

## Capítulo

# 13

## Ensino de Programação (e Computação) na Era da IA Generativa

Rodolfo Azevedo, Jacques Wainer

### *Resumo*

*O surgimento da IA generativa está desafiando profundamente as metodologias tradicionais de ensino de programação. Ferramentas como assistentes de IA são capazes de transformar enunciados de problemas diretamente em códigos funcionais, levando à obsolescência dos modelos pedagógicos baseados na progressão incremental de complexidade. Isso exige uma adaptação urgente da comunidade acadêmica para manter a relevância e eficácia do ensino de programação. Este desafio propõe a reformulação do ensino em dois eixos principais: a adaptação das disciplinas de programação à IA generativa e o desenvolvimento de habilidades diferenciadas para os programadores da era da IA, com a potencial consequência em todo o currículo de computação. A experiência prática conduzida pelos autores revelou aprendizados valiosos, como a importância da decomposição de problemas e a necessidade de ensinar teste e depuração de código. A aplicação de IA nas aulas resultou em um aumento na motivação dos alunos e em índices de aprovação superiores. No entanto, o desafio permanece em criar atividades que não possam ser resolvidas automaticamente pela IA, garantindo que o aprendizado continue sendo um processo ativo e enriquecedor. Este artigo desafia a comunidade a colaborar na busca por soluções inovadoras para essas questões, enquanto nos preparamos para uma realidade onde a IA desempenhará um papel central na educação em computação.*

### **13.1. Introdução**

Ao ensinar programação, estamos ensinando simultaneamente técnicas de resolução de problemas e formas de expressão em linguagem de programação. Como consequência, o resultado das atividades de um aluno também representa a sequência mental que ele desenvolveu durante o processo de criação da solução. Posteriormente, a principal forma de verificação desse modelo mental é através da correta execução do programa desenvolvido, tipicamente utilizando entradas padronizadas e esperando as saídas desejadas pelo programa. A metodologia das atividades práticas de computação coíbe plágio criando novas atividades a cada semestre e checando similaridade entre as soluções submetidas.

Nos anos recentes, surgiram ferramentas que utilizam IA Generativa e são capazes de transformar o enunciado de um problema de introdução à programação em um código que é aprovado nos corretores automáticos. Uma análise preliminar, feita pelos autores, indicou que metade dos problemas de programação utilizados numa disciplina foram resolvidos apenas copiando o enunciado e utilizando a solução proposta. Para os demais problemas, algumas simples interações extras foram capazes de cobrir metade das atividades faltantes, índice suficiente para aprovação e que deve ser ampliado com o avanço das ferramentas de IA.

A consolidação do uso de IA tornará essa metodologia tradicional de ensino de programação obsoleta rapidamente. É crucial que a comunidade de computação realize múltiplos experimentos e compartilhe os resultados de novas técnicas de ensino/aprendizagem para mitigar os desafios impostos por essas tecnologias emergentes. A colaboração urgente é necessária para desenvolver estratégias inovadoras e eficazes para adaptar o currículo e procedimentos de avaliação aos avanços da IA.

Diante desse cenário, os autores desse desafio realizaram uma oferta da disciplina de Introdução a programação com a utilização de assistentes de IA para alunos ingressantes do curso de Engenharia de Alimentos, que será detalhada na Seção 1.3. Os aprendizados e questões abertas por essa disciplina nos leva a propor o desafio de **“Reformular o ensino de Programação de Computadores face ao surgimento dos Assistentes de IA”**. Acreditamos que existam 2 sub-temas a serem abordados pela comunidade nesse desafio:

1. Adaptar o ensino de programação à IA generativa
2. Como desenvolver habilidades para ser um programador competente na era da IA generativa

Os temas refletem o ensino de programação e, potencialmente, as novas técnicas serão refletidas em todo o curso de computação. As próximas seções apresentarão uma visão mais aprofundada do modelo atual (Seção 1.2), da experiência motivadora com a nova edição da disciplina (Seção 1.3) e com os 2 temas que acreditamos que a comunidade de computação precisará direcionar esforços (Seção 1.4) com urgência.

## 13.2. Métodos Tradicionais de Ensino de Programação

O ensino de programação tradicionalmente segue uma abordagem estruturada e incremental, projetada para construir gradualmente o conhecimento e as habilidades dos alunos. Este método se baseia na introdução progressiva de conceitos, acompanhada por exercícios práticos cuidadosamente elaborados. Em cada aula, novos conceitos ou comandos de programação são apresentados aos alunos, seguindo uma ordem que permite uma progressão lógica do mais simples ao mais complexo, com tarefas desenvolvidas especificamente para cada parte do processo. Essas tarefas são criadas de forma a permitir que os alunos utilizem conceitos já aprendidos de maneira mais sofisticada, enquanto aplicam o novo conceito de forma simples. Assim, exercícios que estão centrados em exemplos de *loop* em programação básica, já supõem que os alunos podem usar comandos de *ifs* complexos (*if-elseif* ou *ifs* aninhados) com expressões de teste complexas.

