Apoio, não atalho: feedback automatizado no ensino de programação

Francisco Genivan Silva[1,2], Thiago Reis da Silva[3], Mário Jorge Ferreira Rodrigues[4], Eduardo Henrique da Silva Aranha[1]

[1]Universidade Federal do Rio Grande do Norte (UFRN)

[2]Instituto Federal do Rio Grande do Norte (IFRN) – Campus Currais Novos

[3]Instituto Federal do Maranhão (IFMA) – Campus São João dos Patos

[4]Universidade de Aveiro (UA)

genivan.silva@ifrn.edu.br,thiago.reis@ifma.edu.br

mjfr@ua.pt, eduardoaranha@dimap.ufrn.br

Abstract. *In introductory programming courses, automatic judges quickly indicate whether a solution has been accepted, but their verdict provides little guidance on what students should revise. This paper seeks to understand how an LLM-based feedback tool, integrated into an automatic grading platform, becomes part of the attempt-and-revision cycle of technical secondary education students. We analyzed platform records from two assessment sessions in which access to the tool alternated among students. The results indicate similar performance with and without feedback, but fewer attempts per problem when the tool was available. In most requests, students revised their code before submitting it again, suggesting that the tool was used to support revision. The experience illustrates one way of integrating generative AI into programming education under teacher mediation.*

Resumo. *Em disciplinas introdutórias de programação, juízes automáticos informam rapidamente se uma solução foi aceita, mas seu veredito pouco orienta o estudante sobre o que revisar. Este artigo busca compreender como uma ferramenta de feedback baseada em LLM, integrada a uma plataforma de correção automática, se articula ao ciclo de tentativa e revisão de estudantes do ensino médio técnico. Foram analisados os registros de duas sessões avaliativas, nas quais o acesso ao recurso foi alternado entre os estudantes. Os resultados indicam desempenho semelhante com ou sem o feedback, porém com menos tentativas por questão quando ele estava disponível. Na maioria dos pedidos, os estudantes revisaram o código antes de submetê-lo novamente, o que sugere o uso do recurso como apoio à revisão. A experiência ilustra uma forma de integrar a IA generativa ao ensino de programação sob mediação docente.*

1. Introdução

A correção automática de código baseada em juízes é amplamente utilizada no ensino de programação, especialmente em disciplinas introdutórias com grande volume de submissões. Esse tipo de recurso oferece retorno rápido ao estudante e ajuda o professor a acompanhar atividades práticas, mas raramente produz um retorno que ajude o

DOI: 10.5753/wie.2026.27943

aluno a compreender por que sua solução falhou. Em muitos casos, o estudante recebe apenas a indicação de que a solução foi aceita ou rejeitada, acompanhada de mensagens de erro que nem sempre são compreensíveis. Essa limitação é importante porque aprender a programar envolve revisar hipóteses, interpretar falhas, testar alternativas e construir gradualmente uma compreensão mais precisa sobre o problema [Shute 2008, Nicol and Macfarlane-Dick 2006]. Nesse sentido, o feedback tem valor formativo quando ajuda o estudante a reconhecer o que precisa melhorar e a planejar uma próxima ação [Sadler 1989, Black and Wiliam 1998].

O avanço recente dos modelos de linguagem de grande porte ampliou as possibilidades de uso de feedback automatizado em atividades de programação, permitindo a geração de orientações textuais relacionadas ao código produzido pelo estudante e ao enunciado da tarefa [Hellas et al. 2023, Kiesler et al. 2023]. Ao mesmo tempo, o uso desse tipo de tecnologia em sala de aula exige cuidado. O feedback gerado por inteligência artificial não deve substituir o acompanhamento docente nem assumir sozinho decisões avaliativas. Seu papel mais adequado é apoiar o estudante enquanto ele resolve o problema, oferecendo pistas, explicações e orientações que preservem sua autonomia e mantenham o professor como mediador da aprendizagem [Ministério da Educação 2026, Vicari et al. 2023].

Apesar do crescimento das pesquisas sobre modelos de linguagem no ensino de programação, parte importante da literatura ainda privilegia aspectos técnicos da geração de feedback, como o desenho de prompts, a qualidade das respostas e a integração com ambientes de correção automática [Silva and Aranha 2025, Pimentel et al. 2025]. São menos frequentes relatos que documentem como essas ferramentas são incorporadas a práticas concretas de sala de aula, especialmente na educação básica. Permanecem, assim, pouco visíveis aspectos que a avaliação isolada das mensagens não revela, como o momento em que os estudantes recorrem ao feedback, o que fazem depois de recebê-lo e as escolhas pedagógicas necessárias para integrar o recurso à atividade.

