

FLUXOGRAMAS, ALGORITMOS, MATEMÁTICA E TECNOLOGIA

Você pode me explicar como os gráficos de uma função qualquer são construídos em um aplicativo de celular (ou de um computador)? Você tem ideia de como são calculados logaritmos, raízes quadradas, raízes cúbicas, exponenciais, senos, cossenos e tangentes em um programa de computador? Você tem ideia de como são produzidos os aplicativos para celular? E se eu te disser que já há adolescentes criando aplicativos mundo afora? Já pensou se em algum país as crianças já soubessem criar programas?

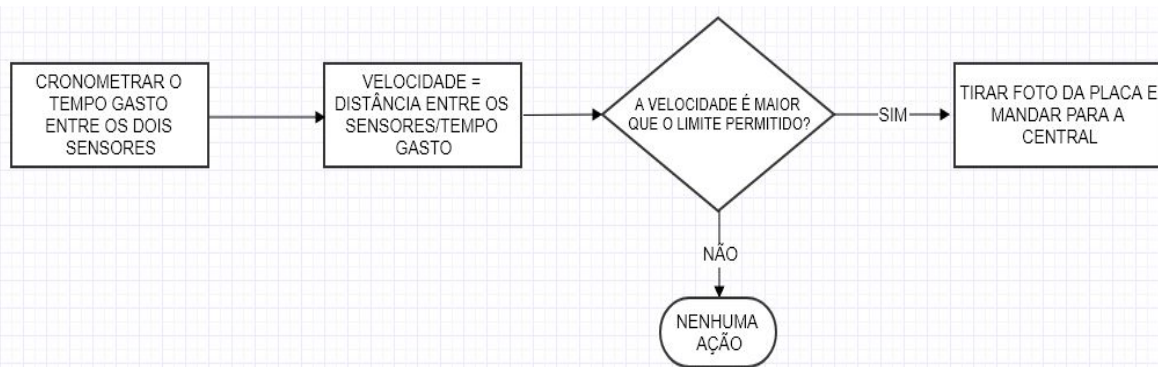
Em 2007, nos Estados Unidos, nascia o Scratch, plataforma que ensina programação a crianças a partir de 8 anos. A iniciativa do MIT (*Massachusetts Institute of Technology*) tem como prerrogativa o fato de que a programação é uma outra forma de enxergar o mundo, ou seja, trata-se de uma linguagem, como o inglês, a matemática, ciências, o português, etc...

Uma pergunta que naturalmente pode surgir a partir desse fato é: podemos dizer que essas crianças, que mesmo tão jovens são capazes de programar um computador, são pequenos gênios? É de se notar que em março de 2018 a plataforma já conta com 30 milhões de projetos compartilhados. Seriam milhões de gênios? Um exercício simples de imaginação pode nos levar a prever as consequências educacionais, sociais e tecnológicas ao se ter tantas mentes aptas a enxergar o mundo através de mais este viés, segundo mais essa linguagem.

Após o advento do computador e com o constante aperfeiçoamento das técnicas computacionais, ficou em maior evidência a Matemática Discreta, que trata basicamente de tudo o que pode ser traduzido para linguagem computacional, ou mais precisamente, tudo o que pode ser traduzido em algoritmos para o computador. A palavra algoritmo é atribuída a *sequência finita de instruções a serem seguidas e que atingem um objetivo após um número finito de passos*. Sendo assim, para você ler este texto você realizou uma sequência de passos, um algoritmo. Para você ir trabalhar diariamente, você realiza uma sequência de passos. Algoritmos estão presentes o tempo todo em nossas atividades, basta observar nossa rotina.

Para a surpresa de muitos, os computadores executam passos muito bem determinados, e, portanto, não ambíguos. Se a instrução não for clara, nenhum programa funciona. O computador executa EXATAMENTE as instruções dadas por seu programador. Quando você utiliza algum *software* que realiza coisas incríveis, lembre-se que o mérito na verdade está em quem deu estas instruções ao dispositivo (computador, celular, tablet, etc...).

Vamos pensar em um exemplo bem simples: como funcionam os radares que identificam a velocidade de um automóvel em um determinado trecho da estrada? Inicialmente pensemos no que queremos que seja realizado pelo nosso "software". Imaginemos um software que calcule a velocidade do carro e caso esta velocidade seja maior do que o limite permitido, que ele me diga de alguma forma qual veículo cometeu esta infração. Observe o Fluxograma abaixo:



De forma bem simplificada, este fluxograma apresenta o algoritmo a ser executado pelo sistema que vai definir aplicar a multa ou não, de forma totalmente automática. Preste atenção ao fato de que as instruções tem que estar muito bem definidas. O dado que entra no sistema é o tempo gasto entre dois sensores. Já que é conhecida a distância percorrida pelo automóvel entre os dois sensores, então fica fácil determinar a velocidade com a qual o automóvel passou pelos sensores.

Imagine agora fluxogramas (e algoritmos) que identificam máximos divisores comuns, mínimos múltiplos comuns, raízes de equações, áreas de quadrados, volumes de sólidos, calculam médias, medianas, desvio padrão, etc... Segue abaixo um fluxograma que serve de base para um algoritmo que resolve equações de 2º grau ($ax^2+bx+c=0$) utilizando a fórmula de Bháskara (A, B e C representam os coeficientes da equação padrão).



Consegue perceber que toda e qualquer atividade repetitiva pode ser executada por um algoritmo computacional? Mais uma pergunta que pode ficar no ar: e a maneira pela qual ensinamos Matemática está levando isso em conta?

