

# EMC 5003 | Tecnologia & Desenvolvimento



**UNIVERSIDADE FEDERAL  
DE SANTA CATARINA**

## O PAPEL DO ENSINO DE PROGRAMAÇÃO NA FORMAÇÃO DO ENGENHEIRO DO FUTURO

Douglas Felipe Marangon

Mateus Bortoli Harano

Florianópolis, 08 de julho de 2019



**Douglas Felipe Marangon** – douglasfmarangon@gmail.com  
Departamento de Engenharia Mecânica, UFSC  
Campus Reitor João David Ferreira Lima

**Mateus Bortoli Harano** – mateus.harano@ufsc.br  
Departamento de Engenharia Mecânica, UFSC  
Campus Reitor João David Ferreira Lima

## Resumo

“Diante do desenvolvimento tecnológico brasileiro, a programação se mostra a cada dia, uma habilidade mais valorizada. No estudo de engenharia, o desenvolvimento acadêmico formal por si só não é capaz, no estado vigente, de dar todo o suporte necessário para a excelência dos graduandos nessa área. Assim, o presente artigo apresenta uma avaliação sobre a importância da programação no desenvolvimento técnico do engenheiro brasileiro.

Em um primeiro momento, apresenta-se o panorama social atual da computação e se apresenta o movimento *Everyone Should Code* [1]. Em seguida, é feita uma análise do contexto brasileiro de formação de jovens engenheiros e um levantamento das oportunidades e dificuldades enfrentadas pelas universidades. Por fim, conclui-se como uma ponderação sobre a necessidade ou não de uma reformulação na grade curricular nos cursos de engenharia.”

**Palavras-chave:** programação, ensino, engenharia.



## Abstract

“Looking at the Brazilian technological development, programming is becoming more and more valued every day. In the engineering study, formal academic development alone is not able, in the current state, to provide all the necessary support for the excellence of undergraduates in this area. Thus, the present article presents an evaluation about the importance of programming in the technical development of the Brazilian engineer.

At first, the current social panorama of programming is presented, followed by the *Everyone Should Code* movement. Then, an analysis is made of the Brazilian training context for young engineers and a survey of the opportunities and difficulties faced by universities. Finally, it is concluded as a consideration about the necessity or not of a reformulation in the curriculum in the courses of engineering.”

**Keywords:** coding, teaching, engineering.



---

# 1.Introdução

Principalmente após 1990, com a popularização da internet, a habilidade de programar um computador se torna mais importante e presente no mundo. Empresas como a Google, o Facebook, a Microsoft e a Apple trouxeram como proposta de valor o desenvolvimento de programas que aumentassem, de uma forma ou de outra, a qualidade de vida da população. Enquanto isso, a sociedade também aderiu às inovações tecnológicas, incorporando-as ao seu estilo de vida, o que tornou a programação uma parte quase que vital para a vivência diária.

Para acompanhar esse crescimento do consumo de aplicativos e programas, as empresas aumentaram sua produção e, portanto, a demanda por mão-de-obra qualificada. A quantidade de programadores, porém, que surgiam por graduação tradicional através de universidades, não acompanhava a quantidade de vagas que apareciam nessa área. Assim, as grandes empresas começaram a incentivar o aprendizado de programação para toda a população.

O movimento *Everyone Should Code* - Todos Devem Programar - nasceu, assim, como consequência orgânica da mentalidade implementada pelas empresas americanas. Logo, uma quantidade massiva de jovens estava aprendendo computação e desenvolvendo seus próprios códigos ainda através da *internet*.

O movimento proporcionou, ainda, consequências muito positivas para o cenário norte-americano. A versatilidade da programação trouxe a oportunidade de se colocar em prática ideias disruptivas de maneira bastante rápida, sem que se exigisse um capital inicial exacerbado. Empresas como o Snapchat - os chamados Unicórnios - foram e são capazes de gerar uma imensa quantidade de valor rapidamente.