Este artigo relata uma experiência conduzida em uma disciplina introdutória de programação do curso técnico de informática integrado ao ensino médio. Durante duas sessões consecutivas de avaliação, os estudantes resolveram blocos equivalentes de questões, alternando entre situações com e sem acesso ao recurso. São descritos o planejamento e a realização da atividade, o que os registros da plataforma permitiram observar sobre o uso da ferramenta e os aprendizados decorrentes de sua adoção pedagógica. Mais do que avaliar isoladamente a qualidade das mensagens geradas, busca-se compreender como o feedback automatizado se articula ao ciclo de tentativa, revisão e nova submissão em uma atividade real de programação.

2. Métodos

Esta seção descreve como a experiência foi organizada em sala de aula, incluindo o contexto da turma, a plataforma utilizada, a organização das sessões, as questões propostas e a forma como os registros da plataforma foram analisados.

2.1. Contexto da disciplina e participantes

A atividade foi realizada em uma turma de disciplina introdutória de programação do curso técnico de informática integrado ao ensino médio, composta por 20 estudantes regularmente matriculados. No momento da aplicação, a turma já havia trabalhado conteúdos

relacionados a entrada e saída de dados, operações matemáticas básicas e estruturas condicionais em Python.

Por se tratar de uma aplicação em contexto real de sala de aula, o comparecimento não foi uniforme nas duas sessões de avaliação. A atividade ocorreu durante o horário regular das aulas, no laboratório utilizado pela turma ao longo do semestre, em condições próximas às práticas usuais da disciplina. Nos dias que antecederam a aplicação, os estudantes tiveram um contato breve com a plataforma de submissão e com a ferramenta de feedback automatizado, apenas para familiarização inicial com o ambiente, sem uso regular do recurso ao longo do semestre.

2.2. A plataforma e a ferramenta de feedback

A atividade foi realizada através de uma plataforma de submissão de código apresentada aos estudantes nos dias que antecederam a avaliação. No ambiente, cada estudante acessava o enunciado, escrevia sua solução em Python e submetia o programa para correção automática. A correção era feita por um juiz automático, que executava o código contra um conjunto de casos de teste e indicava quais deles haviam sido aprovados ou reprovados, exibindo também mensagens de erro quando ocorriam falhas de execução. Esse funcionamento permitia ao estudante verificar rapidamente o estado de sua solução, mas exigia atenção não apenas à lógica do programa, mas também ao formato esperado de entrada e saída.

Integrada a esse ambiente, a ferramenta de feedback formativo, desenvolvida pelos autores para uso na própria plataforma, oferecia um apoio adicional durante a resolução das questões. Entre uma submissão e outra, o estudante podia solicitar uma orientação textual automatizada, gerada a partir do enunciado, do código submetido e dos resultados produzidos pelo juiz. Na versão utilizada no estudo, essa geração foi instrumentalizada por meio da API da OpenAI, com o modelo `5.4-mini`. A mensagem era exibida junto ao código do estudante e não alterava a solução escrita por ele.

A ferramenta foi projetada para apoiar a revisão sem antecipar a solução. Em vez de apontar a linha do erro ou fornecer trechos prontos, o feedback gerado orienta o próximo passo de raciocínio do estudante, chamando atenção para a natureza do problema, para uma regra do enunciado a rever ou para algum aspecto da saída esperada. Essa escolha deliberada preserva a autonomia do estudante e mantém com ele o esforço de interpretar e corrigir a solução.

O feedback apresentado ao estudante não correspondia simplesmente à saída bruta do modelo. Antes de ser exibida, a resposta passava por uma camada de controle com regras pedagógicas, restrições contra a entrega de soluções prontas e uso das evidências disponíveis sobre a submissão. O modelo também gerava um marcador interno sobre a corretude da solução, usado apenas como sinal de apoio ao controle da resposta, sem ser exibido ao estudante nem substituir a avaliação do juiz automático. A decisão sobre a corretude permaneceu vinculada aos resultados dos testes e à mediação docente. A plataforma também registrava automaticamente as interações realizadas durante a atividade, que constituíram a principal fonte de dados do estudo.

