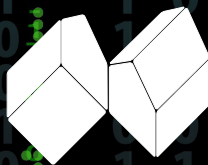


Pensamento Computacional

Com introdução à programação em Python 3.x

Apoio aos professores do ensino básico de acordo
com as novas aprendizagens essenciais

JOÃO NUNO TAVARES



Pensamento Computacional

Com introdução à programação em *Python 3.x*

Apoio aos professores do ensino básico e secundário de acordo com as novas aprendizagens essenciais

João Nuno Tavares

Revisão científica:

João Pedro Pedroso (Universidade do Porto)

REVISTA DE CIÊNCIA ELEMENTAR

FICHA TÉCNICA

Rev. Ciência Elem., V11(A)

**Publicação trimestral
da Casa das Ciências**

ISSN 2183-9697 (versão impressa)

ISSN 2183-1270 (versão online)

rce.casadasciencias.org

DEPÓSITO LEGAL

452636/19

TÍTULO

Pensamento Computacional

AUTOR

João Nuno Tavares

REVISÃO CIENTÍFICA

João Pedro Pedroso (UNIVERSIDADE DO PORTO)

IMAGENS

João Nuno Tavares

PAGINAÇÃO

Raul Seabra

IMPRESSÃO E ACABAMENTO

Uniarte Gráfica S.A.

TIRAGEM

250 exemplares

CORPO EDITORIAL DA REVISTA DE CIÊNCIA ELEMENTAR

EDITOR

João Nuno Tavares (UNIVERSIDADE DO PORTO)

CONSELHO EDITORIAL

Alexandre Lopes Magalhães (Universidade do Porto)

Jorge Manuel Canhoto (Universidade de Coimbra)

Paulo Ribeiro-Claro (Universidade de Aveiro)

José Cidade Mourão (Instituto Superior Técnico)

Rute Coimbra (Universidade de Aveiro)

Sónia Gouveia (Universidade de Aveiro)

José Francisco Rodrigues (Universidade de Lisboa)

PRODUÇÃO E SECRETARIADO

Alexandra Coelho

Guilherme Monteiro

Liliana Freitas

Raul Seabra

NORMAS DE PUBLICAÇÃO NA RCE

A Revista de Ciência Elementar dirige-se a um público alargado de professores do ensino básico e secundário, aos estudantes de todos os níveis de ensino e a todos aqueles que se interessam pela Ciência. Discutirá conceitos numa linguagem elementar, mas sempre com um rigor superior.

INFORMAÇÃO PARA AUTORES E REVISORES

Convidam-se todos os professores e investigadores a apresentarem os conceitos básicos do seu labor diário numa linguagem que a generalidade da população possa ler e compreender.

Para mais informação sobre o processo de submissão de artigos, consulte a página da revista em rce.casadasciencias.org

**CASA DAS
CIÊNCIAS**
EDULOG · FUNDAÇÃO BELMIRO DE AZEVEDO

© Todo o material publicado nesta revista pode ser reutilizado para fins não comerciais, desde que a fonte seja citada.



PROPRIETÁRIO

Casa das Ciências/ICETA

Faculdade de Ciências,

Universidade do Porto

Rua do Campo Alegre, 687

4169-007 Porto

rce@casadasciencias.org

PREFÁCIO

Este é um livro de apoio ao Professor do ensino básico e secundário e a todos os que se interessam por matemática feita com a ajuda de um computador, usando a linguagem *Python*. O livro surge numa altura em que existe a intenção, na minha opinião muito louvável, de introduzir este tipo de aprendizagem nos currículos oficiais de ensino básico e secundário. Os temas propostos incidem sobre “Pensamento Computacional”, abordado de um ponto de vista pragmático, com muitos exemplos concretos, que permitem uma abordagem algorítmica, e consequente construção de códigos que, neste livro, são implementados computacionalmente usando a linguagem *Python*.

O livro inclui uma introdução aos principais conceitos de programação em linguagem *Python 3.x* (capítulo 2), logo seguida de um conjunto de tarefas em que se discute a estrutura da construção de algoritmos e sua implementação computacional, num conjunto diversificado de situações e tarefas.

