

# LLMs na Educação em Programação: Estratégias para Avaliação e Feedback Formativo

Francisco Genivan Silva<sup>1,2</sup>, Eduardo H. S. Aranha<sup>1</sup>,

<sup>1</sup>Programa de Pós-Graduação em Sistemas Computacionais – PPgSC/UFRN

<sup>2</sup>Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte - IFRN

<sup>3</sup>Instituto Metrópole Digital – IMD/UFRN

**Abstract.** Programming education faces challenges such as concept assimilation difficulties and the workload on instructors for grading exercises. Large Language Models (LLMs) emerge as an alternative for providing immediate and adaptive feedback. This research investigates the feasibility of LLMs in code assessment and formative feedback, analyzing pedagogical and technical aspects. Experimental studies evaluate the effectiveness of this approach, focusing on prompt engineering and computational optimization. The goal is to develop guidelines for efficiently implementing LLMs in programming education, ensuring scalability and pedagogical alignment.

**Resumo.** O ensino de programação enfrenta desafios como dificuldades na assimilação de conceitos e a sobrecarga dos professores na correção de exercícios. Modelos de Linguagem de Grande Escala (LLMs) surgem como alternativa para oferecer feedback imediato e adaptativo. Esta pesquisa investiga a viabilidade dos LLMs na avaliação de código e feedback formativo, analisando aspectos pedagógicos e técnicos. Estudos experimentais avaliam a eficácia da abordagem, incluindo engenharia de prompts e otimização computacional. O objetivo é desenvolver diretrizes para implementação eficiente de LLMs na educação em programação, garantindo escalabilidade e alinhamento pedagógico.

## 1. Caracterização do problema

O ensino de programação para iniciantes apresenta desafios tanto para alunos quanto para professores. Os estudantes enfrentam dificuldades na transição para a lógica formal da programação, exigindo habilidades como abstração e depuração, além de lidar com a falta de base matemática e o ritmo acelerado dos cursos, o que pode gerar frustração. Já os professores lidam com a sobrecarga de trabalho devido à correção de exercícios e à necessidade de feedback detalhado, especialmente em turmas numerosas, o que compromete a personalização do ensino e impacta negativamente a progressão e a retenção dos conceitos pelos alunos.

Com o avanço dos Modelos de Linguagem de Grande Escala (LLMs), novas possibilidades surgem para tornar a avaliação mais contextualizada e adaptativa. Essas tecnologias permitem análises detalhadas do código, explicação de erros e sugestões de melhorias, indo além da mera verificação de saída esperada [Mcgowan et al. 2024; Rahman and Watanobe 2023]. Estudos indicam que, quando bem aplicados, os LLMs favorecem a autonomia dos alunos e incentivam a reflexão crítica sobre os problemas [Al-Hossami et al. 2024].

Entretanto, o uso dessas ferramentas também apresenta desafios. Modelos de IA podem reforçar padrões inadequados se não forem calibrados corretamente e há o risco de dependência excessiva por parte dos estudantes, comprometendo o desenvolvimento do pensamento computacional [Joshi et al. 2024]. Questões éticas, como privacidade dos dados e viés algorítmico, também precisam ser abordadas para garantir o uso responsável dessas tecnologias no ensino.

Assim, o desafio não está apenas na adoção dos LLMs, mas no desenvolvimento de soluções que equilibrem custo, escalabilidade e impacto educacional. A engenharia de prompts e a estruturação de respostas pedagógicas eficazes são essenciais para que essas ferramentas promovam um aprendizado significativo e personalizado. Assim, com o intuito de delimitar e criar um ponto de partida para a investigação sobre o uso de LLMs na análise, avaliação e feedback de código no ensino de programação, definimos três Questões de Pesquisa (QP):

- QP1: Como a utilização de LLMs na avaliação de código influencia o processo de aprendizagem de estudantes iniciantes em programação?
- QP2: Quais estratégias de engenharia de prompt e ajustes metodológicos podem tornar o feedback gerado por LLMs mais pedagógico, eficaz e alinhado às necessidades dos estudantes?
- QP3: Quais são os desafios técnicos e pedagógicos envolvidos na implementação de um sistema de feedback automático baseado em LLMs, e quais soluções podem ser adotadas para garantir escalabilidade e viabilidade prática em ambientes educacionais?

## 2. Fundamentação Teórica

Os LLMs representam um avanço significativo na Inteligência Artificial aplicada à educação. Seu desenvolvimento se apoia em arquiteturas como Transformers [Bengesi et al. 2024] , que aprimoram a capacidade dos modelos de processar e gerar linguagem natural de forma coerente e contextualizada. Inicialmente, redes neurais profundas eram utilizadas principalmente para reconhecimento de padrões, mas a introdução dos Transformers e o pré-treinamento massivo permitiram que esses modelos fossem capazes de gerar respostas sofisticadas e altamente adaptativas [Brown et al. 2020] .