A complexidade dos problemas é cuidadosamente calibrada, especialmente no início dos cursos de programação. Os exercícios são mantidos relativamente simples para não sobrecarregar os alunos, mas são projetados para desafiar e expandir suas habilidades de forma gradual. Esta abordagem permite que os estudantes consolidem seu entendimento de conceitos fundamentais antes de prosseguir para tópicos mais avançados.

A metodologia de resolução mais comumente empregada envolve fornecer aos alunos apenas a especificação do problema e, possivelmente, alguns casos de teste. Espera-se que os alunos desenvolvam o programa completo por conta própria, aplicando os conceitos aprendidos. Esta abordagem de “programação a partir da especificação” parece ser considerada valiosa por várias razões: acreditamos ela promove uma compreensão profunda dos conceitos, acreditamos que ela desenvolve habilidades cruciais de resolução de problemas e provavelmente erradamente, acreditamos que prepara os alunos para cenários do mundo real, onde os programadores frequentemente trabalham com requisitos e não com códigos pré-existentes.

O surgimento das ferramentas de IA Generativa é uma ameaça clara a esta metodologia tradicional e provocou reações distintas entre os membros da comunidade, situando-os entre dois extremos:

**Proibir o uso de IA generativa:** Essa estratégia se baseia na premissa de que é possível criar mecanismos eficazes de detecção de soluções geradas por IA, de modo a garantir a integridade do trabalho dos estudantes. No entanto, essa solução enfrenta algumas dificuldades. Primeiro, proibir completamente o uso de IA generativa pode ser uma medida controversa, uma vez que os alunos formados nessas disciplinas irão, inevitavelmente, se deparar com essas ferramentas em seu dia a dia profissional. Negar-lhes a oportunidade de aprender a usar a IA de forma responsável e produtiva pode ser visto como uma abordagem inflexível e desconectada da realidade do mercado de trabalho. Além disso, a capacidade de detectar com precisão o uso de IA generativa em trabalhos acadêmicos ainda é um desafio em evolução. À medida que a tecnologia avança, os sistemas de detecção também precisarão ser constantemente atualizados, o que pode se tornar uma tarefa árdua e onerosa para as instituições de ensino.

**Adaptar o ensino para incentivar o uso responsável da IA:** Essa estratégia reconhece que a IA é uma realidade presente e futura no campo da programação, e que os estudantes precisarão aprender a integrá-la de forma eficaz em seu fluxo de trabalho. Os educadores podem desenvolver exercícios e projetos que exijam dos alunos não apenas a resolução de problemas por conta própria, mas também a capacidade de avaliar quando e como usar ferramentas de IA de modo apropriado. Acreditamos que essa seja a melhor abordagem e como implementá-la ainda é um problema aberto, que deve levar em conta: dependência da sorte, avaliação da capacidade do aluno ler e entender o código gerado, avaliação da capacidade do aluno testar e depurar o código gerado, e a aquisição do pensamento algorítmico.

### **13.3. Aprendizados da oferta de Introdução à Programação com IA**

Durante o primeiro semestre de 2024, os autores dessa proposta ofertaram uma turma de introdução à programação utilizando assistentes de IA. Em especial, algumas mudanças foram realizadas na programação da disciplina original para construção dessa nova edição, dentre elas: 1) Todos os tópicos da disciplina original foram cobertos, em um intervalo de tempo menor para cada parte 2) Todos os componentes da linguagem de programação foram apresentados, já que o programa gerado pelo assistente de IA pode incluir esses componentes. A ênfase da disciplina não foi em ensinar os alunos a gerar o programa mas sim ler/entender o programa gerado pelo assistente.

Uma ênfase importante da disciplina é o ensino da decomposição de problemas em subproblemas menores, estes sim capazes de serem resolvidos total ou parcialmente pelos assistentes de IA.

Como primeira edição da disciplina, notamos temas que afetam diretamente o ensino para novatos: 1) As respostas não são determinísticas, tomando mais interações de alguns alunos em relação a outros, o que precisa ser analisado como um problema de justiça. 2) O grau de dificuldade de uma especificação para um humano não é o mesmo que para IA e construir um conjunto de atividades em grau crescente de dificuldade para a IA é bastante desafiador. 3) É muito importante ensinar testes e convencer os alunos que eles não devem confiar cegamente no programa gerado. 4) É muito importante ensinar depuração e não fizemos isso de forma apropriada nessa primeira edição.