O pensamento computacional consiste de um conjunto de capacidades e práticas, relacionadas entre si, para resolver problemas complexos, se possível, com a ajuda de um computador. Estimula os alunos a desenvolverem competências de pensamento crítico, capacidades de abstração, de sequenciação, de estruturação de ciclos, eventos, modularização, condicionais, processos algorítmicos, uso de dados e escalabilidade, que formam a base do pensamento algorítmico (ou seja, como resolver problemas usando uma progressão estruturada de etapas lógicas).

As crianças aprendem naturalmente algoritmos (por exemplo, a sequência de tarefas que tencionam executar num dia normal de aulas), bem como sistemas lógicos (por exemplo, as regras de um jogo). Como o pensamento computacional é derivado de processos de pensamento humano, pode ser integrado em muitos contextos diferentes. Este livro é constituído por um vasto e diversificado número de tarefas para que os professores do ensino básico e secundário incorporem o pensamento computacional na resolução de problemas concretos. Em vez de descrever conceitos e metodologias sobre pensamento computacional, bem discutidos numa vasta bibliografia, desde os livros pioneiros de Seymour Papert ([1] e [2]) até aos mais recentes ([3], [4], [5]), o livro adopta a atitude pragmática de ilustrar, com tarefas concretas, como pode ser implementado o pensamento computacional, como uma mais-valia na aprendizagem de diferentes estratégias e soluções algorítmicas para a resolução de problemas.

O autor utilizou uma linguagem informal, colocando-se na posição de um professor dirigindo-se aos seus alunos (que trata por tu!). Esta linguagem deverá ser adaptada pelo próprio professor ao perfil dos seus alunos.

O tratamento da linguagem é um ponto essencial no processo de ensino/aprendizagem. Neste livro esta questão mereceu uma particular atenção — não se adotou uma linguagem formal, procurou-se usar um discurso dirigido à intuição e à experiência do quotidiano do aluno, adequado ao seu estado de desenvolvimento cognitivo, mas, por outro lado, procurou evitar-se uma linguagem demasiado infantilizada, com excessivo recurso a figuras ou meios audiovisuais, atrasando o processo de concentração que o aluno deverá adquirir gradualmente, para interiorizar conceitos e ter tempo para os assimilar e para os praticar.

O livro apresenta mais de 50 tarefas, agrupadas por capítulos, e com um grau de dificuldade variável. As tarefas contêm notas explicativas detalhadas sobre a parte conceptual que as sustenta. Houve um cuidado especial, quer na escolha das atividades, procurando garantir diversidade e variedade de situações que sejam familiares aos alunos, quer na justificação conceptual (com um mínimo de formalismo) das mesmas.

Cada tarefa integra e desenvolve um conjunto de competências, implícitas no chamado Pensamento Computacional. Isto é, cada tarefa não desenvolve apenas uma competência específica, mas sim um conjunto delas. Parece-nos que, desta forma, o ensino/aprendizagem pode ser mais eficaz quando, partindo de um mesmo objeto de estudo (um problema particular), se desenvolve uma parte substancial dos conceitos e competências previstas. Esta “fórmula” é replicada para cada tarefa. Desta forma incentiva-se a que o aluno apreenda os conceitos em diferentes situações e identifique as analogias inerentes.

Em várias tarefas propõem-se questões e problemas para o aluno discutir com colegas e/ou professor(a), ou tentar resolver individualmente. Para todos os problemas são apresentadas as respetivas soluções.

Sendo essencialmente um livro de apoio ao professor, compete a este selecionar as tarefas que ache mais adequadas e, eventualmente, com base nas tarefas aqui apresentadas, criar outras análogas (ou não) que sejam necessárias a um ritmo de ensino/aprendizagem mais adequado ao perfil de alunos com que lida.

