Uma proposta de ensino de aritmética com Python

Cavalcante, Rogério da Silva
{Instituto Federal de Goiás}, {IFG}
rogerio.cavalcante@ifg.edu.br

3 de julho de 2020

.

1 Introdução

O avanço tecnológico trouxe consigo inúmeros impactos nos mais variados ramos da atuação humana, desde uma plantação à produção em larga escala, telecomunicações, meios de transporte, medicina, pesquisas científicas, relações sociais (globalização), segurança nacional, educação, etc.

Esse desenvolvimento tecnológico goza de uma velocidade maior que as mudanças ocorridas em uma escola, nesse sentido uma questão que torna-se pertinente é: Como mediar tal avanço na busca de uma nova educação, novas práticas, novas estratégias de ensino?

As mudanças dentro do ambiente escolar são complexas pois demandam uma harmonia entre docentes, discentes, gestão, comunidade e outras instâncias conforme citação abaixo.

> As mudanças demorarão mais do que alguns pensam, porque os modelos tradicionais estão muito sedimentados, em parte, eles funcionam, e com isso torna-se complicado fazer mudanças profundas. Por outro lado encontramo-nos em processos desiguais de aprendizagem e evolução pessoal e social. Apesar de haver avanços, muitas instituições e muitos profissionais mantêm uma distância entre teoria e prática, entre suas ideias e ações. Se as pessoas têm dificuldades para evoluir, conviver e trabalhar em conjunto, isso se reflete na prática pedagógica [Moran et al., 2013, pp.24-25].

Nesse contexto um fator marcante ao utilizarmos tecnologias como mediadores educacionais é o impacto explícito no espaço geográfico da sala de aula (distanciamento por exemplo) e sem perder de vista uma a educação focada na formação do ser humano, pode-se fazer uso de recursos tecnológicos para inovar por alguns motivos, tais como:

1. Desafiam as instituições a sair do tradicional;
2. Facilitam a pesquisa, multiplicam os espaços de atuação ;
3. Possibilitam uma aprendizagem mais participativa e integrada;
4. Oferecem aplicativos, jogos, vídeos, dentre outros recursos, que proporcionam dinamismo nos processos educativos em geral.

Desse modo o presente trabalho tem a proposta de relacionar o ensino de matemática (mais especificamente aritmética) com a utilização da linguagem de programação Python, dentre tantas opções em tecnologias, a escolha da linguagem de programação Python se sustenta pelos fatores que serão elencados abaixo:

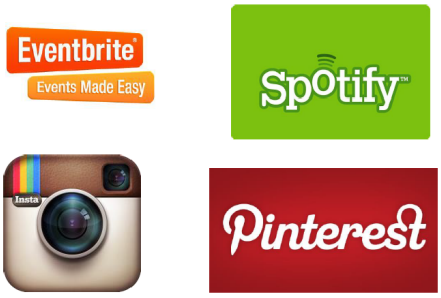
"Python é, provavelmente, a linguagem de programação popular mais fácil e agradável de lidar. O código Python é simples para ler e escrever, e consegue ser conciso sem ser enigmático" [Summerfield, 2012, p.1].

> A linguagem de programação Phyton é muito interessante como primeira linguagem de programação devido a sua simplicidade e clareza. Embora simples, é também uma linguagem poderosa, podendo ser usada para administrar sistemas e desenvolver grandes projetos. É uma linguagem clara e objetiva, pois vai direto ao ponto, sem rodeios. [Menezes, 2016, p.24]

Além do fato de Python ser um software livre (pode ser baixado gratuitamente) e multiplataforma (funciona nos sistema operacionais Windows, Linux e MacOS X, entre outros), é uma linguagem que vem crescendo nos últimos anos e é utilizadas por várias empresas. Além disso, é recomendada por diversos profissionais da área da computação e adodata por diversas empresas segundo [Masanori, 2018]

Figura 1: Instituições que usam Python

Figura 2: Aplicativos que usam Python



Apesar de fenomenais, importante destacar que o fato de se ter um computador (ou qualquer outro recurso tecnológico), não acarreta que se tenha aprendizagem. O professor como mediador do conhecimento deve buscar e planejar atividades, com objetivos bem determinados que potencializem tais recursos.

2 Desenvolvimento

O desensvolvimento do trabalho dar-se-á com uma rápida apresentação da linguagem Python seguido de sua aplicação em alguns resultados da aritmética.