Os resultados da disciplina foram avaliados comparativamente com os alunos ingressantes de 1 ano antes, do mesmo curso e que participaram da edição tradicional da disciplina, em 3 métricas: 1) Utilizamos 3 atividades de laboratório da versão tradicional na versão de IA, selecionados por não serem resolvidos apenas com copiar e colar do enunciado. Nas duas primeiras atividades, a diferença entre acertos entre as duas turmas ficou abaixo de 5 %. Na última atividade, houve uma grande diferença de 80% vs 8% para a turma com IA. 2) Os alunos da edição tradicional apresentam tendência de parar de enviar atividades ao conseguirem nota de aprovação, fato que não aconteceu na edição com IA, que creditamos a uma maior motivação para resolver problemas mais complexos. 3) Ao final do semestre, 96% dos alunos da versão com IA foram aprovados vs 82% da tradicional.

Além de disciplinas introdutórias, o modelo de avaliação de programas também é utilizado em diversas disciplinas durante um curso e pode afetar claramente os conceitos em múltiplas disciplinas e linguagens como: Orientação a Objetos (Java), Circuitos Digitais (Verilog), Sistemas Embarcados (C e Assembly), Programação Paralela e Distribuída (C e Cuda), entre outras.

### **13.4. Desafio: Reformular o ensino de Programação de Computadores face ao surgimento dos Assistentes de IA**

Diante do cenário apresentado, dividimos o desafio em 2 partes que precisam de atenção e engajamento da comunidade com urgência na busca e compartilhamento de experiências e metodologias:

### **13.4.1. Adaptar o ensino de programação à IA generativa**

Existe um novo conjunto de ferramentas disponíveis tanto para os professores quanto para os alunos e que pode ocasionar um grande ganho de produtividade ou tornar a tarefa de programação desnecessária. É importante explorar as alternativas de implementação de disciplinas que envolvam tarefas de programação, seja através de maiores projetos, aprendizagem ativa, sala de aula invertida, ou outras. A nossa experiência indicou que problemas maiores nos leva a ensinar melhor as técnicas de decomposição, que são cruciais para o ensino de abstração e encapsulamento nas linguagens modernas. Algumas perguntas a considerar: 1) Qual deve ser o programa dessa nova disciplina de introdução? 2) Como criar atividades relevantes que gerem o efetivo aprendizado dos alunos e não uma resposta automática do assistente? 3) Qual o impacto dessa nova habilidade no restante do curso?

Nosso desafio para a comunidade é experimentar diferentes abordagens no uso de assistentes de IA no ensino de programação (em diferentes níveis) e reportar os resultados e lições aprendidas. Infelizmente esse desafio é urgente e precisa ser atacado no próximo ano.

### **13.4.2. Como desenvolver habilidades para ser um programador competente na era da IA generativa?**

O código gerado automaticamente por sistemas de IA generativa pode ter qualidade inferior ao código desenvolvido por programadores competentes. Os múltiplos relatos existentes em competições indicam a capacidade de superar 70% a 90% dos programadores. Como formar um programador *top 10%* sem passar pelas capacidades inferiores?

Nos métodos tradicionais de ensino de programação, os instrutores sabiam como guiar os alunos para se tornarem programadores competentes. Eles projetavam uma sequência de problemas cada vez mais desafiadores, que exigiam o domínio gradual de conceitos e habilidades essenciais. No entanto, em um cenário em que os alunos aprendem a programar usando soluções geradas pela IA, não está claro se eles realmente estão desenvolvendo a mesma proficiência que seria alcançada pelo método tradicional.

O código gerado automaticamente pela IA, embora possa resolver problemas de forma funcional, pode carecer de características importantes, como legibilidade, eficiência, modularidade e manutenibilidade. Essas qualidades são fundamentais para o desenvolvimento de programas robustos e sustentáveis no mundo real. Ao mesmo lado, talvez esses mesmos programas sejam facilmente interpretáveis pela IA, de forma a tornar irrelevante todas as técnicas de garantia de manutenibilidade da comunidade.

Nosso desafio para a comunidade é de novo experimentar diferentes abordagens no uso de assistentes de IA no ensino de programação e reportar os efeitos em programadores de maior eficiência, por exemplo times de maratonas de programação. Este desafio não é tão urgente e pode ser desenvolvido num período de até 5 anos.

Acreditamos que ambos os sub-desafios nos levarão a repensar de forma profunda o ensino de computação como um todo nos próximos anos e o papel da computação em nossa sociedade.