A visualização é um aspeto muito importante para permitir uma comunicação e compreensão mais rápida dos resultados. Por isso, a ilustração gráfica mereceu um cuidado muito especial da nossa parte. Espera-se que seja suficientemente motivadora e esclarecedora, evitando cair num excesso figurativo que, na nossa opinião, como já referimos, tende a infantilizar a aprendizagem, desviando-a dos momentos de reflexão e assimilação dos conceitos.

Todas as atividades exigem a participação ativa dos alunos, sob supervisão do professor, é claro! A participação ativa dos alunos, individual ou em grupo, deve ser estimulada, bem como a discussão das questões propostas e respetivos resultados.

Uma última nota sobre a escolha da linguagem de programação *Python*. Na minha opinião, esta linguagem pode começar a ser utilizada, com os devidos cuidados, a partir do 7.º ano de escolaridade, ou eventualmente mais cedo, com muitas vantagens sobre o *Scratch*, por exemplo, esta mais apropriada para alunos de 1.º ciclo. Vale a pena vencer eventuais

obstáculos ou resistências ao uso desta linguagem — na verdade uma das mais utilizadas atualmente, mesmo em computação de elevada complexidade. Quisemos manter o uso do *Python* a um nível perfeitamente elementar e, por isso, não usamos as bibliotecas usuais do *Python*, com exceção das bibliotecas *numpy*, *matplotlib* e *random* em algumas (poucas) tarefas que são explicadas com todo o detalhe. Também não usamos a biblioteca *Turtle Geometry*, nem *Processing*, para o desenvolvimento de aplicações à Geometria e à chamada Arte Generativa. Tencionamos fazê-lo num próximo livro, dedicado ao ensino da Geometria.

Todos os códigos incluídos neste livro, foram construídos usando o IDLE (Ambiente Integrado de Desenvolvimento e Aprendizagem) *Thonny*, que nos pareceu o mais simples, e visualmente o mais atraente, para o uso dos alunos. **Todos os códigos foram testados, e estão disponíveis no portal da Casa das Ciências, seguindo as instruções lá indicadas para fazer a respetiva transferência.** Os códigos são uma componente essencial deste livro. O leitor poderá modificá-los, alterar os parâmetros e assim fazer experiências computacionais sobre as tarefas propostas. Poderá inclusive utilizá-los para construir outros códigos e outras situações.

Como todo o livro, este não estará isento de erros ou gralhas, embora tivesse sido revisto várias vezes. Agradeço, desde já, que me façam chegar, através do e-mail: jntavar@fc.up.pt, quaisquer críticas, dúvidas e sugestões de correção/alteração de conteúdos, que possam melhorá-lo. Procurarei responder a todos com a máxima brevidade.

Agradeço muito ao revisor científico, Professor João Pedro Pedroso, da FCUP, cujos comentários e correções muito valorizaram este livro.

Agradece-se ainda à Fundação Manuel António da Mota por ter permitido concretizar este projeto editorial.

Posto isto, resta-me desejar que este livro seja muito útil para os professores, e outros leitores interessados, e que seja um bom início para a efetiva inclusão do pensamento computacional nos currículos de ensino básico e secundário, como um paradigma essencial para uma educação do século XXI.

Desejo a todos um BOM TRABALHO.

João Nuno Tavares

Coordenador da Casa das Ciências

O MECENAS

A Fundação Manuel António da Mota (FMAM) foi constituída em 2009 e oficialmente reconhecida em 2010.

Constitui o contemporâneo e natural corolário da matriz e tradição filantrópicas do Grupo Mota-Engil, na senda do legado do seu fundador, Manuel António da Mota.

Manuel António da Mota foi um empreendedor e filantropo que se superou a si próprio.

Apesar de todas as adversidades, através da sua atividade empresarial ou por iniciativa própria, contribuiu sempre para o bem comum, afirmando os valores que a FMAM e a família Mota preservam até hoje.

Não tendo ido além do ensino primário, mas dotado de forte determinação e ousadia, fundou em 1946 a Mota & Companhia, uma empresa com atividade centrada no setor da Engenharia e Construção, e que mais tarde deu origem ao Grupo Mota-Engil.

