Aprendizagens (e competências) matemáticas com TI-Python

O PENSAMENTO COMPUTACIONAL, A PROGRAMAÇÃO, A TECNOLOGIA DIGITAL E A MATEMÁTICA

As competências digitais assumem cada vez mais um papel central na sociedade e inevitavelmente na educação. O nosso sistema educativo reflete esta tendência através de um número cada vez maior de iniciativas, decorrentes sobretudo do Perfil dos Alunos à Saída da Escolaridade Obrigatória, mas enquadradas com uma perspetiva internacional, no mínimo europeia. No entanto, estas iniciativas não visam, ainda, uma integração curricular plena, sobretudo quando estas competências se constroem associadas ao desenvolvimento do pensamento computacional e consequentemente à programação. Não falamos da mesma coisa quando nos referimos ao Pensamento Computacional e Programação, havendo cada vez mais informação, assente em trabalhos de investigação, que destrinça estes dois conceitos. O pensamento computacional, conceito que tem evoluído, está mais relacionado com a resolução de problemas, suscetíveis de serem traduzidos numa linguagem de programação, mas não necessariamente (Bocconi, 2016). Neste sentido, ganha maior força a integração curricular do pensamento computacional, nomeadamente numa disciplina como a matemática, e não é alheio a isto o facto de terem surgido nas Aprendizagens Essenciais de Matemática A (agosto de 2018) referências explícitas à algoritmia e à programação, mas também na disciplina de Matemática Aplicada às Ciências Sociais. Esta integração ocorre na disciplina de opção do 12.º ano, Aplicações Informáticas B, sendo por isso muito limitada a sua abrangência.

Portanto, têm sido criadas condições teóricas para que o ensino e aprendizagem nas nossas escolas se possa alterar e evoluir no sentido do plasmado no Quadro Europeu de Competência Digital para Educadores (DigCompEdu), que aponta e descreve um significativo número de competências digitais que se consideram essenciais para que os professores possam melhor potenciar o digital ao serviço das aprendizagens dos seus alunos (Lucas, 2018). Mas, apesar de se ter aberto esta janela de oportunidade, o que a prática observada nos indica, salvo raras exceções, é que as Aprendizagens Essenciais serviram essencialmente para balizar o que estava consagrado nos programas e era obrigatório lecionar, em particular nas disciplinas sujeitas a Exame/Prova Nacional, e a Autonomia e Flexibilização Curricular serviu para reduzir a ansiedade com a escassez de tempo para se lecionar e esporadicamente se realizarem um ou outro trabalho transdisciplinar. Parece-nos que faltou mais orientação, mais formação e mais objetividade na definição de trilhos a seguir, para além de se absorver os professores com muitas e simultâneas alterações ao nível da política educativa. Por isso, a oportunidade aberta parece não ter ainda vingado. Com as "Recomendações para a melhoria das aprendizagens dos alunos em Matemática", de 27 de março de 2020, que certamente irão balizar os futuros programas da disciplina de matemática, abre-se uma nova janela de oportunidade. Está assim reunida, mais do que nunca, a possibilidade da abordagem à algoritmia e programação deixar a timidez que se tem notado, o que é de todo desejável. Será, também, uma oportunidade de se avançar corajosamente de modo a reduzir substancialmente o atraso em relação a outros sistemas educativos, como o francês, até porque está em jogo a competitividade económica entre países.

Os futuros programas de matemática nos dirão se a relevância desta temática será acomodada, mas é legítimo o receio de que continuemos com uma abordagem demasiado tímida pela forma como tal surge na Recomendação 2 (pág. 293).

Não podemos terminar esta abordagem inicial sem referir que não se pode pretender que se ensine programação, mas sim que a programação seja uma ferramenta para aprendizagens matemáticas mais significativas e mais ricas. Também nos parece que a programação não deverá ser uma ferramenta exclusiva às aprendizagens matemáticas, mas sim uma ferramenta transversal a outras as áreas curriculares e não curriculares, obviamente com níveis e estratégias adequadas. Para desenvolver essa ferramenta é necessário recorrer a uma linguagem de programação, sendo o Python aquela que melhor se ajustará aos alunos do ensino secundário.

LINGUAGEM DE PROGRAMAÇÃO – PORQUÊ O PYTHON?