No contexto educacional, os LLMs são empregados para suporte automatizado, feedback inteligente e personalização do aprendizado. Modelos como o ChatGPT têm sido explorados para auxiliar estudantes, fornecendo respostas ajustadas ao seu nível de compreensão e permitindo interações dinâmicas [Kasneci et al. 2023]. Estudos mostram que essas ferramentas podem reduzir a carga dos professores e fornecer um suporte mais ágil e acessível aos alunos, desde que seu uso seja orientado pedagogicamente [Jukiewicz 2024]

No ensino de programação, os LLMs podem aprimorar a avaliação ao interpretar não apenas a saída do código, mas também sua estrutura lógica e estilo. Isso possibilita uma correção mais qualitativa, destacando boas práticas e fornecendo feedback instrutivo sobre erros comuns. No entanto, a utilização desses modelos apresenta desafios, como a necessidade de engenharia de prompt refinada para garantir que o

feedback gerado seja útil e pedagógico. Além disso, os LLMs operam como sistemas de "caixa-preta", dificultando a compreensão exata de seus processos decisórios, o que pode comprometer a confiabilidade do feedback.

Questões éticas também são um ponto central na adoção dessas tecnologias. O risco de dependência excessiva dos alunos, a necessidade de garantir a privacidade dos dados e a possibilidade de viés algorítmico exigem abordagens criteriosas para sua implementação. Assim, o foco deve estar na criação de metodologias que utilizem os LLMs de forma equilibrada, promovendo autonomia e reflexão crítica nos estudantes.

### **3. Trabalhos Relacionados**

A aplicação de IA na avaliação de código tem sido amplamente explorada nos últimos anos. Estudos apontam três principais abordagens para o uso de LLMs nessa área: geração de feedback formativo, correção automatizada de código e avaliação de qualidade e estilo.

Trabalhos como o de Hou et al. (2024) apresentam ferramentas baseadas em LLMs para fornecer suporte interativo a estudantes iniciantes, personalizando sugestões de acordo com suas dificuldades. Al-Hossami et al. (2024) investigaram o uso do método socrático em assistentes baseados em IA, visando estimular a autonomia dos alunos na depuração de código. Esses estudos indicam que abordagens interativas podem aumentar o engajamento e melhorar a retenção do conhecimento.

No campo da correção automatizada, Zhang et al. (2024) desenvolveram o PyDex, uma ferramenta baseada em Codex para detectar e corrigir erros em programas Python de estudantes iniciantes. A abordagem mostrou taxas elevadas de sucesso na correção de código, mas carece de explicações detalhadas sobre os erros, o que limita sua aplicação pedagógica. Smolić et al. (2024) investigaram o uso de LLMs na revisão automática de código, destacando que, embora esses modelos possam acelerar o processo de avaliação, ainda há desafios na precisão e consistência das respostas.

Diante desses estudos, percebe-se que os LLMs oferecem um potencial significativo para otimizar a avaliação de código, mas ainda apresentam limitações que exigem ajustes metodológicos. A pesquisa proposta se insere nesse contexto, buscando desenvolver estratégias para aprimorar a qualidade e confiabilidade do feedback gerado por essas ferramentas.

### **4. Metodologia**

A metodologia adotada nesta pesquisa segue um conjunto estruturado de etapas progressivas, que incluem a revisão da literatura, estudos experimentais e refinamento das abordagens utilizadas para garantir uma aplicação eficaz dos LLM na análise e correção de código no ensino de programação.

A primeira fase consiste na Revisão Sistemática da Literatura (RSL), atualmente em andamento, cujo objetivo é mapear pesquisas que exploram o uso de IA na análise e avaliação de código. A revisão busca identificar os desafios, benefícios e limitações

dessas abordagens, permitindo uma compreensão abrangente do estado da arte e fundamentando as etapas subsequentes da pesquisa

Após a revisão da literatura, será conduzido um estudo exploratório com aproximadamente 300 alunos, que visa avaliar a viabilidade da utilização de LLMs para análise de código e fornecimento de feedback formativo. Para isso, foi desenvolvida uma ferramenta específica que permite a análise do código submetido pelos estudantes em tempo real, utilizando técnicas de processamento e interpretação baseadas em modelos de linguagem avançados. O estudo investigará a aceitação e usabilidade da ferramenta, sua integração ao ambiente de ensino e a eficácia do modelo em gerar respostas coerentes. Para isso, a ferramenta inclui mecanismos de coleta de dados que permitem acompanhar a interação dos alunos com o sistema e analisar sua evolução no aprendizado.

