ReTER, Santa Maria, v.7. ISSN:2675-9950 | Submissão: 14/08/2026 | Aprovação: 07/10/2026 | Publicação: 15/10/2026

POSSIBILIDADES PEDAGÓGICAS DA INTELIGÊNCIA ARTIFICIAL NA EDUCAÇÃO COM A LABIA BLOCKS: MEDIAÇÃO E AUTONOMIA NO ENSINO DE PROGRAMAÇÃO

Tiago Saidelles
Universidade Federal de Santa Maria - tiago-saidelles@redes.ufsm.br

Miguel Augusto Bauermann Brasil
Universidade Federal de Santa Maria - miguelbrasil@ctism.ufsm.br

Resumo: A inteligência artificial generativa passou a ocupar espaço crescente em práticas educacionais, sobretudo por sua capacidade de oferecer explicações, pistas, feedback e apoio individual durante a resolução de problemas. Este estudo analisa as possibilidades pedagógicas de uso da inteligência artificial na educação por meio da LabIA Blocks, plataforma educacional voltada à iniciação em programação e criação de jogos com blocos de HTML, CSS e JavaScript. Caracteriza-se como análise de tecnologia educacional inovadora, de natureza aplicada e descritiva, baseada na inspeção funcional do editor em blocos e sua relação com o código textual, do Tutor IA, dos níveis de assistência controlados pelo professor e dos recursos de apoio ao percurso didático. Os resultados da análise indicam que a IA pode atuar como mediação contextual na compreensão de conceitos, no planejamento da lógica do jogo, na seleção e configuração de blocos, na depuração, na comparação entre blocos e código textual, na diferenciação pedagógica e na avaliação formativa. A plataforma também permite graduar o apoio da IA, favorecendo percursos com maior ou menor autonomia. Entre as potencialidades pedagógicas inferidas, destacam-se a ampliação do feedback durante a aprendizagem, o apoio à experimentação e o fortalecimento da autoria quando as respostas são testadas e justificadas pelo estudante. Como a investigação não envolveu aplicação empírica com estudantes, os achados são interpretados como possibilidades pedagógicas e não como evidências de eficácia de aprendizagem. Conclui-se que a IA pode enriquecer metodologias ativas no ensino de programação quando permanece subordinada à intencionalidade pedagógica, à supervisão docente e à verificação crítica das sugestões geradas.

Palavras-chave: Inteligência artificial na educação; Programação em blocos; Ensino de programação; LabIA Blocks; Metodologias ativas.

PEDAGOGICAL POSSIBILITIES OF ARTIFICIAL INTELLIGENCE IN EDUCATION WITH LABIA BLOCKS: MEDIATION AND AUTONOMY IN PROGRAMMING EDUCATION

Abstract: Generative artificial intelligence has gained increasing relevance in educational practices because it can provide explanations, hints, feedback, and individualized support during problem solving. This study analyzes the pedagogical possibilities of using artificial intelligence in education through LabIA Blocks, an educational platform designed for introductory programming and game creation with HTML, CSS, and JavaScript blocks. The work is characterized as an analysis of innovative educational technology, with an applied and descriptive approach based on functional inspection of the block editor and its relationship with textual code, the AI Tutor, teacher-controlled assistance levels, and resources that support the instructional process. The analysis indicates that AI can act as contextual mediation in concept understanding, game-logic planning, block selection and configuration, debugging, comparison between visual blocks and textual code, pedagogical differentiation, and formative assessment. The platform also allows the teacher to regulate AI assistance, enabling learning paths with different levels of autonomy. Inferred pedagogical potential includes broader feedback during learning, support for experimentation, and stronger student authorship when generated suggestions are tested, explained, and revised. Because the study did not include empirical application with students, the findings are interpreted as pedagogical possibilities rather than evidence of learning effectiveness. The study concludes that

AI can enrich active methodologies in programming education when its use remains guided by pedagogical purposes, teacher supervision, and critical verification of generated responses.

Keywords: Artificial intelligence in education; Block-based programming; Programming education; LabIA Blocks; Active methodologies.

1 INTRODUÇÃO

A presença da inteligência artificial generativa em contextos educacionais modifica a forma como dúvidas, feedbacks e orientações podem ser oferecidos durante uma atividade. Em vez de ocorrer somente depois da correção do professor, parte do apoio pode ser disponibilizada no próprio momento em que o estudante formula uma hipótese, encontra um erro ou tenta compreender uma estrutura de programação. Esse potencial é especialmente relevante na iniciação à computação, etapa em que conceitos abstratos e detalhes sintáticos aparecem simultaneamente e podem competir pela atenção do aprendiz. Entretanto, a mesma rapidez que torna a IA atraente também exige cuidado: uma resposta plausível pode estar incompleta, induzir dependência ou levar o estudante a aceitar uma solução sem compreender o processo que a produz. Por isso, discutir IA na educação requer observar não apenas o que o sistema consegue responder, mas como a atividade é planejada, que grau de autonomia permanece com o aluno e quais mecanismos de verificação são incorporados ao percurso.