Tomemos como exemplo o que ocorre em França, em que durante o College, período que corresponde ao 3.º Ciclo do ensino básico português, os alunos trabalham algoritmia e programação, de forma obrigatória desde 2016, utilizando Scratch, uma linguagem de programação por blocos. Esta linguagem é também sugerida em diversas oportunidades, não obrigatórias, no nosso sistema educativo no ensino básico, em particular nas experiências piloto desenvolvidas pela Equipa de Recursos e Tecnologias Educativas - ERTE, da DGE em 2015, 2016 e 2017, e posteriormente avaliadas (Ramos, 2016). Em 2019, a algoritmia e a programação surgem no ensino secundário francês, com passagem para uma linguagem em texto, sendo obrigatoriamente o Python.

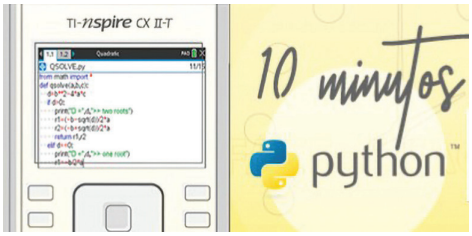
Mas porquê o Python? O que levará nos dias de hoje à opção pela utilização obrigatória desta linguagem de programação?

O Python é uma linguagem concisa, de sintaxe simples e clara, o que facilita a transição de uma linguagem por blocos para uma linguagem de texto. Por outro lado, há uma grande comunidade de utilizadores, em particular no contexto educacional, e uma grande riqueza de recursos disponíveis. Além disso, é *open source*, portanto gratuita, e funciona em vários sistemas operativos. A linguagem Python possuí ainda uma enorme quantidade de bibliotecas disponíveis, que podem ser úteis à matemática, mas também a outras áreas disciplinares, sendo esta a via da continuidade da sua evolução. Note-se que, em França, esta linguagem é referida nos programas de Física e Química do ensino secundário, sendo sugerida a sua aplicação em muitos dos temas tratados. Reforçamos que, a simplicidade da linguagem permitirá de certa forma desviar o foco do desenvolvimento de atividades de algoritmia e programação, diminuindo-se dificuldades acrescidas pela necessidade de aprendizagem da linguagem, mantendo o objetivo na aprendizagem matemática, no processo de resolução e decomposição de problemas, no pensamento criativo e reflexivo e na competência digital. A prioridade deve ser o ensino da matemática, não a programação e a algoritmia. Os detalhes técnicos não devem ser privilegiados como critério de avaliação da compreensão dos alunos, como é aliás defendido nos programas franceses.

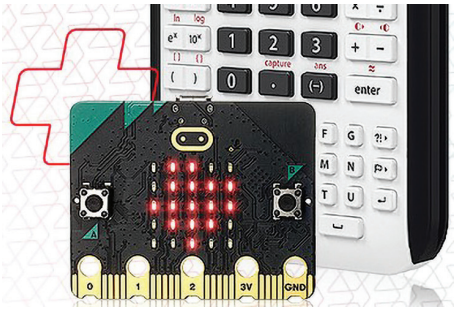
O criador desta linguagem de programação, em 1991, foi o holandês Guido van Rossum, a qual evoluiu de tal forma que hoje é já a linguagem de programação mais utilizada em todo o mundo. Ao contrário do que possa parecer, o nome da linguagem deve-se ao famoso programa de televisão "Monthy Python's Flying Circus" e não à cobra com igual nome, a qual apareceu numa publicação da O'Reilly, editora que habitualmente coloca animais na capa dos livros que edita. No entanto, é a imagem da cobra pitão que prevalece no logotipo do Python.

Segundo o seu criador, numa entrevista publicada no sítio do T3 Europa (da Texas Instruments Educacional) em março de 2020, o poder desta linguagem reside no facto do seu código ser curto e muito literal, de fácil leitura, tornando-se adequada para programadores que estejam a iniciar e mais avançados. Diz ainda que, esta linguagem de programação é muito intuitiva, sendo por isso uma das melhores opções para o ensino da programação no Ensino Secundário, em que os alunos podem adquirir competências essenciais para a vida moderna através da sua utilização, aprendendo ao mesmo tempo Matemática. Embora Python seja fácil de aprender, isso não significa que seja uma linguagem "básica", por exemplo, todas as empresas de tecnologia de Silicon Valley, a capital global da tecnologia, utilizam Python em pequena e em grande escala.