Com base nos achados desse primeiro estudo, será conduzida uma segunda fase experimental para refinamento das abordagens utilizadas e otimização do modelo. Esse novo estudo comparará diferentes estratégias, como engenharia de prompt avançada e o uso de Recuperação Aumentada por Geração (RAG) para melhorar a qualidade do feedback gerado pelos LLMs, minimizando erros e reduzindo custos operacionais. Essa fase permitirá um aprofundamento metodológico na personalização do feedback e ajustes nas interações para melhorar a experiência do usuário e a eficácia da ferramenta.

Por fim, será realizado um estudo de validação em larga escala, no qual a ferramenta será aplicada em um número significativamente maior de turmas, em diferentes níveis e instituições de ensino, para avaliar a escalabilidade da solução e consolidar os artefatos desenvolvidos ao longo da pesquisa. Essa etapa visa validar o modelo de uso de LLMs para feedback automático em um contexto real e diversificado, analisando desafios técnicos, impactos pedagógicos e viabilidade econômica da solução.

## 5. Resultados obtidos até o momento

Até o presente estágio da pesquisa, dois resultados concretos se destacam: a revisão sistemática em andamento sobre avaliação de código utilizando IA e o desenvolvimento de uma aplicação capaz de analisar o código dos estudantes e fornecer feedback estruturado. Essa ferramenta será utilizada em estudos experimentais, sendo o primeiro estudo piloto previsto para o primeiro semestre de 2024.

A aplicação desenvolvida foi projetada para atender tanto a requisitos técnicos de uso eficiente de LLMs quanto a princípios pedagógicos que favorecem a aprendizagem significativa. O modelo adotado até o momento é o **GPT-3.5 Turbo**, escolhido devido ao seu equilíbrio entre custo, desempenho e qualidade das respostas.

O diferencial da aplicação desenvolvida está em sua abordagem pedagógica, que evita apontar diretamente os erros no código do aluno. Em vez disso, o assistente fornece dicas abstratas, seguindo um princípio educativo que incentiva a descoberta guiada, como pode ser visto na interação abaixo:

Código com erro:

```
valores = [10, 20, 30]
print('Último valor:', valores[3]) # Erro: índice inválido
```

A ferramenta apresentou o feedback:

"Seu código apresenta uma questão relacionada à forma como acessamos elementos em uma lista. Lembre-se de que, em Python, os índices começam em 0. Reflita sobre qual seria a posição do último elemento em uma lista de três itens."

Essa abordagem é sustentada por um prompt estruturado com diretrizes rígidas, como:

1. Não aponte diretamente o erro.
2. Associe o erro a um conceito abstrato (ex: índices, loops, funções).
3. Ajude o aluno a pensar sobre o problema

A ferramenta evita respostas diretas e permite que o aluno solicite reexplicações por meio de uma interface que envia automaticamente a mensagem "*Explique melhor esta dica*", garantindo um aprofundamento gradual na compreensão do erro. Além disso, oferece sugestões de melhorias relacionadas a boas práticas e estilo de programação, mas apenas quando o código submetido está livre de erros, assegurando uma aprendizagem estruturada e evitando sobrecarga de informações que poderia prejudicar a assimilação do conteúdo.

A análise inicial do uso da ferramenta sugere que a abordagem adotada é eficaz em evitar respostas diretas e estimular o raciocínio dos estudantes, contribuindo para um aprendizado mais estruturado. Nos próximos estudos, serão analisadas métricas relacionadas à eficácia das dicas fornecidas, ao impacto no tempo de aprendizado e ao nível de aceitação da ferramenta por alunos e professores.

## 6. Contribuições esperadas

A pesquisa proposta tem o potencial de trazer avanços significativos tanto para a teoria quanto para a prática no ensino de programação assistido por IA. Um dos principais resultados esperados é a construção e validação de um framework estruturado para o uso de LLMs na correção e avaliação de código, fornecendo diretrizes que permitam a replicação e adaptação da abordagem em diferentes contextos educacionais. Esse framework poderá servir como base para a criação de novas soluções baseadas em IA, aprimorando a maneira como a tecnologia é integrada ao processo de ensino e aprendizagem.

Além disso, a pesquisa aprofundará o estudo sobre engenharia de prompt e adaptação de modelos gerativos ao ensino de programação, estabelecendo melhores práticas para garantir que o feedback fornecido pelos modelos seja pedagógico, coerente e eficaz na construção do conhecimento dos alunos.

Outro impacto esperado é a melhoria no suporte oferecido aos professores e instrutores, aliviando a carga de trabalho associada à correção manual de exercícios de programação. Com um sistema automatizado capaz de fornecer feedback detalhado e personalizado, os docentes poderão redirecionar seus esforços para atividades pedagógicas mais estratégicas, como mentorias individuais e acompanhamento do progresso dos alunos. Isso permitirá um ensino mais eficiente e adaptado às dificuldades específicas de cada estudante.