A LabIA Blocks insere a inteligência artificial nesse cenário de iniciação à programação por meio de uma arquitetura em que blocos, código textual, execução de jogos, desafios, planejamento algorítmico e Tutor IA permanecem articulados. A plataforma organiza blocos de HTML, CSS e JavaScript e permite que o estudante observe a correspondência entre uma montagem visual e a sintaxe produzida. Paralelamente, o Tutor IA pode explicar conceitos, fornecer pistas, auxiliar na localização de erros e responder a solicitações sobre a construção de jogos utilizando os nomes reais dos blocos disponíveis. O professor controla o nível de assistência da IA ao publicar uma atividade, o que desloca a discussão de uma pergunta binária — usar ou não usar inteligência artificial — para outra mais pedagógica: em que momento, com qual intensidade e para qual objetivo o apoio automatizado deve ser disponibilizado.

A literatura recente sobre IA generativa na educação em computação identifica oportunidades relacionadas à personalização, ao apoio ao ensino, à motivação e à ampliação de conhecimento, mas também aponta riscos de imprecisão, dependência e perda de

habilidades quando a resposta automática substitui o esforço cognitivo necessário à aprendizagem (HUMBLE, 2024). Em direção semelhante, a UNESCO (2024) defende uma adoção centrada no ser humano, com validação pedagógica, atenção à privacidade e desenvolvimento de capacidades críticas. Esses princípios são relevantes para analisar ferramentas que integram IA diretamente ao ambiente de aprendizagem, pois a qualidade educacional não decorre apenas da presença da tecnologia, mas da forma como suas possibilidades são enquadradas por objetivos, limites, feedbacks e responsabilidades.

Em relação a ambientes tradicionais de programação em blocos, como o Scratch, a LabIA Blocks acrescenta um Tutor IA integrado ao editor e mecanismos de controle docente da intensidade de assistência; em relação a tutores baseados em grandes modelos de linguagem de uso genérico, procura vincular a orientação ao catálogo real de blocos e ao projeto em construção (RESNICK et al., 2009; GARCÍA-MÉNDEZ; DE ARRIBA-PÉREZ; SOMOZA-LÓPEZ, 2025). O aspecto inovador examinado neste estudo está, portanto, na articulação dessas funcionalidades com a transição entre blocos e código textual e com a gestão pedagógica do nível de apoio, sem que essa distinção de desenho seja tomada como evidência de superioridade ou de eficácia de aprendizagem.

Diante desse contexto, a questão norteadora do estudo é: de que modo as funcionalidades da LabIA Blocks, especialmente o Tutor IA e os níveis de assistência controlados pelo professor, podem apoiar a mediação pedagógica e a autonomia discente no ensino introdutório de programação sem substituir a autoria do estudante? O objetivo geral é analisar as possibilidades pedagógicas de uso da inteligência artificial na educação a partir das funcionalidades da LabIA Blocks, com ênfase no ensino introdutório de programação e criação de jogos. Como objetivos específicos, busca-se identificar como o Tutor IA pode apoiar explicação, depuração, planejamento, diferenciação pedagógica e avaliação formativa; discutir a relação entre níveis de assistência e autonomia discente; e propor formas de utilização em metodologias ativas, preservando a autoria do estudante. O escopo analítico concentra-se nas funcionalidades do artefato e em suas possibilidades pedagógicas. O trabalho não pretende demonstrar eficácia de aprendizagem, pois esta etapa da pesquisa não envolve aplicação experimental com turmas. A contribuição está em organizar, a partir da análise funcional do artefato e do diálogo com trabalhos relacionados, um conjunto de

possibilidades, limites e condições de uso que possa orientar práticas pedagógicas e pesquisas posteriores.

2 INTELIGÊNCIA ARTIFICIAL, FEEDBACK E MEDIAÇÃO NO ENSINO DE PROGRAMAÇÃO

2.1 Potenciais e riscos da IA generativa

A IA generativa pode ser compreendida, no contexto deste estudo, como recurso de mediação que produz respostas em linguagem natural a partir das solicitações do usuário. Em atividades de programação, essa capacidade abre espaço para interações diferentes das consultas tradicionais a manuais ou exemplos prontos: o estudante pode perguntar por que determinado evento não foi acionado, solicitar uma pista sobre uma condição, pedir uma explicação sobre variáveis ou comparar duas estratégias de implementação. O valor pedagógico dessas interações depende, porém, de a resposta ser incorporada a um processo de investigação. Se o aluno apenas copia uma sequência sugerida, a IA pode reduzir a atividade intelectual. Se ele interpreta a orientação, testa no editor, observa o resultado e revisa a solução, a mesma resposta pode funcionar como ponto de partida para raciocínio e depuração.

A UNESCO (2024) acrescenta a necessidade de centralidade humana, transparência de objetivos, proteção de dados e desenvolvimento de competências críticas. Em termos didáticos, isso implica tratar a IA como ferramenta sujeita a verificação e não como autoridade final. O estudante precisa aprender que uma resposta gerada deve ser confrontada com critérios, evidências e resultados observáveis. Na programação, esse princípio pode ser operacionalizado de forma concreta: o jogo executa ou não executa; a variável assume ou não o valor esperado; a colisão é ou não detectada; a condição de vitória ocorre no momento correto ou não. A presença de um artefato executável torna possível transformar a validação crítica em parte da própria atividade.