Neste sentido, procurando proporcionar aos alunos do Ensino Secundário a oportunidade de trabalho com esta linguagem de programação, as principais empresas que produzem calculadoras gráficas estão a aderir à sua integração nesses dispositivos. A Texas Instruments adicionou uma nova aplicação às suas calculadoras gráficas, a aplicação TI-Python, mantendo a sua anterior aplicação de programação básica, que já denotava uma visão do interesse da programação nas "mãos" dos alunos. Desta forma, procura-se mais uma vez criar as condições técnicas para que os alunos tenham a oportunidade de explorar a tecnologia portátil que, infelizmente, ainda é muito vista e usada como apenas uma calculadora gráfica. Neste sentido, a Texas Instruments disponibiliza, ainda, um conjunto de módulos iniciais de introdução à programação com Python, em contexto educativo, através da sua iniciativa "10 minutos de Código com Python" disponível em https://education.ti.com/pt/atividades/ti-codes/python/nspire.



Sendo esta uma linguagem aberta, será possível acrescentar novos módulos/bibliotecas à aplicação TI-Python procurando colocar ao dispor dos professores e dos alunos ferramentas que permitam integrar maior número de artefactos digitais/tecnológicos, o último dos quais foi a placa BBC micro:bit, excelente para a realização de projetos STEAM. Esta placa, programável em Python, começou a ser distribuída gratuitamente a todos os estudantes dos países do Reino Unido em 2016, com programas educacionais desenvolvidos pela BBC e com resultados comprovados



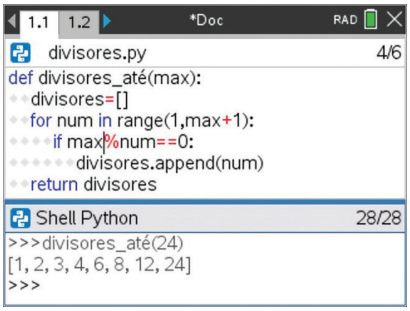
APRENDIZAGENS (E COMPETÊNCIAS) MATEMÁTICAS COM O TI-PYTHON

Nos exemplos simples que apresentamos de seguida reforçamos a importância da necessidade do conhecimento matemático para a construção do algoritmo e do programa, isto é, a programação requer a mobilização e reforça os conhecimentos matemáticos. Quando proposto ao aluno como projeto, de preferência em trabalho colaborativo com colegas, a tarefa de planear e criar a estrutura do programa possibilita o desenvolvimento de competências, como por exemplo, entre muitas outras, o simples testar, descobrir o erro e retificar. Além disso, é importante referir que o DigCompEdu, já citado, engloba três grandes competências, sendo uma delas a dos aprendentes. Deste

modo, estará a contribuir-se para a promoção da competência digital dos aprendentes, não só nos aspetos da literacia digital, mas sobretudo na resolução de problemas, comunicação e colaboração.

Divisores de um número natural

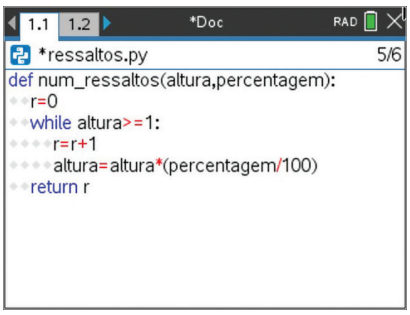
No ecrã da TI-Nspire CX II que se segue, pode observar-se um exemplo de programação associada à matemática, onde se pretende apresentar uma lista dos divisores de um dado número natural.



Este exemplo, com conteúdos do ensino básico, poderá ser explorado no ensino secundário como revisão de um conteúdo e uma primeira experiência na linguagem Python. Permitirá ao aluno conhecer alguns dos operadores algébricos do Python e ciclos repetitivos.

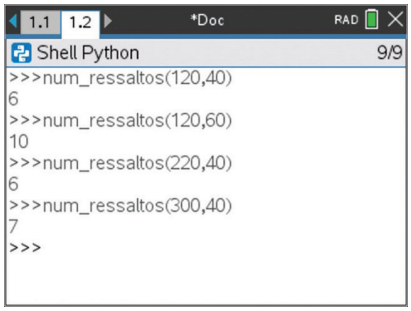
Ressaltos de uma bola

Neste exemplo, adaptado do que consta da publicação, "10 minutos de código com Python", pode observar-se um contexto com maior aproximação à Física. Pretende-se obter o número de ressaltos realizados por uma bola até que esta atinja uma altura estritamente inferior a 1 cm, sabendo que a bola é largada de uma determinada altura (em centímetros) e que ressalta no solo atingindo uma altura que é uma percentagem da altura do ressalto anterior.