2.3. Organização da atividade

A experiência ocorreu em duas sessões avaliativas, em dias consecutivos, no mesmo horário e no laboratório usado regularmente pela turma. Em cada sessão, os estudantes resolveram um bloco de cinco questões em até noventa minutos. Toda a turma resolveu o Bloco A no primeiro dia e o Bloco B no segundo, em dois conjuntos planejados como equivalentes em conteúdo e complexidade, detalhados na próxima subseção.

Para permitir que os estudantes vivenciassem as duas situações, a turma foi organizada em dois grupos de tamanho aproximado, e o acesso à ferramenta alternou entre eles a cada dia. No primeiro dia, um grupo resolveu o bloco com o feedback disponível e o outro sem acesso ao recurso. No segundo dia, a configuração foi invertida. Dessa forma, o arranjo previa que os estudantes presentes nas duas sessões passassem por um momento com apoio do feedback e por outro sem. A definição de qual grupo começaria com a ferramenta foi feita por sorteio, registrado em planilha.

Essa alternância atendeu a duas preocupações práticas da sala de aula. Como os dois grupos resolviam as mesmas questões no mesmo dia, reduziu-se a circulação prévia de enunciados. Além disso, como o acesso à ferramenta não ficou restrito a uma parte da turma durante toda a atividade, evitou-se que a experiência de uso do recurso ficasse concentrada em apenas um grupo. A intenção foi observar o uso do feedback em condições reais, usando a comparação entre os dois momentos como apoio para interpretar os registros produzidos pela plataforma.

No início de cada sessão, o aplicador indicou a cada estudante o ambiente correspondente ao seu grupo, confirmou o ingresso no local correto e leu uma orientação padronizada sobre o número de questões, o tempo disponível e a presença ou ausência do feedback naquela sessão. Os estudantes não receberam, de antemão, informações sobre o conteúdo das questões ou sobre a equivalência entre os blocos, e nenhum conteúdo novo sobre o tema foi apresentado entre as duas sessões.

2.4. As questões

Foram elaboradas dez questões de programação em Python, organizadas em dois blocos de cinco. Os blocos foram planejados como equivalentes em conteúdo, estrutura e complexidade. Para cada questão do Bloco A, criou-se uma correspondente no Bloco B, abordando o mesmo conceito, exigindo raciocínio semelhante e mobilizando operações análogas, ainda que com enunciados, contextos e valores diferentes.

Os cinco pares cobriam focos conceituais em ordem crescente de complexidade, apresentados na Tabela 1. O último par correspondia a uma questão-desafio, que combinava aritmética em etapas e condicionais com vários ramos, exigindo a decomposição do problema antes da escrita do código.

Em cada par, buscou-se manter próximas a quantidade de entradas, a estrutura da solução esperada e a forma da saída. Cada questão tinha exemplos de entrada e saída, gabarito de referência e casos de teste definidos previamente, usados pelo juiz automático na correção. Para gerar o feedback, a ferramenta usava o enunciado, o código submetido e os resultados dos testes como contexto. A montagem dos blocos buscou reduzir a chance de que variações de enunciado, formato de saída ou casos de teste explicassem diferenças no número de tentativas. Os enunciados completos e os casos de teste estão disponíveis junto aos artefatos.

Tabela 1. Focos conceituais das questões, em pares equivalentes entre os blocos.

Par	Foco conceitual
1	Expressões aritméticas com múltiplas operações
2	Divisão inteira e resto
3	Condicional com três faixas (`if/elif/else`)
4	Condicional com múltiplas variáveis e operadores lógicos
5	Aritmética em etapas e condicionais combinadas (desafio)

2.5. Registros da plataforma e critérios de análise

A análise partiu dos dados produzidos automaticamente durante as duas sessões. Eles incluíram tentativas dos estudantes, submissões executadas pelo juiz, resultados dos testes quando disponíveis, solicitações de feedback, código analisado e mensagens geradas. As interações foram organizadas por estudante e por questão, na ordem em que ocorreram, para reconstituir como cada resolução se desenrolou.

A leitura dos registros seguiu três movimentos complementares. O primeiro voltou-se ao uso da ferramenta nas sessões em que ela estava disponível. Foram considerados o número de estudantes que solicitaram feedback, o total de pedidos e o que ocorreu depois de cada solicitação. Para isso, cada pedido foi associado à submissão testada seguinte do mesmo estudante na mesma questão, quando existente, permitindo observar se o estudante voltava a testar o código após receber a orientação.