2.2 Aprendizagem ativa, gamificação e autoria apoiada

As possibilidades de IA analisadas neste trabalho ganham sentido quando articuladas a metodologias em que o estudante produz, testa e revisa uma solução. A aprendizagem ativa tende a envolver participação cognitiva mais intensa do que abordagens exclusivamente

expositivas, especialmente quando o aluno precisa tomar decisões, justificar procedimentos e responder a feedbacks durante a tarefa (FREEMAN et al., 2014). Em programação de jogos, essa participação pode ocorrer na definição de regras, na escolha de estruturas, na montagem dos blocos, na depuração e na apresentação do produto. A IA pode integrar esse percurso sem ocupar o lugar de quem decide, desde que suas intervenções sejam usadas para ampliar perguntas, oferecer pistas ou explicitar conceitos. Adota-se neste artigo a ideia de autoria apoiada. O estudante permanece responsável pelo problema, pelas escolhas e pela validação do resultado, enquanto a IA fornece apoio circunstancial. Essa concepção evita dois extremos: rejeitar qualquer uso de IA ou delegar à IA a construção integral do projeto. Entre esses polos existe uma faixa ampla de mediações possíveis, como pedir uma explicação antes de programar, solicitar uma pista após uma tentativa fracassada, comparar a solução do aluno com outra alternativa ou usar a IA para formular perguntas que ajudem a revisar a lógica. O controle pedagógico sobre essa faixa é uma das dimensões centrais analisadas na LabIA Blocks.

No componente de gamificação, adotam-se elementos de design de jogos para estruturar desafios, progressão e retorno ao estudante, em consonância com a definição de gamificação de Deterding et al. (2011). Esses elementos não constituem um fim em si mesmos: no contexto analisado, sua função é sustentar ciclos de tentativa, feedback, revisão e produção autoral, articulados às metodologias ativas.

2.3 Trabalhos relacionados e posicionamento da LabIA Blocks

Os estudos sobre programação em blocos constituem uma base direta para situar a LabIA Blocks. O Scratch consolidou uma abordagem de programação visual orientada à criação e à expressão, reduzindo parte da barreira sintática para iniciantes (RESNICK et al., 2009). No campo do pensamento computacional, Grover e Pea (2013) destacam a importância de ambientes e estratégias que apoiem a compreensão progressiva de conceitos computacionais. Weintrop e Wilensky (2018), ao compararem programação em blocos e programação textual no ensino médio, mostram que as duas representações oferecem vantagens e dificuldades distintas, o que reforça a relevância de propostas que tornem explícita a relação entre a estrutura visual e o código textual.

Em paralelo, a literatura sobre tutores baseados em grandes modelos de linguagem registra aplicações como assistência virtual, explicação e correção de código e apoio personalizado, ao mesmo tempo em que ressalta a necessidade de supervisão humana e de desenho pedagógico específico (GARCÍA-MÉNDEZ; DE ARRIBA-PÉREZ; SOMOZA-LÓPEZ, 2025; HUMBLE, 2024). Entre as tecnologias mais próximas ao objeto deste artigo, o AIBlockAgent foi desenvolvido para apoiar a aprendizagem de programação em blocos por meio de fornecimento gradual de andaimes, verificação do código em tempo real e geração de respostas apoiada por uma base de informações sobre os blocos, tendo passado por avaliação de usabilidade com professores (CHOA; KOO, 2026). Esse trabalho evidencia que a integração de IA generativa diretamente a ambientes de programação visual constitui uma linha emergente de pesquisa.

A LabIA Blocks se insere nessa linha, mas articula um conjunto particular de recursos: programação em blocos de HTML, CSS e JavaScript com visualização do código correspondente, Tutor IA contextualizado pelo catálogo real de blocos, níveis de assistência definidos pelo professor e recursos de criação de jogos, desafios e gestão de atividades. Saidelles e Brasil (2026) já apresentaram a arquitetura e as funcionalidades gerais da plataforma, com ênfase em programação em blocos, inteligência artificial e gamificação. O presente artigo delimita uma contribuição distinta: concentra a análise na IA como mediação pedagógica, na relação entre assistência e autonomia e na derivação de uma proposta didática a partir de critérios explicitados. Além dessa publicação de apresentação da plataforma, não foram identificadas, na literatura consultada para este estudo, avaliações independentes especificamente dedicadas à LabIA Blocks.

3 METODOLOGIA

O trabalho caracteriza-se como análise de tecnologia educacional inovadora, de natureza aplicada e descritiva. O objeto de análise é a LabIA Blocks como ambiente digital para iniciação em programação e criação de jogos. O escopo concentra-se na inspeção funcional do artefato e na interpretação pedagógica das funcionalidades observadas. Não foram realizados experimentos com estudantes, aplicação de questionários ou coleta de dados com seres humanos. Consequentemente, os resultados correspondem a possibilidades e limitações pedagógicas inferidas a partir das funcionalidades efetivamente disponíveis na

plataforma e de sua articulação com a literatura, sem pretensão de medir aprendizagem, engajamento ou desempenho.

A versão analisada possui catálogo de 108 blocos distribuídos entre HTML, CSS e JavaScript, e o Tutor IA utiliza esse catálogo para orientar a construção de jogos com nomes de blocos, ordem, valores e encaixes compatíveis com a interface. A versão examinada corresponde a um protótipo funcional em fase de pesquisa e testes, utilizado em ambiente controlado pelos autores. No momento da análise, a plataforma não se encontrava disponibilizada publicamente, no escopo desta pesquisa, por link de acesso aberto ou repositório de código aberto. Assim, a análise se refere especificamente à versão funcional inspecionada.