Atualmente, cerca de 40% das escolas americanas já tem ao menos uma aula de computação por semana e países como Reino Unido, Finlândia e Estônia têm a programação como matéria obrigatória no ensino de base. Por isso, espera-se que a quantidade de vagas na área de Tecnologia da Informação aumente para os próximos anos. Inclusive, é previsto pelo movimento *Everyone Should Code*, que em breve será difícil conseguir um emprego sem que se saiba programar.

Na área da engenharia, os conceitos de programação fazem parte do dia-a-dia do profissional, que se utiliza diariamente de *softwares* de CAD e CAE. Mais do que isso, em muitas funções, engenheiros são alocados para desenvolver e resolver métodos numéricos para a resolução de problemas usuais em fábricas montadoras, por exemplo. Para que isso seja feito, a habilidade de programação deve ser pré-requisito na contratação desse profissional.

Contrariamente à década passada, grande parte das vagas de empregos ofertadas a engenheiros atualmente têm a habilidade de programação como característica desclassificatória, não vista mais como diferencial competitivo. Apesar disso, as universidades brasileiras, hoje, não acompanham esse desenvolvimento em sala de



---

aula, de modo que a maior parte dos engenheiros em formação devem buscar esse conhecimento fora das universidades.

## **2. Quantidade e necessidade de habilidades de programação no mercado de trabalho atualmente e no futuro**

O código de Ada Lovelace para calcular números de Bernoulli foi o que ficou conhecido mundialmente como o primeiro programa de computador da história humana. Nesse momento, em 1843, as utilizações desse conhecimento de programação ainda eram bastante restritas. Isso se deve principalmente porque computadores como se conhece atualmente não eram produzidos. [2]

Nessa época, cada máquina era montada quase que especificamente para uma determinada função e os avanços em tecnologia não eram suficientes para a construção de um dispositivo que congregasse várias operações. Desse modo, a programação de um aparelho era, na realidade, ajustes mecânicos que se fazia para que a máquina operasse de forma a retornar um resultado específico de acordo com os parâmetros inseridos. Portanto, a programação do sistema era inerente à própria concepção mecânica do aparelho, de forma que, após sua criação e fabricação, a presença de um “programador” já não era necessária.

Foi apenas na década de 1940 que os primeiros computadores elétricos multifuncionais foram criados. Assim, para que se executasse diferentes funções, era necessário que se programasse a máquina de forma transformar a ideia de uma nova função em um conjunto de procedimentos a serem seguidos. Dessa necessidade de transcrever algoritmos para uma forma compreensível para o computador, nasceram as primeiras linguagens de programação: Plankalkül, ENIAC coding system e C-10.

Em 1990, enfim, as possibilidades que a internet ofereceu ao mundo da computação permitiu o desenvolvimento de linguagens de programação mundialmente aceitas e conhecidas. O potencial de intercomunicação entre diversas partes do mundo foi essencial para que algoritmos fossem compartilhados e, assim, novas ideias surgissem.

Atualmente, o alto poder de processamento em dispositivos cada vez mais compactos e acessíveis tornou possível a criação de funções nunca esperadas anteriormente, tais como redes sociais, aplicativos de produtividade e jogos virtuais. Mais do que isso, o uso de processadores em aparelhos médicos como o ultrassom, por exemplo, mostra como a sociedade se utilizou da capacidade adaptativa dos aparelhos eletrônicos e como continua a se desenvolver nesse sentido.

Dessa maneira, para que todas essas novas funções fossem criadas, cada vez mais foi necessária a presença de programadores capazes de transformar essas funções



---

em algoritmos implementáveis. Segundo a estimativa da organização Code.org, é esperado que o mercado brasileiro alcance a marca de 1,4 milhões de vagas para programadores até 2020. Isso representa o dobro da quantidade atual de vagas. [3]

Assim, como se percebeu nos últimos 50 anos, o desenvolvimento tecnológico hoje está intimamente atrelado à programação. O maior problema disso é que, enquanto a tecnologia cresce exponencialmente, a quantidade de programadores cresce linearmente. A Associação para a Promoção da Excelência em Software Brasileiro projetou que em 2020 faltarão mais de 400.000 desenvolvedores para preencher vagas de Tecnologia da Informação no Brasil [4]. Caso isso realmente aconteça, o desenvolvimento tecnológico brasileiro deve sofrer com essa estagnação, com efeitos árduos em termos de economia e, por isso, sociedade.