O segundo movimento comparou as situações com e sem acesso à ferramenta, considerando os estudantes que passaram pelas duas. Foram observados os resultados alcançados, pela taxa de resolução e pela melhor proporção de testes aprovados, e o caminho percorrido até eles, pelo número total de tentativas registradas e pelo número de tentativas testadas, isto é, submissões efetivamente avaliadas pelo juiz em cada questão.

O terceiro movimento aprofundou os episódios de revisão. Foram identificados os casos com resultado de teste comparável antes e depois do feedback, nos quais era possível observar mudança na proporção de testes aprovados entre uma tentativa e a seguinte. Também foram selecionadas trajetórias contrastantes e um episódio ilustrativo, com o objetivo de tornar mais concreta a forma como o feedback apareceu durante a atividade.

A organização dos registros buscou tornar a leitura da experiência mais cuidadosa. A alternância da ferramenta entre os grupos e a equivalência entre os blocos ajudaram a observar a atividade sem concentrar o uso do feedback em um único grupo ou em um conjunto específico de questões. Ainda assim, por se tratar de uma turma pequena e de uma aplicação em sala de aula real, os resultados são apresentados de forma descritiva. Eles não procuram demonstrar que o feedback causou aprendizagem, mas caracterizar como os estudantes usaram a ferramenta, como percorreram as questões e que aprendizados pedagógicos emergiram da atividade.

2.6. Cuidados éticos

O estudo foi conduzido observando os cuidados éticos aplicáveis a pesquisas em contexto educacional com estudantes. Os participantes assinaram termo de consentimento para uso dos registros produzidos durante a atividade, mesmo sendo a análise realizada com dados

anonimizados. Os registros considerados incluíram submissões de código, resultados dos testes, solicitações de feedback e mensagens geradas pela ferramenta.

Antes da análise, os dados foram anonimizados, com substituição de identificadores pessoais por códigos internos. A participação na pesquisa não alterou os critérios de avaliação da disciplina nem produziu consequências acadêmicas individuais. Os resultados foram analisados de forma agregada, sem identificação dos estudantes, e a análise restringiu-se aos registros necessários para compreender o uso da ferramenta durante a atividade.

3. Resultados e discussão

Como a experiência ocorreu em uma sala de aula real, nem todos os estudantes participaram das duas sessões da mesma forma. Dos 20 estudantes da turma, 18 deixaram registros válidos após a verificação dos dados. Desse conjunto, 13 atenderam simultaneamente aos critérios necessários para comparar os dois momentos da atividade, pois estiveram presentes nos dois dias e passaram tanto pela sessão com feedback quanto pela sessão sem o recurso. Os registros dos demais não entraram nessa comparação, mas ajudam a descrever a participação, o uso da ferramenta e os padrões de tentativas. A Tabela 2 resume esse recorte.

Tabela 2. Recorte analisado.

Indicador	Valor
Estudantes na turma	20
Estudantes com registros válidos	18
Estudantes comparáveis entre os dois momentos	13
Estudantes que usaram a ferramenta	8
Solicitações de feedback	80

3.1. Solicitações de feedback no ciclo de resolução

A resolução de uma questão frequentemente envolveu mais de uma passagem pelo juiz, como é comum em ambientes desse tipo. Entre as questões que os estudantes chegaram a tentar, 60,7% das resoluções sem feedback envolveram duas ou mais submissões ao juiz. Na situação com feedback disponível, esse padrão apareceu em 70,0% das resoluções. Tentar novamente, portanto, já fazia parte da dinâmica da atividade, com ou sem a ferramenta.

É nesse fluxo iterativo de trabalho que os pedidos de feedback aparecem. Dos 80 pedidos registrados, 79 foram seguidos por uma nova submissão testada do mesmo estudante na mesma questão. Como novas tentativas eram frequentes nas duas situações, esse dado caracteriza principalmente o lugar ocupado pelo feedback na trajetória de resolução. Na quase totalidade dos casos, a orientação não apareceu como uma consulta final ou desconectada da atividade, mas como parte de uma resolução que prosseguiu no juiz automático.

O momento dos pedidos reforça essa leitura. Boa parte deles ocorreu antes mesmo da primeira submissão testada da questão, quando o estudante ainda estava estruturando a lógica ou transformando sua ideia em código. Nesses casos, o feedback não serviu apenas para explicar uma falha já apontada pelo juiz, mas também para apoiar a construção da solução.