Para tornar explícitos os critérios aplicados, a análise foi organizada em quatro dimensões. No editor em blocos e sua relação com o código textual, observaram-se a correspondência entre representação visual e sintaxe, a possibilidade de execução do projeto e o apoio à transição entre as duas representações. No Tutor IA, foram considerados a aderência das orientações ao catálogo real de blocos, a pertinência contextual das respostas, os tipos de ajuda oferecidos e a possibilidade de verificar as sugestões pela execução do projeto. Nos níveis de assistência, analisaram-se o controle docente, a adequação do grau de apoio a diferentes momentos da atividade e a possibilidade de retirada progressiva do suporte. Nos recursos didáticos, examinaram-se a articulação de desafios, jogos prontos, planejamento em Portugol e acompanhamento de entregas com resolução de problemas, gamificação, autoria e avaliação formativa.

O percurso analítico compreendeu a inspeção dos fluxos representativos de criação, publicação e resolução de atividades; o registro das funcionalidades observáveis em cada uma das quatro dimensões; e a interpretação desses achados à luz da literatura sobre programação em blocos, aprendizagem ativa, gamificação e IA generativa. As Figuras 1 a 3 documentam exemplos das evidências funcionais utilizadas na análise. Não foi atribuída pontuação às dimensões: a análise é qualitativa e descritiva, orientada pelos critérios acima para identificar potencialidades, fragilidades e condições de uso pedagógico.

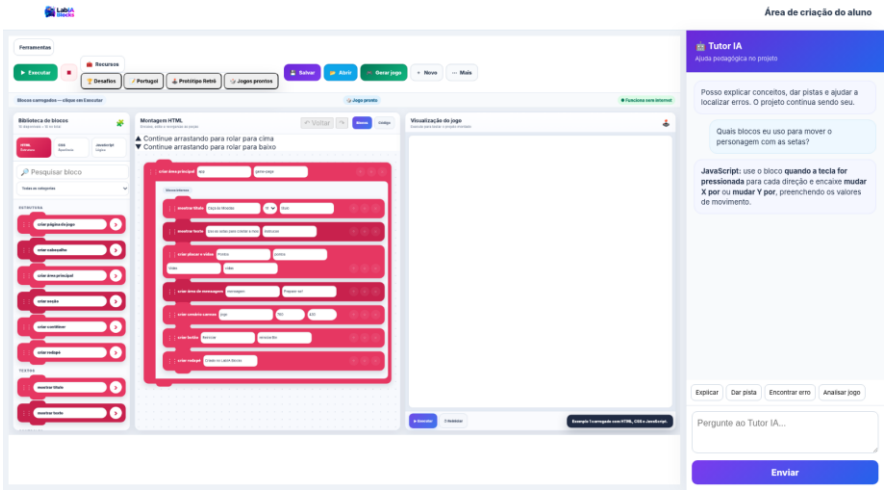
4 A LABIA BLOCKS COMO AMBIENTE PARA USO EDUCACIONAL DA IA

4.1 Tutor IA contextual e orientação vinculada ao projeto

O Tutor IA aparece integrado ao ambiente de criação, ao lado do editor no qual o estudante monta e executa o jogo. Essa proximidade altera a natureza da consulta: a dúvida não precisa ser formulada fora do contexto do projeto, pois o aluno pode perguntar sobre conceitos, blocos, erros e regras enquanto observa sua própria montagem. A interface oferece ações rápidas para explicar, dar pistas, localizar possíveis problemas e analisar o projeto, além de um campo de pergunta livre. A intenção pedagógica é oferecer apoio no momento em que a dificuldade surge, sem separar a orientação da ação que precisa ser revista.

Quando a pergunta envolve construção de jogos, o Tutor foi estruturado para usar o catálogo real da plataforma. Isso significa que a resposta pode indicar os nomes dos blocos disponíveis, a sequência em que devem ser usados, os campos que precisam ser preenchidos, valores sugeridos e relações de encaixe. A proposta procura reduzir um problema comum em sistemas generativos: recomendar comandos que não existem na interface ou entregar uma solução abstrata demais para ser aplicada. Ao relacionar o pedido a entidades, eventos, condições, ações, estados, vitória e derrota, o sistema busca transformar linguagem natural em uma orientação compatível com o repertório de blocos. A Figura 1 evidencia essa integração ao reunir, no mesmo fluxo de trabalho, o editor de blocos e o Tutor IA. Essa disposição sustenta o critério de contextualidade empregado na análise, pois a orientação pode ocorrer enquanto o estudante observa e revisa o próprio projeto.

