Silva

Fátima Mendes

Helena Romano

José Duarte

Laura Filipe

Mário Baía

Rosário Rodrigues

Teresa Marques

APOIO:

GOVERNO DE PORTUGAL | MINISTÉRIO DA EDUCAÇÃO E CIÊNCIA

Direção-Geral da Educação ■ Av. 24 de Julho, 140 - 1399-025 Lisboa - Portugal
Telef.: +(351) 21 393 45 00 - Fax: +(351) 21 393 46 95 ■ Email: dge@dge.mec.pt Internet: www.dge.mec.pt