3.2. Diferenças observadas entre as situações com e sem feedback

A alternância entre sessões com e sem feedback permitiu observar como os estudantes percorreram as questões nas duas situações. Entre os 13 estudantes com registros comparáveis, consideramos indicadores de resultado, como taxa de resolução e melhor proporção de testes aprovados, e indicadores do percurso, como número de tentativas por questão e número de tentativas avaliadas pelo juiz. A leitura é descritiva e serve para caracterizar o que ocorreu nessa turma, sem tratar a comparação como medida isolada de eficácia da ferramenta.

Nos indicadores de resultado final, as diferenças foram pequenas. Considerando uma questão como resolvida quando o estudante aprovou todos os seus casos de teste em alguma tentativa, a taxa média de resolução foi de cerca de 83% tanto na situação com feedback quanto na situação sem feedback. A mediana foi de 100% nas duas situações, indicando que pelo menos metade dos estudantes resolveu todas as questões que chegou a tentar em cada uma delas. A melhor proporção média de casos de teste aprovados também foi semelhante, com 0,86 na situação com feedback e 0,85 sem feedback. Esses valores indicam que a presença da ferramenta não se traduziu em vantagem clara no resultado final da atividade.

No percurso até esse resultado, porém, apareceu uma diferença mais nítida. Com feedback disponível, os estudantes realizaram menos tentativas por questão, com mediana de 6,0, em comparação a 7,2 sem o recurso. O mesmo padrão aparece quando são consideradas apenas as tentativas testadas, cuja mediana foi de 2,6 por questão com feedback e 3,6 sem feedback. A diferença também se manteve no nível individual, já que em 9 dos 13 estudantes comparáveis a sessão sem feedback concentrou mais tentativas por questão.

Essa leitura ganha força porque os blocos foram planejados com conteúdo e estrutura equivalentes, reduzindo a chance de que o padrão resulte apenas da dificuldade das questões. O ponto central, portanto, não é afirmar um efeito mensurável da ferramenta, mas registrar que, nesta turma, o feedback esteve associado a um modo menos reiterado de percorrer as questões. O tamanho reduzido do grupo impede generalizações, mas esse achado é relevante para o relato porque ajuda a compreender como a ferramenta se integrou à dinâmica real de resolução das questões. A Tabela 3 resume esses indicadores.

Tabela 3. Desempenho e percurso nas situações com e sem feedback ($n = 13$).

Indicador	Com feedback	Sem feedback
Taxa de resolução		
Média	0,83	0,83
Mediana	1,00	1,00
Melhor proporção de testes aprovados		
Média	0,86	0,85
Mediana	1,00	1,00
Tentativas por questão		
Todas, mediana	6,0	7,2
Avaliadas pelo juiz, mediana	2,6	3,6

3.3. Trajetórias e episódios de revisão

Depois dos indicadores gerais, observamos três percursos individuais para compreender como o feedback apareceu no trabalho dos estudantes. A intenção não foi representar

a turma, mas tornar visíveis formas distintas de interação com a plataforma, com o juiz automático e com o serviço de feedback.

No primeiro percurso, o estudante E1 resolveu todas as questões que tentou com baixa necessidade de iteração. Foram 21 tentativas ao todo, com mediana de duas por questão e máximo de três em uma mesma questão. Mesmo nas questões em que a ferramenta estava disponível, não solicitou feedback. O caso ilustra uma trajetória de maior autonomia operacional, em que o estudante usou apenas os retornos do juiz automático para chegar às soluções.

O segundo percurso, associado ao estudante E2, mostra um uso intenso da ferramenta. Ele resolveu todas as questões nas duas situações, mas o caminho até a solução foi diferente. Sem feedback, realizou em média 12,4 tentativas por questão; com o recurso disponível, esse valor foi 8,0. Além disso, solicitou 17 feedbacks distribuídos pelas cinco questões da sessão em que a ferramenta estava acessível. O caso mostra um estudante que recorreu sistematicamente à orientação e reproduz, em sua própria trajetória, o padrão observado nos registros gerais.