Figura 1 – Tutor IA integrado ao ambiente de criação da LabIA Blocks

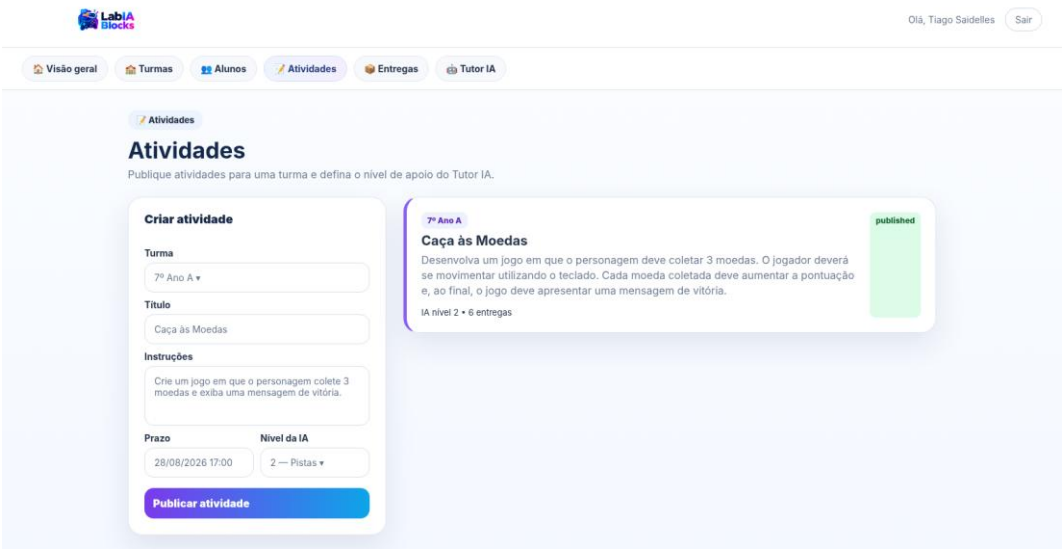


Fonte: elaboração própria a partir da versão analisada da LabIA Blocks (2026).

4.2 Níveis de apoio da IA definidos pelo professor

Um diferencial relevante para uso educacional é a possibilidade de o professor selecionar o nível de ajuda da IA ao publicar uma atividade. A plataforma prevê cinco níveis: 0 – Sem IA; 1 – Perguntas; 2 – Pistas; 3 – Tutor; e 4 – Laboratório. Essa gradação permite que a mesma ferramenta seja utilizada em propostas com graus diferentes de autonomia. Em uma atividade diagnóstica, por exemplo, o professor pode desabilitar a IA para observar o repertório inicial da turma. Em uma etapa de aprendizagem guiada, pode liberar perguntas ou pistas. Em projetos mais complexos, pode habilitar o modo Tutor ou Laboratório para ampliar a exploração. Como mostra a Figura 2, o nível de apoio é definido no próprio fluxo de publicação da atividade, junto a prazo e instruções. Esse vínculo operacionaliza o controle docente analisado, permitindo associar a intensidade da IA aos objetivos de cada tarefa. A existência de níveis não elimina a necessidade de mediação. O professor continua responsável por formular o desafio, definir critérios de sucesso, observar as entregas e interpretar o uso que o estudante faz das sugestões recebidas. O nível de IA deve ser compreendido como parte do planejamento, assim como tempo, materiais, agrupamento e complexidade da tarefa. Essa leitura evita que a inteligência artificial seja tratada como adição automática à aula e favorece decisões mais intencionais sobre quando uma pista amplia o raciocínio e quando uma resposta completa pode reduzi-lo.

Figura 2 – Publicação de atividade com prazo, instruções e nível de apoio da IA



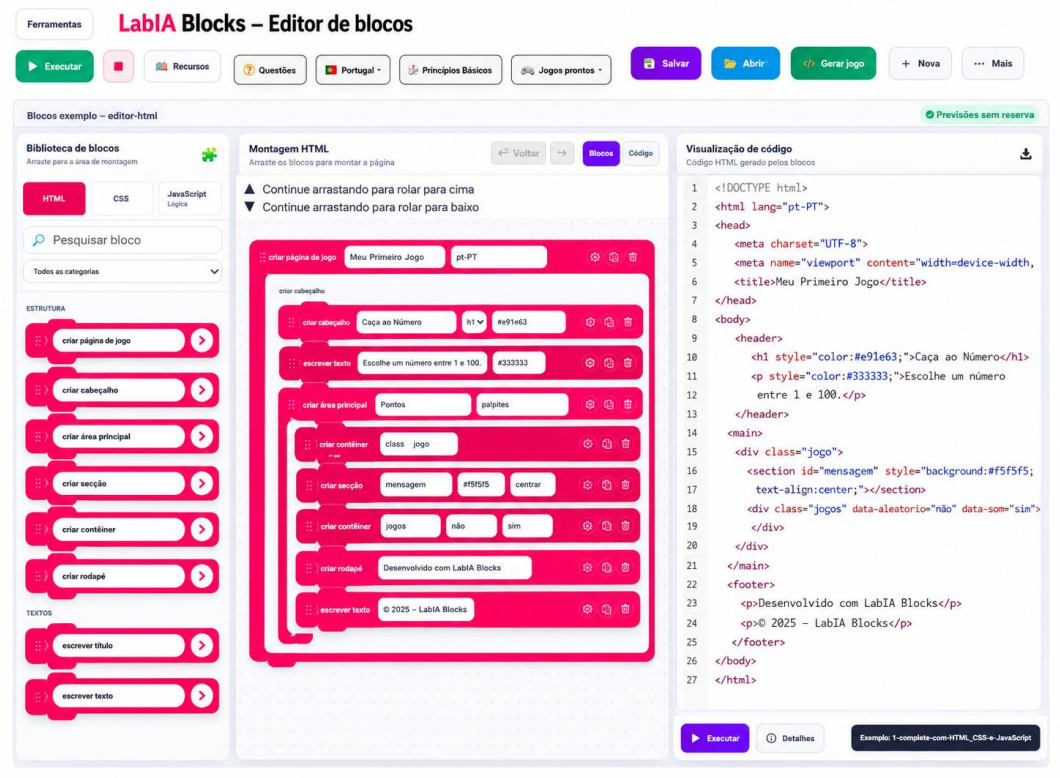
Fonte: elaboração própria a partir da versão analisada da LabIA Blocks (2026).