Note-se que, se o dado de saída fosse a altura da bola nos seus vários ressaltos, então poder-se-ia ter uma exploração de progressões geométricas, num contexto de Matemática A. Bastaria, para isso, alterar a variável de saída (*return*) para a *altura* e definir como condição do ciclo recursivo a altura ser muito próxima de zero, podendo-se ainda explorar o conceito de limite de uma sucessão.

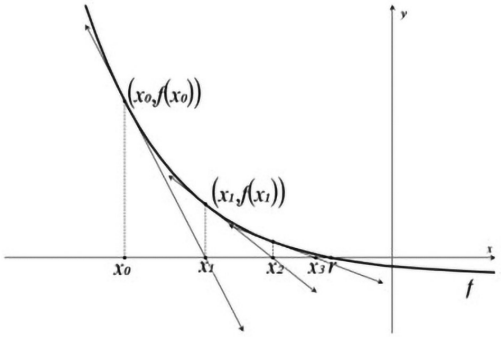
Neste exemplo torna-se ainda evidente o conceito de função, de argumento (objeto), e de saída (imagem), sendo, no caso, um exemplo de uma função de duas variáveis.



MÉTODOS NUMÉRICOS DE RESOLUÇÃO DE EQUAÇÕES E RECURSIVIDADE

Há cerca de 40 anos, o grande pedagogo Sebastião e Silva, referia-se às vantagens da integração curricular precoce do cálculo de aproximações numéricas num contexto de uma certa crítica ao "ensino tradicional" à época (Gonçalves, 2014). As suas reflexões não caducaram e a título de exemplo, observe-se a ideia do método de Newton-Raphson.

Neste método, considera-se uma função f da qual se pretende obter uma aproximação de um zero dada uma sua aproximação inicial. Em condições não muito exigentes obtemos a cada aplicação de uma função iteradora uma melhor, muito melhor, aproximação do zero, utilizando cada aproximação obtida. Dada a aproximação inicial x_0, a aproximação seguinte, x_1, é a abcissa do ponto de interseção da reta tangente ao gráfico de f no ponto de abcissa x_0 com o eixo das abcissas. Cada nova aproximação resulta da anterior aplicando o mesmo procedimento, ilustrado na representação gráfica da figura.



Pode ser um momento interessante o trabalho algébrico para determinar uma aproximação a partir da anterior e com isso perceber que tal pode ser gerado pela fórmula (função iteradora) .

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Um processo destes pode traduzir-se em linguagem de programação Python como se apresenta nas figuras seguintes, no caso para a função $f(x)=x^2 - 2$:

```
3.2 3.3 3.4 *S11 RAD
*newton.py 7/13
from math import *
print("nitera(aproximação,nº de iterações)")
def f(x):
    return x**2-2
def df(x):
    return 2*x

def nitera(a,n):
    for i in range(n):
        a=a-f(a)/df(a)
    return a
```

```
3.2 3.3 3.4 *S11 RAD
Shell Python 9/9
>>>nitera(2,1)
1.5
>>>nitera(2,2)
1.416666666666667
>>>nitera(2,3)
1.41421568627451
>>>sqrt(2)
1.414213562373095
>>>
```

AS PROMOÇÕES E OS DESCONTOS SOBRE DESCONTOS

A literacia financeira é essencial para qualquer cidadão, mais ainda nos dias de hoje em que somos constantemente bombardeados com agressivas campanhas e promoções.

Na disciplina de Matemática Aplicada às Ciências Sociais, aconselha-se que a introdução ao tema de Modelos Financeiros se inicie de forma simples, mas contribuindo de forma clara para o conhecimento do cálculo de percentagens, de descontos e para a importância do cálculo mental.