2.1 Uma rápida apresentação do Python

Comecemos com a instalação do Python que pode ser vislumbrada em [Masanori, 2020], alternativamente para quem não quer instalar nada, pode-se programar em Python pelo google colab para maiores detalhes ver [Anibal Azevedo, 2020], nesse texto será abordado o Python 3.8.3 instalado no sistema operacional Windows 10.

A código 1 abaixo apresenta o shell do Python que é o seu modo interativo, isto é, ele mostra o resultado dos comando logo após apertar enter

Código 1: Operações básicas e uso do print

```
>>> print('Olá Mundo!') # primeiro programa (uso do print)
Olá Mundo!
>>> 16+5 # Sempre apertar Enter após o comando
21
>>> 16 - 5 #  Subtração.
11
>>> 16*5 #  Multiplicação.
80
>>> 16**5 # Potência.
1048576
>>> 16/5 # Divisão.
3.2
>>> 16//5 # Quociente da divisão inteira.
3
>>> 16%5 # Resto da divisão inteira.
1
```

Observe que no código 1 após # tem um comentário sobre o que o comando em questão retorna, além de trabalhar com números explicitamente podemos declarar variáveis que podem ser do tipo: número inteiro (int), número real(float), lista (array ou vetor) e string (cadeia de caractere que estará sempre entre aspas simples ou duplas), booleano (verdadeiro ou falso).

O Python consegue distinguir se uma dada expressão lógica é verdadeira ou falsa, expressão esta que é composta pelos operadores relacionais que são: $>$ (maior), $<$ (menor), $>=$ (maior ou igual), $<=$ (menor ou igual), $==$ (pergunta se é igual) e $!=$ (pergunta se é diferente) e podem conter os operadores lógicos **not**, **or** e **and** vejamos no código abaixo:

Código 2: Alguns comandos com variáveis

```
>>> a=2 # declarando a variável a (tipo inteiro)
>>> b=5.2 # declarando b tipo float
>>> c=a+b # declarando c
>>> c==7.2 # perguntando se c é igual a 7.2
True
>>> b<a # perguntando se b é menor que a
False
>>> d='laranja' # declarando d tipo string
>>> e=[1,2,'oi'] # variável e tipo string
>>> len(d) # retorna o "tamanho" de d
7
>>> e.append(5) # acrescenta o 5 ao final da lista
>>> e # enter para ver a nova variável e
[1, 2, 'oi', 5]
>>> 3*a+b/a+10 # uma expressão numérica
18.6
```

Temos ainda estruturas básicas que se valem da lógica exposta nos parágrafos acima, as duas que usaremos serão o if e o while de modo suscinto o if executa um bloco de comandos quando a expressão que ele avalia é verdadeira e o while executa seu bloco de comando enquanto a expressão que ele avalia for verdadeira, caso as expressões sejam falsas o programa segue seu fluxo. Vale ressaltar que o print e o input são basicamente saída e entrada de dados.

2.2 Aplicações na aritmética

2.2.1 Fatoração

Seguem definições e resultados bem estabelecidos:

Definição 2.1 *Um número natural $p > 1$ é dito primo se é divisível apenas por 1 e por si mesmo.*

Teorema 2.1 (Teorema Fundamental da Aritmética) *Seja $n \in \mathbb{N}$, $n > 1$. Então n é primo ou se escreve de modo único (a menos da ordem dos fatores) como produto de primos.*

Para ver uma demonstração consultar [HEFEZ, 2011, Sidki, 1975, p.83,p.20], segue um código em Python que retorna a forma fatorada de um número natural

Código 3: Fatoração de n

```
n = int(input('Digite o natural que será fatorado: '))
k=n # preservando o valor de n em k
d=2
fatores='' #string vazia
while n>1:
    if n%d==0:
        fatores=fatores+str(d)
        n=n//d
    else: # o else executa quando o if é falso
        d=d+1
print(k,'=','*'.join(fatores)) #apenas um print mais usual
RESTART:
Digite o natural que será fatorado: 210
210 = 2*3*5*7

```

2.2.2 Máximo Divisor Comum

Dados dois naturais a e b ambos não nulos, diremos que d é um divisor comum de a e b se $d|a$ e $d|b$. Por exemplo, 3 é divisor comum de 15 e 27.

Definição 2.2 *Diremos que d é o máximo divisor comum de a e b se:*

i) *d é um divisor comum de a e b;*

ii) *Se c é um divisor comum de a e b, então $c|d$.*

Denotaremos o mdc de a e b por mdc(a,b).