5 POSSIBILIDADES PEDAGÓGICAS DA IA NA LABIA BLOCKS

5.1 Explicação conceitual e transição entre blocos e código

Uma primeira possibilidade de uso da IA está na explicação de conceitos no momento em que eles se tornam necessários. Em vez de introduzir antecipadamente todas as propriedades de HTML, CSS e JavaScript, o professor pode organizar uma sequência em que o estudante encontra conceitos ao construir o jogo. Ao criar um placar, surge a necessidade de compreender elementos e identificadores; ao estilizar o cenário, propriedades de CSS passam a ter função concreta; ao programar movimento ou colisão, eventos, variáveis e condições ganham significado. O Tutor IA pode responder perguntas específicas sobre esses conceitos e relacioná-los ao bloco que está sendo usado. A Figura 3 evidencia a visualização simultânea da montagem em blocos e do código correspondente, aspecto utilizado como evidência para discutir a transição entre representações e a explicitação da sintaxe.

Figura 3 – Programação em blocos e visualização do código correspondente na LabIA Blocks



Fonte: elaboração própria a partir da versão analisada da LabIA Blocks (2026).

5.2 Pistas, depuração e feedback durante a resolução de problemas

A depuração é um dos momentos em que a IA pode assumir papel particularmente produtivo. Erros de programação não são apenas falhas a eliminar; podem ser situações de aprendizagem nas quais o estudante formula hipóteses sobre causa e efeito. Na LabIA Blocks, o aluno pode executar o projeto, observar o comportamento, retornar aos blocos e solicitar uma pista. Uma orientação útil não precisa revelar a solução inteira. Pode chamar atenção para uma variável não inicializada, uma condição colocada fora do evento adequado, um nome de personagem inconsistente ou um valor que não corresponde ao objetivo do jogo. O ciclo executar-observar-perguntar-revisar aproxima a IA de uma função de feedback formativo. A resposta surge enquanto ainda existe possibilidade de corrigir a estratégia, e não somente após a entrega. Para que esse potencial se concretize, é importante que o professor solicite evidências do processo: qual era o erro, qual hipótese foi testada, que sugestão foi recebida, o que precisou ser alterado e por que a nova versão funcionou. Esse registro pode ser breve, oral ou escrito, mas ajuda a transformar a consulta em reflexão sobre o próprio raciocínio.

5.3 Planejamento de jogos e tradução de regras em estruturas programáveis

Outra possibilidade está na passagem entre a ideia do jogo e a lógica implementável. Muitos estudantes conseguem narrar uma mecânica — “pegar três estrelas e depois abrir a porta” — mas encontram dificuldade para decompor essa regra em estados, eventos e condições. O Tutor IA pode ajudar a transformar a narrativa em uma sequência de blocos, indicando quais elementos precisam existir, que variáveis armazenam o estado, qual evento verifica a colisão e em que condição a vitória deve ocorrer. A utilidade pedagógica reside menos em receber uma lista pronta e mais em tornar explícita a estrutura lógica escondida na narrativa.

5.4 Diferenciação pedagógica, autonomia e acompanhamento formativo

Em turmas heterogêneas, a combinação entre atividades estruturadas e níveis de IA abre possibilidade de diferenciação pedagógica. Estudantes que ainda precisam de apoio podem trabalhar com exemplos prontos, desafios curtos e pistas; estudantes mais autônomos

podem receber a mesma missão com menor assistência e maior liberdade de implementação. O objetivo não é criar rótulos fixos, mas ajustar temporariamente o suporte à dificuldade da tarefa e ao estágio de aprendizagem. A possibilidade de mudar o nível de IA entre atividades permite ao professor reduzir o apoio à medida que o estudante demonstra maior domínio.

5.5 IA, gamificação e metodologias ativas

Essa combinação também favorece metodologias ativas baseadas em projetos. Em um projeto de jogo, o aluno precisa interpretar o problema, criar um artefato, testar, depurar e apresentar o resultado. A IA pode participar em diferentes momentos, mas a aprendizagem continua centrada na ação do estudante. A literatura sobre aprendizagem ativa destaca a importância de tarefas que exigem participação cognitiva e resolução de problemas (FREEMAN et al., 2014). Quando a IA é utilizada como apoio dentro desse ciclo, seu valor não está em acelerar a entrega, e sim em ampliar as oportunidades de feedback durante a construção. Nesse sentido, desafios e elementos de gamificação funcionam como estrutura de participação e progressão, e não como substitutos do objetivo conceitual da atividade (DETERDING et al., 2011).

5.6 Síntese crítica: potencialidades, fragilidades e contribuição

A leitura conjunta das dimensões analisadas indica três potencialidades articuladas: apoio contextual durante a construção do projeto, gradação da assistência sob decisão do professor e aproximação entre programação visual e código textual. Em ambientes clássicos de programação em blocos, como o Scratch, a criação visual e a autoria constituem elementos centrais (RESNICK et al., 2009); na LabIA Blocks, esse percurso é acrescido de um Tutor IA e da visualização do código web correspondente. Em relação a tutores baseados em grandes modelos de linguagem de uso genérico, a orientação é restringida pelo repertório efetivamente disponível no editor. Em comparação com o AIBlockAgent, há convergência no uso de apoio contextual e gradual, enquanto a LabIA Blocks enfatiza a definição prévia do nível de assistência pelo professor, a integração entre HTML, CSS e JavaScript e a gestão de atividades. Essas diferenças são características de desenho e não constituem evidência de

maior eficácia de uma tecnologia sobre outra (CHOA; KOO, 2026; GARCÍA-MÉNDEZ; DE ARRIBA-PÉREZ; SOMOZA-LÓPEZ, 2025).