O Grupo Mota-Engil, liderado pela família Mota, integra hoje um conjunto alargado e multisetorial de negócios, englobando as áreas da Engenharia e Construção, Gestão de Resíduos, Energia, Multisserviços, Concessões de Infraestruturas de Transportes, Mineração e Logística. Presente na Europa, Américas e África através das suas sucursais e empresas participadas espalhadas pelo mundo, o seu volume de negócios cifra-se em mais de 3,8 mil milhões de euros, contando nas suas fileiras com milhares de colaboradores.

Criada em homenagem a Manuel António da Mota, a Fundação é um importante instrumento da política de responsabilidade social do Grupo Mota-Engil, enquanto expressão organizada e sistematizada de uma gestão ética e socialmente comprometida, em nome de uma cidadania empresarial ativa e participativa.

A Fundação, sediada na cidade do Porto, tem por fins estatutários a promoção, desenvolvimento e apoio a iniciativas de natureza social nos domínios da beneficência e solidariedade social, e de natureza cultural nos domínios da educação, saúde, ambiente, organização e apoio à atividade artística, exercendo a sua atividade em todo o território nacional e nos países onde o Grupo Mota-Engil marca presença.

Institui ainda anualmente o “Prémio Manuel António da Mota”.



MECENAS



MOTA-ENGIL

ÍNDICE

Contextualização. Novas Aprendizagens Essenciais	11
Capítulo 1 - Tarefas sem programação	
Resumo	19
Tarefa 1 - Fazer um bolo.....	21
Tarefa 2 - Calcular a área de uma figura complicada.....	24
Tarefa 3 - Como construir uma casa	27
Tarefa 4 - Organizar o teu dia	29
Tarefa 5 - Uma versão do jogo de <i>Sudoku</i>	31
Tarefa 6 - Percorrer um labirinto.....	33
Tarefa 7 - Colorir mapas.....	35
Tarefa 8 - Atravessar o rio.....	39
Tarefa 9 - Um jogo simples.....	43
Tarefa 10 - O jogo das 11 bolinhas	47
Tarefa 11 - O jogo dos 3 anões e do gigante	50
Tarefa 12 - Caminho mais curto.....	52
Tarefa 13 - Problema das duas torneiras	56
Tarefa 14 - O jogo dos saltinhos	58
Tarefa 15 - Quadrados mágicos	64
Tarefa 16 - Um labirinto e o grafo associado	67
Tarefa 17 - Como somar frações.....	71
Tarefa 18 - Alguma matemática do <i>Facebook</i>	73
Capítulo 2 - Introdução à programação em <i>Python</i>	
Resumo	77
Tarefa 1 - Introdução	79
Tarefa 2 - O que é uma variável? O que é o valor de uma variável?.....	80
Tarefa 3 - O IDLE (Ambiente Integrado de Desenvolvimento e Aprendizagem) <i>Thonny</i>	81
Tarefa 4 - Operações aritméticas elementares	87
Tarefa 5 - Operações aritméticas em <i>Python</i>	90
Tarefa 6 - Decisões. Condicionais <i>if</i> , <i>else</i> e <i>elif</i>	91
Tarefa 7 - Decisões baseadas em várias condições.....	96
Tarefa 8 - Comparando dois valores.....	99
Tarefa 9 - Ciclos (<i>loops</i>) <i>for</i> e <i>while</i>	100
Tarefa 10 - Listas. Operações em listas	105
Tarefa 11 - <i>Strings</i>	108