E para finalizar vamos ao famoso Algoritmo de Euclides[1] que é considerado uma pérola computacional.

O código 4 se embasa na prova construtiva da existência do mdc(a,b) em [HEFEZ, 2011, p.56-57] que na literatura geralmente se apresenta sob o dispositivo prático (jogo da velha) abaixo.

	q_1	
b	a	
r_1		

	q_1	q_2	
b	a	r_1	
r_1	r_2		

	q_1	q_2	$\ldots$	q_{n-1}	q_n	q_{n+1}
b	a	r_1	$\ldots$	r_{n-2}	r_{n-1}	$r_n = \text{mdc}(a,b)$
r_1	r_2	r_3	$\ldots$	r_n	0	

Onde q_1 e r_1 são respectivamente o quociente e o resto da divisão de b por a, q_2 e r_2 são respectivamente o quociente e o resto da divisão de a por r_1 e assim sucessivamente e o mdc(a,b) será o resto imediatamente anterior ao resto zero.

Exemplificando vamos calcular o mdc(255, 221)

	1	6	2
255	221	34	17
34	17	0	

Logo mdc(255, 221) = 17

O objetivo agora é fazer um programa que calcule o mdc de dois números fornecidos pelo usuário

Código 4: mdc(a,b)

```
a=int(input('entre com o valor de a: '))
b=int(input('entre com o valor de b: '))
x,y=a,b # guardando a e b para o print
while b!=0:
    a,b=b,a%b
print(' mdc (%d,%d)='%(x,y),a)
RESTART
entre com o valor de a: 255
entre com o valor de b: 221
mdc (255,221)= 17
```

3 Considerações Finais

O presente trabalho teve como objetivo central apresentar a possibilidade da linguagem de programação Python como ferramenta para o ensino de aritmética. Neste sentido, apresentamos algumas funcionalidades do Python, aquelas que seriam utilizadas nos programas desenvolvidos. A ideia era propor um material que permitisse interligar o ensino de uma linguagem de programação com o entendimento dos conceitos e sua lógica, resultados apresentando-os sob a forma de um algoritmo.

Para a visualização dos programas de modo dinâmico, para entender melhor seu funcionamento consultar [Philip Guo, 2018]

Por fim o que foi aqui exposto é uma gota no imenso oceano de possibilidades que se pode explorar tanto em Matemática(matrizes, gráficos, trigonometria, funções elementares e estatística etc) como em Python(suas bibliotecas, jogos, orientação à objeto, banco de dados, dentre outros assuntos).

Referências

[Anibal Azevedo, 2020] Anibal Azevedo (2020). Projeto python para todos. Disponível em: `https://www.youtube.com/watch?v=kzocN6Zugb4&list=PLH9knZH6lcgo0ndsXfwX0CGO_ad1t6cSO&index=2&t=0s`. Acessado em 03/06/2020.

[HEFEZ, 2011] HEFEZ, A. (2011). *Elementos de Aritmética, 2ª Edição, Rio de Janeiro.* SBM.

[1] Matemático grego que viveu por volta de 300 a.C. e escritor da inestimável obra *Os Elementos*

[Masanori, 2018] Masanori (2018). Python zumbi 0. Disponível em: https://github.com/fmasanori/PPZ/blob/master/TWP00%20Apresenta%C3%A7%C3%A3o%20e%20Motiva%C3%A7%C3%A3o.pdf. Acessado em 03/06/2020.

[Masanori, 2020] Masanori (2020). Python zumbi 2.0. Disponível em: https://www.youtube.com/watch?v=HMeshqR0QEY&list=PLUukMN0DTKCtbzhbYe2jdF4cr8MOWClXc&index=2. Acessado em 03/06/2020.

[Menezes, 2016] Menezes, N. N. C. (2016). *Introdução à programação com Python–2ª edição: Algoritmos e lógica de programação para iniciantes*. Novatec Editora.

[Moran et al., 2013] Moran, J., Maseto, M. T., and Behrens, M. A. (2013). *Novas Tecnologias E Mediação Pedagógica*. Coleção Papirus Educação. Papirus.

[Philip Guo, 2018] Philip Guo (2018). Python tutor. Disponível em: http://pythontutor.com/. Acessado em 15/01/2018.

[Sidki, 1975] Sidki, S. (1975). *Introdução à teoria dos números*. IMPA.

[Summerfield, 2012] Summerfield, M. (2012). *Programação em Python 3: Uma introdução completa à linguagem Python*. Biblioteca do Programador. Alta Books.