As mesmas características também apresentam fragilidades. O vínculo do Tutor ao catálogo reduz a possibilidade de indicar blocos inexistentes, mas não elimina respostas inadequadas ou interpretações equivocadas, risco reconhecido na literatura sobre IA generativa em educação em computação (HUMBLE, 2024). A gradação do apoio pode favorecer autonomia somente se houver planejamento para reduzir a assistência quando apropriado; caso contrário, o nível mais alto pode favorecer dependência. Além disso, a condição de protótipo e a ausência de aplicação empírica com estudantes limitam a generalização dos achados. Assim, a contribuição do estudo é analítica e propositiva: explicita condições de uso e relações pedagógicas que precisam ser testadas em pesquisas posteriores.

O Quadro 1 sintetiza os resultados dessa análise, relacionando cada possibilidade de uso da IA à ação esperada, à intenção pedagógica e ao cuidado docente correspondente. A coluna de cuidados explicita que as potencialidades identificadas dependem de verificação, mediação e preservação da responsabilidade do estudante sobre a solução.

Quadro 1 – Possibilidades pedagógicas de uso da IA na LabIA Blocks

Possibilidade	Ação possível da IA	Intenção pedagógica	Cuidado docente
Explicação conceitual	Explicar bloco, evento, variável, condição ou propriedade.	Relacionar conceito à necessidade do projeto.	Evitar substituir a exploração por respostas prontas.
Pistas graduais	Sugerir onde observar ou qual relação revisar.	Manter o estudante responsável pela solução.	Graduar a ajuda conforme a dificuldade.
Depuração	Apontar inconsistências e formular hipóteses de erro.	Transformar falhas em situações de investigação.	Exigir execução e verificação da sugestão.
Planejamento lógico	Traduzir regras do jogo em entidades, eventos, estados e condições.	Apoiar decomposição do problema.	Comparar orientação com planejamento do aluno.
Diferenciação	Oferecer apoio distinto conforme o nível de assistência.	Adequar suporte a percursos heterogêneos.	Evitar dependência permanente do nível máximo.
Feedback formativo	Questionar, explicar e revisar decisões do projeto.	Favorecer metacognição e justificativa.	Valorizar o processo, não apenas o produto final.

Fonte: elaboração própria (2026).

6 PROPOSTA DE UTILIZAÇÃO DIDÁTICA DA IA COM A LABIA BLOCKS

A proposta apresentada nesta seção foi construída a partir dos resultados das etapas anteriores, e não como um roteiro independente da análise. A integração contextual do Tutor IA, observada na Seção 4.1, fundamentou seu uso durante momentos de construção e depuração; os níveis de apoio examinados na Seção 4.2 fundamentaram a gradação e a retirada do suporte; a relação entre blocos e código discutida na Seção 5.1 orientou atividades de explicação e transição entre representações; e os achados sobre depuração, planejamento e diferenciação das Seções 5.2 a 5.4 justificaram ciclos de tentativa, pista, teste e revisão. Por sua vez, as fragilidades discutidas na Seção 5.6 e na Seção 7 levaram à exigência de planejamento prévio, validação por execução e justificativa das decisões pelo estudante. Com base nesses resultados, os critérios usados para definir a proposta foram: progressão da autonomia, pertinência do apoio ao momento da tarefa, verificabilidade da resposta, preservação da autoria e produção de evidências de aprendizagem.

6.1 Sequência didática com assistência graduada

Uma forma de utilizar a LabIA Blocks em sala de aula é organizar o trabalho em etapas nas quais a IA assume funções diferentes. O ponto de partida é uma missão clara, com mecânica, critérios de funcionamento e produto esperado. O professor cria a turma, cadastra os estudantes, publica a atividade e define o nível inicial de assistência. Antes de abrir o Tutor, o aluno é convidado a descrever o problema com suas próprias palavras e planejar uma solução. Essa exigência inicial é importante porque registra o raciocínio anterior à consulta e oferece referência para comparar mudanças posteriores.

Após a execução, recomenda-se uma fase de validação. O estudante deve demonstrar que cada regra solicitada está funcionando e explicar os blocos responsáveis por ela. Se a IA tiver sugerido uma sequência, o aluno pode indicar quais partes utilizou, quais modificou e quais descartou. A apresentação final pode incluir uma pequena mudança solicitada pelo professor — por exemplo, alterar a quantidade de itens, trocar a condição de vitória ou adicionar uma vida — para verificar se o estudante compreende a lógica além da configuração específica do projeto. O Quadro 2 operacionaliza esses critérios ao associar, em cada etapa,

objetivo, ação discente, nível de IA sugerido e evidência de aprendizagem, tornando explícita a relação entre os achados da análise e a sequência didática proposta.