Embora a maior parte dessas vagas serem preenchidas por profissionais formados em áreas de Tecnologia da Informação, as engenharias não relacionadas diretamente à computação têm um papel vital no desenvolvimento trabalhista. Uma pesquisa feita pelo fórum *StackOverflow* [5] em mostra que, dos 101.592 programadores entrevistados, 8.940 são engenheiros cuja formação não é diretamente relacionada à computação. Ou seja, 8,8% dos desenvolvedores que usam o fórum para aprender programação são engenheiros mecânicos, eletricitas, civis, etc. Entende-se, dessa forma, que há uma necessidade real de aprendizado de programação por parte dos jovens engenheiros.

### **3.Situação atual do ensino de programação nos cursos de engenharia**

Apesar de o mercado de trabalho ter se desenvolvido no sentido a aumentar a demanda por programadores, os currículos dos cursos de engenharia no Brasil não acompanharam essa demanda com alterações em sua grade curricular. O curso de engenharia mecânica da UFSC é um exemplo disso, no qual os estudantes têm pouco contato com programação durante o curso.

Na UFSC, os cursos de engenharias mecânica, civil e sanitária e ambiental apresentam duas disciplinas que tratam de assuntos relacionados à computação: “Introdução à Ciência da Computação” e “Cálculo Numérico em Computadores”. Apesar disso, é importante salientar que o conteúdo oferecido nessas disciplinas não é comparável ao nível de habilidade requerido pelo mercado de trabalho.

Depois de aplicadas essas disciplinas, os cursos não oferecem, oficialmente, nenhuma abordagem computacional às matérias seguintes. Desse modo, assuntos como análise estrutural são ensinados sem que se comente sobre a aplicação da programação para a resolução de problemas. Apesar disso, no mundo executivo, as soluções manuais como se conheceu em sala de aula não são aplicáveis. Cabe, portanto, ao estudante, a abstração desorientada de soluções computacionais.



## **4. Vantagens e desvantagens da inclusão do ensino de programação na base nacional curricular comum dos cursos de engenharia**

### **Desenvolve Raciocínio**

Steve Jobs, em uma entrevista em 1995 anunciou que “todos nesse país deveriam aprender a programar um computador, porque isso o ensina a pensar”. [6]

A maior parte do currículo de um curso de engenharia, nos moldes atuais, foca no desenvolvimento de um perfil generalista: ensinar o básico de vários assuntos de modo a mostrar o caminho do estudo para cada área específica, sem que se desenvolvam estudos aprofundados em cada disciplina [7]. A ideia por trás disso é que o estudante seja exposto a raciocínios diferentes, que o prepararão para enfrentar desafios técnicos no mercado de trabalho.

Por exemplo, o objetivo da disciplina de termodinâmica aplicada é apresentar conceitos e métodos de resolução de problemas. É claro que, na vida prática de um engenheiro, será necessário muito mais do que os cálculos feitos em sala de aula. Será necessário que o profissional enuncie o problema em uma modelagem matemática cuja solução se traduza em ajustes concretos factíveis e, posteriormente, uma análise econômica das possíveis soluções. Contudo, tendo estudado essa disciplina, o engenheiro deve ser capaz de abstrair aquela determinada situação em um problema passível de aplicação do método que conheceu ou, em alguns casos, de uma adaptação da metodologia.

Analogamente, o aprendizado de programação tem papel crucial no desenvolvimento de análises sequenciais lógicas e, posteriormente, na categorização dos resultados em procedimentos claros, levando em conta os princípios de causa e efeito.

Dessa forma, o ensino de programação pode ser colocado como alicerce desse desenvolvimento racional. Desde o começo da computação, o raciocínio lógico foi requisito básico para qualquer programador. OSMAN, M.N., MAGHRIBI, M. HAMID, J.N., ZAIN, N.M., JAMALUDDIN, M.N.F., MAZLAN, U.H., JAN, N.Y.C. e AHMAD, H, 2015 [8], mostraram como o aprendizado em programação é capaz de aumentar o poder de raciocínio lógico dos estudantes.