No exemplo que se apresenta em seguida, encontra-se construído na aplicação TI-Python um curto e simples programa para cálculo do custo final perante um desconto, ou perante um desconto sobre o custo resultante de um anterior desconto, isto é, o designado "desconto sobre o desconto". Será interessante perceber-se o que diriam a maioria dos cidadãos sobre o desconto final, quando temos um "desconto sobre desconto", será que dirão que é igual à soma dos dois descontos! Será mesmo assim?!

```
maisdescontos.py 13/13
from math import *
print("Funções custo1 e custo 2:")
print("▸ custo1(c0,pdesc1)")
print("▸ custo2(c0,pdesc1,pdesc2)=(custof, pdescf)")
def custo1(c0,pdesc1):
    desc1=c0*pdesc1/100
    c1=c0-desc1
    return c1
def custo2(c0,pdesc1,pdesc2):
    c1=c0*(1-pdesc1/100)
    c2=c1*(1-pdesc2/100)
    pdesctotal=(c0-c2)/c0
    return c2, pdesctotal
```

Partindo da aplicação das funções definidas em Python, constantes da figura que se segue, podemos já formular uma conjetura a esse respeito, quer usando a função definida para tal, no caso *custo2*, em que além do custo final surge a percentagem final de desconto, ou então utilizando o conceito de função composta, isto é, a função *custo1* após ela própria.

```
1.1 1.2 MACS10°-MF RAD
Shell Python 9/12
>>>#Running maisdescontos.py
>>>from maisdescontos import *
Funções custo1 e custo 2:
▸ custo1(c0,pdesc1)
▸ custo2(c0,pdesc1,pdesc2)=(custof, pdescf)
>>>custo1(340,25)
255.0
>>>custo2(340,15,10)
(260.1, 0.2349999999999999)
>>>custo1(custo1(340,15),10)
260.1
```

Este simples programa permite uma abordagem pedagógica experimental, de índole investigativa e construtiva, colocando o aluno no centro do processo.

Estas são apenas algumas das vantagens que a programação e o pensamento computacional poderão trazer ao processo de aprendizagem e ao desenvolvimento de competências, numa altura em que urge recuperar o tempo já perdido neste primeiro quarto do século XXI.

Referências bibliográficas

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., Engelhardt, K. (2016). Developing computational thinking in compulsory education – Implications for policy and practice; EUR 28295 EN; doi:10.2791/792158

DGE. (2015). Iniciação à Programação no 1.º Ciclo do Ensino Básico - Linhas Orientadoras Gerais. Retirado de http://www.erte.dge.mec.pt/sites/default/files/Projetos/Programacao/IP1CEB/linhas_orientadoras.pdf

DGE. (2016). Iniciação à Programação no 1.º Ciclo do Ensino Básico - Linhas Orientadoras para a Robótica. Retirado de http://www.erte.dge.mec.pt/sites/default/files/linhas_orientadoras_para_a_robotica.pdf

DGE.(2017). Perfil dos Alunos à Saída da Escolaridade Obrigatória. Retirado de https://dge.mec.pt/sites/default/files/Curriculo/Projeto_Autonomia_e_Flexibilidade/perfil_dos_alunos.pdf

Lucas, M., & Moreira, A. (2018). DigCompEdu: quadro europeu de competência digital para educadores. Aveiro: UA

Ramos, J.L., Espadeiro, R.G. (2016) Estudos de Avaliação - 1.º Ciclo: Iniciação à programação. Centro de Investigação em Educação e Psicologia da Universidade de Évora. ERTE. MEC/DGE

Gonçalves, R. (2014) Utilização de métodos numéricos na resolução de equações e perspetivas de integração curricular no Ensino Secundário. Universidade do Minho

Sítios web consultados em maio de 2021:

https://cache.media.eduscol.education.fr/file/Mathematiques/73/3/Algorithmique_et_programmation_787733.pdf

https://cache.media.eduscol.education.fr/file/Mathematiques/34/0/RA19_Lycee_GT_2-1_MATH_preambule-algorithmique-programmation_1172340.pdf

https://education.ti.com/pt/t3-europe-sites/t3-europe/edublogs/interview-guido-rossum

https://www.dge.mec.pt/noticias/recomendacoes-para-melhoria-das-aprendizagens-dos-alunos-em-matematica-0

Eduardo Cunha
Agrupamento de Escolas de Barcelos

Raul Aparício Gonçalves
Agrupamento de Escolas de Ermesinde
Membros do Grupo de Trabalho T3 da Associação de Professores de Matemática