Um aspecto muito importante a ser discutido diz respeito ao fato de que, por serem passos discretos, o computador tem dificuldades em lidar com problemas do que comumente chamamos de *contínuo*. Já que os números reais contém os números racionais (que são frações de números inteiros) e os irracionais (que não podem ser representados por frações) como podemos utilizar números irracionais (o Pi (π), por exemplo) na resolução de certos problemas? Como podemos dizer para o computador que ele deve utilizar a raiz quadrada de dois em um problema? Será que ele vai ser

capaz de medir uma diagonal de um quadrado, pegar o resultado e utilizar no problema?

Quando estudamos Séries na graduação, normalmente são estudadas as representações de certas funções ($f(x)$) em Séries (normalmente de Taylor e de MacLaurin). Estas séries são somas infinitas que convergem para $f(a)$ quando fizermos $x=a$. Observe:

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}, \text{ para qualquer } x \in \mathbb{R}, \text{ ou seja, para se calcular a exponencial de 5,}$$

por exemplo, basta substituir o x por 5 na igualdade acima. Do lado direito da igualdade aparecerá uma soma infinita cujo resultado final será de fato, e^5 . A questão é que não é humanamente ou computacionalmente possível, se calcular essa soma dos infinitos termos. A solução portanto, é pegar um pedaço da soma infinita, que por ter caráter polinomial é facilmente calculável pelo computador. O caminho é definir quantos termos da soma serão adicionados para se ter o valor tão próximo quanto se queira do resultado real. Dessa mesma forma são calculados senos, cossenos, logaritmos naturais, etc...

Surpreso? Tente imaginar como um computador pode desenhar uma reta, já que uma reta está em correspondência com o conjunto dos números reais e, portanto, trata de um objeto contínuo. Da mesma forma, os gráficos de função contínua passam pelo mesmo dilema. Se o computador executa passos discretos, como ele é capaz de traçar objetos contínuos?

O princípio é exatamente o mesmo usado nas funções transcendentais. Já que o olho humano não detecta certos "buracos" nos gráficos, o computador determina uma discretização que permita uma visualização que nos pareça contínua.

Agora imagine uma aplicação em Engenharia, por exemplo, uma simulação de tensões em uma peça mecânica. A peça em si pertence a um outro contínuo, chamado *contínuo físico*. Normalmente os engenheiros utilizam ferramentas computacionais baseadas em *Elementos Finitos* (às vezes são utilizados *Volumes Finitos* e em outras *Elementos de Contorno*), que nada mais são do que pontos discretos que, em conjunto, representam o que acontece no todo (contínuo). São expressões muito comuns nesse meio: *discretização, malha, interpolação e elemento de controle*.

Toda a teoria é produzida para que quanto mais pontos na discretização, mais próximo do resultado efetivo se chegue. Se lembra da noção intuitiva de Integral? O cálculo de áreas de figuras planas é o que motiva a definição de integral de uma função sobre um intervalo na reta real. O procedimento intuitivo consiste em simplesmente discretizar o domínio transformando-o em um conjunto de segmentos de reta que geram retângulos cuja somatória das áreas representa a área efetiva. Analogamente (e esperadamente) quanto maior a quantidade de retângulos utilizados na discretização do domínio, mais próximo se chega da área efetiva entre o gráfico de $f(x)$ e um intervalo da reta.

Essa tônica vai se repetir em praticamente todas as aplicações envolvendo o contínuo. No que se refere a problemas contendo dados discretos, as possibilidades são ainda maiores. Imagine uma cidade contendo n pontos de coleta de lixo e você dispõe de um caminhão para fazer a coleta. Por razões econômicas você adoraria que esse caminhão fizesse um trajeto ótimo, ou seja, o trajeto que mais economize tempo e combustível. Essa é uma versão de um problema conhecido como *Problema do Caixeiro Viajante* e sua solução envolve um tempo computacional que cresce exponencialmente cada vez que se aumenta n .

De fato, as possibilidades de geração de solução algorítmica para problemas de Matemática Discreta são crescentes. Há algoritmos baseados no comportamento de colônias de formigas, algoritmos baseados em feromônios, algoritmos baseados em física quântica, outros baseados em problemas de presa-predador, algoritmos genéticos, etc... Aqui é um dos muitos momentos em que criatividade e Matemática trabalham juntas. Sinta-se motivado a pesquisar a respeito e incentivar seus alunos a pensarem em linguagem computacional.

Se considerarmos a crescente demanda por automação na indústria, a crescente importância da internet das coisas e a inteligência artificial aparecendo em vários dispositivos, não é difícil perceber a importância de se ensinar a linguagem computacional para crianças. O cidadão de um futuro muito próximo irá lidar diariamente com estas novas tecnologias e torna-se muito importante fazer com que se apropriem deste modo de ver o mundo. As sintaxes das linguagens de programação são secundárias, mas fluxogramas, descrição de etapas e raciocínio já podem ser ensinadas pois isto faz com que as crianças descubram novas formas de resolver problemas e se motivem a usar a criatividade. Quem disse que criatividade e Matemática não podem ser ensinadas simultaneamente?

Sugestões de sites para aprofundamento

<https://scratch.mit.edu/> - Site do projeto Scratch do MIT, pioneiro no ensino de Programação para crianças.

<http://programae.org.br/> - Programaê - Iniciativa da Fundação Lehman para disseminar o ensino de programação para crianças. Permite cadastro para interessados em ensinar programação.

<https://novaescola.org.br/conteudo/7111/atividades-desplugadas-ensinar-linguagem-de-programacao-sem-computador> - Associação Nova Escola - Atividades desplugadas - Linguagem de Programação sem Computador - Nesse artigo são apresentadas atividades que podem servir de introdução à linguagem de programação sem utilizar necessariamente o computador para tal.

<https://www.codecademy.com/pt-BR> - Codecademy - Projeto de ensino de programação gratuito, voltado para adultos e profissionais de programação.