O estudante E3 resolveu sete das dez questões observadas. Realizou 83 tentativas, com mediana de quatro tentativas por questão e máximo de 40 em uma única questão. Esse maior impasse ocorreu em uma questão na situação com feedback disponível, mas o estudante solicitou apenas um feedback automatizado. O caso sugere que disponibilizar a ferramenta não basta para garantir sua adoção nos momentos de maior dificuldade. A busca por ajuda também precisa ser compreendida como parte da cultura de trabalho construída durante a atividade.

Essas trajetórias mostram o feedback inserido no trabalho dos estudantes, mas ainda não indicam o que muda no código quando uma orientação é incorporada. Para observar essa mudança, consideramos apenas os episódios em que havia duas submissões testadas comparáveis, uma antes e outra depois do feedback, com o mesmo conjunto de casos. Esse critério reduziu a análise a 30 episódios. Neles, 16 apresentaram aumento na proporção de testes aprovados, 14 permaneceram sem mudança e nenhum registrou piora. O padrão sugere que, em parte das situações, o feedback acompanhou revisões produtivas do código. Ainda assim, a melhora observada deve ser lida junto aos demais elementos da atividade, já que o estudante também trabalhava com os retornos do juiz automático e com suas próprias hipóteses de solução.

O Quadro 1 ilustra um desses episódios em mais detalhe. A orientação recebida foi breve, não apresentou uma solução pronta e direcionou uma revisão específica no código. Após a alteração, a submissão passou a aprovar todos os casos de teste. Esse episódio torna concreto o papel observado da ferramenta nesses casos, ao apoiar o estudante na identificação de um ponto a ser revisto sem deslocar dele a tarefa de corrigir a própria solução.

4. Considerações

Este artigo relatou a aplicação de uma ferramenta de feedback automatizado baseada em LLM em uma atividade avaliativa de programação no ensino médio técnico, conduzida em uma turma iniciante, com tempo limitado, regras de avaliação e uso de juiz automático, em condições próximas às práticas usuais da disciplina.

Quadro 1: Episódio ilustrativo de feedback e reorganização da lógica.

Antes do feedback

```
if h_inicio >= 6 and h_inicio <= 20:
    valor_km = distancia * 2.80
    total = valor_km + valor_fixo
else:
  if h_inicio >= 21 and h_inicio <= 5:
        valor_km = distancia * 3.90
        total = valor_km + valor_fixo
    else:
  if h_inicio >= 6 and h_inicio <= 20 or distancia > 10:
            valor_km = (distancia * 2.80) - (distancia * (10/100))
            total = valor_km + valor_fixo
        else:
            valor_km = (distancia * 3.90) - (distancia * (10/100))
            total = valor_km + valor_fixo
```

Feedback recebido

O código está parcialmente alinhado, mas a lógica principal ainda mistura condições que deveriam ser separadas. O desconto de 10% precisa ser aplicado sobre o valor dos quilômetros depois de escolher a tarifa correta pelo horário, e só quando a distância passar de 10 km. Revise essa sequência: primeiro decidir se a corrida é diurna ou noturna, depois aplicar o desconto se a distância for maior que 10 km, e só então somar a bandeirada.

Após o feedback, trecho central

```
if h_inicio >= 6 and h_inicio <= 20:
    valor_km = distancia * 2.80
else:
    valor_km = distancia * 3.90

if distancia > 10:
    valor_km = valor_km - (valor_km * 0.10)

total = valor_km + 5.50
```

O conjunto dos registros sugere que o valor da ferramenta não deve ser observado apenas pelo desempenho final na atividade. Nas situações com e sem feedback, a taxa de resolução foi semelhante, mas a presença do recurso associou-se a um percurso mais econômico em tentativas. Quando o feedback foi solicitado, o estudante quase sempre voltou ao juiz, e em parte expressiva dos episódios revisou o código antes de submeter novamente. Esse comportamento, somado ao episódio analisado e ao desenho da ferramenta, indica que o recurso foi usado como apoio à revisão, e não como atalho para a resposta. Para o professor, o ponto é que a ferramenta atua no espaço intermediário entre tentar e ser avaliado, apoiando a reorganização da solução sem substituir o esforço do estudante.

A experiência também reforça uma escolha pedagógica importante no uso de LLMs em atividades avaliativas. A ferramenta pode reconhecer aspectos adequados do código, sugerir revisões e orientar próximos passos, mas esse retorno não precisa assumir

a forma de veredito sobre a solução. Neste relato, o LLM foi usado principalmente como apoio à revisão do código, enquanto a verificação por testes e a mediação docente permaneceram como referências para o julgamento de corretude. Essa separação ajuda a manter a IA no lugar de apoio formativo, em consonância com orientações recentes sobre o uso responsável de IA generativa na educação básica [Ministério da Educação 2026].