### **Novas soluções no mercado de engenharia**

Além disso, assim como a engenharia se adaptou após a revolução industrial, adicionando conceitos de máquinas e métodos mecânicos e elétricos para resolução de problemas, a revolução da internet também requer sua mudança. Embora a sociedade tenha incorporado a internet e a programação em quase todas as suas



---

relações diárias, os métodos ensinados na universidade permanecem, de forma geral, retidos a procedimentos manuais ou mecânicos.

É bastante comum encontrar matérias de desenho técnico manual nos currículos dos cursos de engenharia, muito embora a existência de softwares CAD tenha aumentado em muito a eficiência e a produtividade dos engenheiros. Mais do que isso, saber utilizar desse recurso deixou de ser um diferencial curricular e passou a ser um requisito desclassificatório em processos seletivos.

O mundo do desenvolvimento de tecnologias é, hoje, profundamente dependente de aumento da eficiência de profissionais que consigam executar tarefas de maneira ágil e excelente. Nesse sentido, o desenvolvimento de códigos simples pode acelerar cada tarefa do dia-a-dia do profissional, imprimindo a ele uma velocidade que o transformará em um engenheiro de alta performance.

Ademais, métodos de análise de estruturas, por exemplo, cuja solução apresentada em sala de aula é manual, hoje estão se transformando em métodos automáticos em softwares desenvolvidos internamente pelas empresas. Por isso, além de saber usar os principais programas de computador para engenharia, um profissional deve ser capaz de transformar estudos de diversas áreas em programas que retornem respostas rápidas e com baixo índice de erro.

A habilidade de programar pode alavancar exponencialmente a eficiência e a qualidade do trabalho de um engenheiro. Assim, além de aumentar as possibilidades de contratação de um profissional, o resultado de seu trabalho será entregue mais rapidamente, provando seu valor ao empregador e garantindo sua estabilidade na função.

### **Exclusão de disciplinas necessárias à formação do engenheiro**

Por outro lado, adicionar disciplinas básicas ao currículo comum de engenharia implica, necessariamente, na retirada de alguma outra disciplina. As grades dos cursos de engenharia já encontram dificuldades claras na tentativa de priorizar e aglutinar uma vasta variedade de habilidades técnicas que um engenheiro deve ter em um aprendizado com tempo reduzido.

Embora essa seja uma habilidade importante ao engenheiro atual, a programação não é parte da definição de um profissional na área. É verdade que um engenheiro que não entende de computação tem seu escopo de trabalho reduzido. Porém, ainda não é esperado que um engenheiro mecânico, por exemplo, saiba programar em Python tanto quanto se espera que ele consiga resolver problemas de transmissão de calor.

Dessa maneira, a alteração da base curricular nacional tem uma clara implicação limitante, de se renunciar a disciplinas e assuntos essenciais para a formação do engenheiro para que o ensino de programação tome seu lugar. Não está em discussão, porém, a relevância técnica dos conhecimentos do engenheiro para o



---

mercado de trabalho. É possível que algumas áreas de pesquisa e desenvolvimento das universidades estejam, hoje, obsoletas e não agreguem à formação do engenheiro tanto quanto a computação faria.

### **Infraestrutura e Custo**

Ademais, para que se possa ensinar computação, é necessário que os alunos tenham acesso a computadores. Embora seja possível que se ensine a lógica dos algoritmos em quadro negro, o acesso a máquinas dá a oportunidade aos estudantes de desenvolver a independência lógica, ao invés de apenas decorar códigos. É parte do aprendizado de programação que os estudantes tenham a liberdade para criar, testar, errar e aprender com seus erros.

De acordo com MAZLAM e OTHMAN, 2015 [9], o desenvolvimento da lógica da programação está ligada a uma sequência de seis passos, dentre os quais três necessitam de uma abordagem prática.