Para os alunos, a ferramenta contribuirá para o desenvolvimento de habilidades fundamentais de depuração e pensamento computacional, incentivando uma abordagem

mais ativa no processo de aprendizagem. O modelo adotado favorece a exploração e a compreensão dos erros, ao invés da simples correção, permitindo que os estudantes desenvolvam maior autonomia na resolução de problemas. Como consequência, espera-se uma redução nos índices de frustração e evasão em disciplinas introdutórias de programação, promovendo um ambiente de aprendizagem mais acessível e motivador.

## **7. Dificuldades e desafios da pesquisa**

A pesquisa enfrenta desafios como a imprevisibilidade das respostas dos LLMs, a necessidade de engenharia de prompt precisa e a dependência de infraestrutura computacional robusta para uso em larga escala.

Outro obstáculo importante é equilibrar a oferta de um feedback instrutivo e detalhado sem tornar a ferramenta uma substituta direta para o processo de aprendizado ativo dos estudantes. A adaptação das respostas para cada nível de usuário, evitando simplificações excessivas ou explicações avançadas demais, continua sendo um ponto de atenção.

Além disso, há desafios na integração pedagógica, garantindo que a ferramenta estimule a autonomia dos alunos em vez de promover dependência excessiva. A viabilidade econômica e a ética no uso de IA na educação também exigem análise detalhada, especialmente no que diz respeito à privacidade dos dados e ao viés algorítmico presente nos modelos de linguagem.

## **8. Cronograma**

As atividades fundamentais executadas no estágio atual da pesquisa, assim como o planejamento das etapas subsequentes, encontram-se detalhadas na Tabela 1. A divisão por semestres permite não apenas a sincronização com o calendário acadêmico, mas também viabiliza uma distribuição equilibrada das tarefas investigativas.

**Tabela 1 - Cronograma da pesquisa**

## References

- Al-Hossami, E., Bunescu, R., Smith, J. and Teehan, R. (2024). Can Language Models Employ the Socratic Method? Experiments with Code Debugging. . <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85189288296&doi=10.1145%2f3626252.3630799&partnerID=40&md5=594c0b6805a93e1c2e7283afff9c8860>.
- Bengesi, S., El-Sayed, H., Sarker, M. K., et al. (2024). Advancements in Generative AI: A Comprehensive Review of GANs, GPT, Autoencoders, Diffusion Model, and Transformers. *IEEE Access*, v. 12, p. 69812–69837.
- Brown, T. B., Mann, B., Ryder, N., et al. (22 jul 2020). Language Models are Few-Shot Learners. . arXiv. <http://arxiv.org/abs/2005.14165>, [accessed on Feb 9].
- Hou, X., Wu, Z., Wang, X. and Ericson, B. J. (2024). CodeTailor: LLM-Powered Personalized Parsons Puzzles for Engaging Support While Learning Programming. . <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85197053395&doi=10.1145%2f3657604.3662032&partnerID=40&md5=f06ebc9c5a14b5fda722156d27321b31>.
- Joshi, I., Budhiraja, R., Dev, H., et al. (2024). ChatGPT in the Classroom: An Analysis of Its Strengths and Weaknesses for Solving Undergraduate Computer Science Questions. . <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85189349317&doi=10.1145%2f3626252.3630803&partnerID=40&md5=d7c924a391d9d85243173ceddc23e208>.
- Jukiewicz, M. (jun 2024). The future of grading programming assignments in education: The role of ChatGPT in automating the assessment and feedback process. *Thinking Skills and Creativity*, v. 52, p. 101522.
- Kasneci, E., Sessler, K., Küchemann, S., et al. (apr 2023). ChatGPT for good? On opportunities and challenges of large language models for education. *Learning and Individual Differences*, v. 103, p. 102274.
- Mcgowan, A., Anderson, N. and Smith, C. (2024). The use of ChatGPT to generate Summative Feedback in Programming Assessments that is Consistent, Prompt, without Bias and Scalable. . <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85197514082&doi=10.1145%2f3660853.3660863&partnerID=40&md5=62dd72f53bdd17a00c3c49bb4162beca>.
- Rahman, M. M. and Watanobe, Y. (2023). ChatGPT for Education and Research: Opportunities, Threats, and Strategies. *Applied Sciences* (Switzerland), v. 13, n. 9.
- Smolić, E., Pavelić, M., Boras, B., Mekterović, I. and Jagušt, T. (2024). LLM Generative AI and Students' Exam Code Evaluation: Qualitative and Quantitative Analysis. . <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85198226508&doi=10.1109%2fMIPRO60963.2024.10569820&partnerID=40&md5=d074bc14dafe4d5c832c244fd07c34b5>.
- Zhang, J., Cambronero, J. P., Gulwani, S., et al. (2024). PyDex: Repairing Bugs in Introductory Python Assignments using LLMs. *Proceedings of the ACM on Programming Languages*, v. 8, n. OOPSLA1.