Como todo relato conduzido em sala de aula real, o estudo tem limitações. O número de estudantes foi reduzido, e apenas 13 atenderam aos critérios de comparação entre as situações com e sem feedback. Além disso, a participação não foi uniforme nas duas sessões, e a leitura de ganho nos testes restringiu-se aos episódios com resultados comparáveis antes e depois do feedback. A aplicação revelou ainda um ponto operacional a aprimorar, pois parte dos pedidos ocorreu sem resultado de teste associado à versão de código analisada, já que a execução dos testes dependia da submissão do estudante. Essas restrições impedem conclusões causais sobre aprendizagem e delimitam o alcance do que se pode afirmar.

Ainda assim, a contribuição do relato é clara. Em uma literatura mais voltada a aspectos técnicos da geração de feedback, ainda são escassos registros que documentem, em uma sala de aula real de educação básica, como o feedback automatizado se articula ao trabalho dos estudantes enquanto resolvem questões de programação. Como continuidade, estudos com mais turmas, maior tempo de familiarização e integração mais estreita entre execução de testes e geração de feedback podem aprofundar quando, como e para quem esse tipo de apoio produz maior benefício pedagógico.

Disponibilidade de artefatos

Os artefatos do estudo estão disponíveis em `https://github.com/genivansilva/wie2026-avallm.git`. O repositório inclui os enunciados das questões utilizadas na atividade, exemplos de entrada e saída e os casos de teste empregados pelo juiz automático.

Declaração de Uso de IA

Os autores utilizaram o ChatGPT como apoio à organização de dados, conferência de métricas descritivas e revisão linguística do manuscrito. Todas as análises, interpretações, decisões metodológicas e versões finais do texto foram revisadas e validadas pelos autores, que assumem integral responsabilidade pelo conteúdo do artigo.

Referências

Black, P. and Wiliam, D. (1998). Assessment and classroom learning. *Assessment in Education: Principles, Policy & Practice*, 5(1):7–74.

Hellas, A., Leinonen, J., Sarsa, S., Koutcheme, C., Kujanpää, L., and Sorva, J. (2023). Exploring the responses of large language models to beginner programmers' help requests. In *Proceedings of the 2023 ACM Conference on International Computing Education Research V.1 (ICER '23 V1)*. ACM.

Kiesler, N., Lohr, D., and Keuning, H. (2023). Exploring the potential of large language models to generate formative programming feedback. In *2023 IEEE Frontiers in Education Conference (FIE)*, pages 1–5.

Ministério da Educação (2026). Inteligência artificial na educação básica: Documento orientador sobre caminhos curriculares e práticas éticas de uso de ia nas escolas. Secretaria de Gestão da Informação, Inovação e Avaliação de Políticas Educacionais (SEGAPE). Disponível em: https://www.gov.br/mec/pt-br/escolas-conectadas/documentos/ia-educacao-basica.pdf.

Nicol, D. J. and Macfarlane-Dick, D. (2006). Formative assessment and self-regulated learning: a model and seven principles of good feedback practice. *Studies in Higher Education*, 31(2):199–218.

Pimentel, E. P., da Silva, L. D. L., Takeuti, R. H., and Braga, J. C. (2025). Abordagem com LLM para geração automatizada de feedback no ensino de programação de computadores. In *Anais do XXXVI Simpósio Brasileiro de Informática na Educação*, pages 591–603, Curitiba, PR, Brasil.

Sadler, D. R. (1989). Formative assessment and the design of instructional systems. *Instructional Science*, 18(2):119–144.

Shute, V. J. (2008). Focus on formative feedback. *Review of Educational Research*, 78(1):153–189.

Silva, F. G. and Aranha, E. H. S. (2025). Feedback formativo automatizado com LLMs: Desenvolvimento e análise de um sistema para aprendizagem progressiva em programação. In *Anais do XXXVI Simpósio Brasileiro de Informática na Educação*, pages 1159–1173, Curitiba, PR, Brasil.

Vicari, R., Brackmann, C., Mizusaki, L., and Galafassi, C. (2023). *Inteligência Artificial na Educação Básica: Prática na escola*. Novatec, São Paulo, 1 edition.