Contudo, com a quantidade de computadores que as universidades têm, isso não é possível. Apesar de ser uma habilidade importante a ser desenvolvida, seu desenvolvimento está atrelado a um custo extremamente alto para as universidades. Quando comparado com o ensino de uma disciplina teórica igualmente importante, a alternativa clara é que se opte pela mais barata. Por isso, é razoável que o baixo estímulo à programação esteja atrelado ao custo da infraestrutura necessária dessa área de estudo.

## **5. Conclusão**

O desenvolvimento de tecnologias atualmente está, com certeza, atrelado à programação. Sobretudo na engenharia, o progresso científico e comercial se mostra largamente dependente de mão-de-obra qualificada que seja capaz de unir conceitos de física aplicada e tecnologia digital. Mais do que isso, o mercado brasileiro tem projeção de escassez de profissionais capacitados para atuar na programação, o que faz do momento atual oportuno para o desenvolvimento de políticas de incentivos ao estudo dessa área do conhecimento.

Apesar disso, a infraestrutura do ensino público, caso opte por reformar a base nacional curricular dos cursos de engenharia, enfrentará um grande desafio econômico. Essa proposta só faz sentido se for possível assegurar o retorno desse valor em forma de desenvolvimento técnico à sociedade.

Como sugestões ao leitor desse artigo, recomenda-se uma análise aprofundada sobre a importância de cada disciplina para o mercado de trabalho, relacionando-a à possibilidade de troca de uma matéria menos interessante pelo ensino de programação. Além disso, muito embora a implementação de novas disciplinas de computação tenha um custo elevado, a difusão desse conhecimento tem a possibilidade de ser expandido de outras formas. Como exemplo, pode-se citar o



apoio a iniciativas como o Progamaria, que promove o ensino de programação para mulheres no Brasil, e o Code\_for, o qual oferece a inserção da população de baixa renda – sobretudo jovens – no mercado de trabalho tecnológico.

A decisão, dessa forma, pela aderência ou não do ensino de programação às grades curriculares não reside na avaliação da importância dessa habilidade. Essa escolha tem como problema-raiz a priorização de custos de cada universidade.

## 6.Referências

- [1] J. M. Wing, *Computational Thinking*, Tianjin: China, 2012.
- [2] H. D. H. Velma R. Huskey, “Lady Lovelace and Charles Babbage,” *Annals of the History of Computing*, pp. 299-329, Outubro-Dezembro 1980.
- [3] “O fabuloso mercado da programação,” *Revista EXAME*, 03 Novembro 2016. [Online]. Available: <https://exame.abril.com.br/pme/o-fabuloso-mercado-da-programacao/>. [Acesso em 05 Julho 2019].
- [4] “CAPACITAÇÃO SOFTEX,” Softex, [Online]. Available: <https://softex.br/capacitacao/>. [Acesso em 05 Julho 2019].
- [5] “Developer Survey Results 2018,” *StackOverflow*, 01 Janeiro 2018. [Online]. Available: <https://insights.stackoverflow.com/survey/2018>. [Acesso em 06 Julho 2019].
- [6] T. M. Putra, “Steve Jobs: “Everybody in this country should learn to program a computer, because it teaches you how to think”,” *Bamboo Innovator*, 30 Novembro 2013. [Online]. Available: <https://bambooinnovator.com/2013/11/30/steve-jobs-everybody-in-this-country-should-learn-to-program-a-computer-because-it-teaches-you-how-to-think/>. [Acesso em 07 Julho 2019].
- [7] G. A. D. J. P. d. S. Zacarias M. Chamberlain Pravia, “COMPUTAÇÃO NA ENGENHARIA: ENSINAR A PROGRAMAR E/OU,” *Congresso Brasileiro de Ensino de Engenharia*, Setembro 2005.
- [8] H. M. E. M. N. O. J. N. Mahfudzah Othman, “Enhancing Logical Thinking and Reasoning Skills Through,” *Colloquium in Computer and Mathematical Sciences Education*, pp. 12-17, 1 Agosto 2015.
- [9] M. O. Umi Hanim Binti Mazlan, “Fundamentals of Algorithm Design Course: Issues, Challenges & Proposed Teaching-Learning Approaches,” *Colloquium in Computer and Mathematical Sciences Education*, pp. 18-23, 01 Agosto 2015.