Capítulo 3 - Tarefas com programação em *Python*

Resumo	109
Tarefa 1 - Progressões geométrica e aritmética	111
Tarefa 2 - Resolvendo um problema difícil	114
Tarefa 3 - Adivinha um número	118
Tarefa 4 - Soma dos primeiros N inteiros positivos.....	120
Tarefa 5 - Números Poligonais	126
Tarefa 6 - Números de Fibonacci.....	132
Tarefa 7 - Fatores de um número inteiro positivo N	135
Tarefa 8 - Calcular o Máximo Divisor Comum, MDC, (a, b) e o mínimo múltiplo comum, mmc, (a, b) de dois números inteiros positivos dados a e b	137
Tarefa 9 - Verificar se um certo número é primo.....	140
Tarefa 10 - Gerando primos com o crivo de Eratóstenes	141
Tarefa 11 - Calcular os fatores primos de um número.....	144
Tarefa 12 - Sistema decimal e sistema binário	147
Tarefa 13 - Cifras	150
Tarefa 14 - Aproximação ao valor de π , imaginada por Arquimedes	154
Tarefa 15 - Algoritmos para ordenação de uma lista	159
Tarefa 16 - Simulação de um jogo com dois dados.....	169
Tarefa 17 - Moda, média e mediana.....	177
Tarefa 18 - Triplos Pitagóricos.....	185
Tarefa 19 - Modelos de crescimento	189
Tarefa 20 - Método de Bissecção.....	200
Tarefa 21 - Modelo de Predador e Presa	208
Tarefa 22 - Inverter a ordem dos dígitos de um número.....	211
Tarefa 23 - Soma dos algarismos de um número inteiro positivo	215
Tarefa 24 - Conjetura de Ulam-Collatz	216
Tarefa 25 - Palíndromos.....	222
Tarefa 26 - Gerando tiras (<i>strings</i>) de ADN	224
Tarefa 27 - Duas aplicações à Matemática Financeira.....	228

Capítulo 4 - Projeto. Modelo SIR

Resumo	233
Bibliografia.....	255

Contextualização

Novas Aprendizagens Essenciais

Excerto do texto “Novas aprendizagens essenciais”

A arte de resolver problemas — Método de Polya

Método de Computação de Polya

Excerto das “Novas aprendizagens essenciais”

Pensamento Computacional

Desenvolver e mobilizar o pensamento computacional, capacidade que tem vindo a assumir relevância nos currículos de Matemática de diversos países. O pensamento computacional pressupõe o desenvolvimento, de forma integrada, de práticas como a abstração, a decomposição, o reconhecimento de padrões, a análise e definição de algoritmos, e o desenvolvimento de hábitos de depuração e otimização dos processos. Estas práticas são imprescindíveis na atividade matemática e dotam os alunos de ferramentas que lhes permitem resolver problemas, em especial relacionados com a programação.

Decomposição — Estruturar a resolução de problemas por etapas de menor complexidade de modo a reduzir a dificuldade do problema

Incentivar a identificação de elementos importantes e a sua ordenação na execução de uma tarefa, criando oportunidades para os alunos decomporem a tarefa em partes mais simples, diminuindo desta forma a sua complexidade.

Abstração — Extrair a informação essencial de um problema

Criar oportunidades para que os alunos representem problemas de forma simplificada, concentrando-se na informação mais importante. Realçar processos relevantes e secundarizar detalhes e especificidades particulares. Assim, os alunos devem centrar a atenção no objeto a atingir, considerar os obstáculos e desconsiderar todos os outros objetos.

Reconhecimento de padrões — Reconhecer ou identificar padrões no processo de resolução de um problema e aplicar os que se revelam eficazes na resolução de outros problemas semelhantes

Incentivar a identificação de padrões durante a resolução de problemas, solicitando que os alunos os descrevam e realizem previsões com base nos padrões identificados. Incentivar a procura de semelhanças e a identificação de padrões comuns a outros problemas já resolvidos de modo a aplicar, a um problema em resolução, os processos que anteriormente se tenham revelado úteis.

Algoritmia — Desenvolver um procedimento passo a passo (algoritmo) para solucionar um problema de modo a que este possa ser implementado em recursos tecnológicos, sem necessariamente o ser

Promover o desenvolvimento de práticas que visem estruturar, passo a passo, o processo de resolução de um problema, incentivando os alunos a criarem algoritmos que possam descrever essas etapas nomeadamente com recurso à tecnologia, promovendo a criatividade e valorizando uma diversidade de resoluções e representações que favoreçam a inclusão de todos (Exemplo: Na exploração de jogos que envolvam relações numéricas e as propriedades das operações, conduzir os alunos a definirem o algoritmo (sequência de instruções passo a passo)