Quadro 2 – Sequência didática sugerida com níveis de assistência da IA

Etapa	Objetivo	Ação do estudante	IA sugerida	Evidência de aprendizagem
1. Problematizar	Compreender a missão e decompor regras.	Descrever entidades, ações, estados e objetivo.	0 – Sem IA	Planejamento inicial do próprio aluno.
2. Estruturar	Organizar a interface em HTML.	Montar cenário, textos, placar e componentes.	1 – Perguntas	Relação entre blocos e estrutura.
3. Estilizar	Relacionar CSS à experiência visual.	Configurar cores, tamanhos, disposição e responsividade.	1 ou 2	Justificativa das escolhas visuais.
4. Programar	Implementar eventos, variáveis e condições.	Criar movimento, colisões e regras do jogo.	2 – Pistas	Mecânicas funcionando conforme requisitos.
5. Depurar	Analisar erros e revisar hipóteses.	Executar, perguntar, testar e corrigir.	2 ou 3	Explicação do erro e da correção.
6. Apresentar	Consolidar autoria e autonomia.	Demonstrar o jogo e modificar uma regra.	0 ou 1	Capacidade de explicar e transferir a lógica.

Fonte: elaboração própria (2026).

6.2 Exemplos de atividades orientadas pela IA

Em uma atividade introdutória, o professor pode pedir que o estudante crie um personagem controlado pelas setas e um objetivo simples. O nível de IA pode ser limitado a perguntas conceituais. O foco da avaliação recai sobre a compreensão de evento e alteração de coordenadas. Em um segundo momento, uma missão de coleta pode exigir dois ou três itens independentes, uma variável de pontuação e condição de vitória. Nesse caso, o Tutor pode fornecer pistas se a contagem duplicar ou se o contador não atualizar. O estudante precisa demonstrar como a colisão, a variável e a remoção do item se relacionam. Em atividades intermediárias, podem ser introduzidas vidas, inimigos e condições compostas. Uma missão de fuga pode exigir que o jogador perca uma vida ao tocar no inimigo, retorne à posição inicial e seja derrotado quando as vidas chegarem a zero. O professor pode liberar o

modo Tutor apenas durante a depuração e solicitar que o aluno registre as perguntas feitas. A atividade avalia não somente se o jogo funciona, mas se o estudante consegue justificar a ordem das ações e explicar por que determinadas condições precisam ser verificadas continuamente.

7 LIMITAÇÕES, ÉTICA E CONDIÇÕES PARA USO EDUCACIONAL

A integração da IA à LabIA Blocks não deve ser interpretada como garantia de aprendizagem nem como substituição do professor. Modelos generativos podem produzir respostas inadequadas, interpretar ambiguidades de maneira diferente da intenção do estudante ou sugerir uma solução tecnicamente possível, porém incompatível com o objetivo pedagógico. Por isso, o sistema precisa ser utilizado com uma cultura de verificação. Em programação, essa verificação pode ocorrer pela execução do projeto, pela leitura do código gerado e pela explicação do estudante. O professor continua responsável por decidir quando a resposta atende ao objetivo da aula.

Privacidade e segurança também precisam integrar o planejamento institucional. Na arquitetura analisada, a chave de API do professor permanece no servidor e não é exposta no navegador do aluno, o que reduz um risco técnico específico. Ainda assim, escolas e instituições devem definir políticas para uso de serviços externos, tratamento de dados e responsabilidades de acesso. A UNESCO (2024) recomenda atenção à proteção de dados, adequação etária, transparência e supervisão humana na adoção de IA generativa.

Por fim, há uma limitação metodológica do próprio estudo: as possibilidades discutidas derivam da análise funcional da ferramenta e da literatura, não de resultados medidos com estudantes. Não se pode afirmar, neste momento, que a LabIA Blocks melhora desempenho, aumenta engajamento ou reduz dificuldades de programação. Essas hipóteses precisam ser investigadas em aplicações reais, com delineamentos capazes de observar aprendizagem, qualidade dos projetos, padrões de depuração, autonomia, percepção dos participantes e efeitos dos diferentes níveis de assistência da IA.

8 CONSIDERAÇÕES FINAIS

A análise realizada mostra que a principal contribuição potencial da inteligência artificial na LabIA Blocks não está em produzir jogos automaticamente, mas em ampliar as formas de mediação disponíveis durante a aprendizagem. O Tutor IA pode explicar conceitos, oferecer pistas, apoiar depuração, traduzir regras narrativas em estruturas programáveis e auxiliar a comparação entre blocos e código textual. Esses usos tornam-se pedagogicamente mais consistentes quando o estudante permanece responsável por planejar, executar, verificar e justificar a solução. Frente aos trabalhos relacionados, a contribuição identificada é de desenho pedagógico: combinar tutor contextual, transição entre blocos e código textual, assistência configurável pelo professor e gestão de atividades em um mesmo ambiente. O estudo não sustenta superioridade da plataforma nem efeitos de aprendizagem, que permanecem dependentes de validação empírica. A possibilidade de o professor controlar o nível de assistência é especialmente relevante porque permite integrar a IA a diferentes objetivos. A mesma plataforma pode ser utilizada sem IA em atividades diagnósticas, com perguntas e pistas em tarefas introdutórias ou com apoio mais amplo em projetos autorais complexos. Essa gradação favorece diferenciação pedagógica e cria condições para retirar progressivamente o suporte à medida que o estudante desenvolve autonomia. A gestão de turmas, atividades e entregas complementa esse desenho ao permitir que o uso da IA esteja vinculado a propostas concretas e a critérios de acompanhamento.

Como continuidade, recomenda-se investigar empiricamente diferentes estratégias de uso do Tutor IA em turmas de educação básica, cursos introdutórios e formação docente. Estudos futuros podem comparar níveis de assistência, analisar tipos de pergunta formulados pelos estudantes, observar mudanças na autonomia ao longo do tempo e avaliar se o uso de pistas produz efeitos diferentes do fornecimento de soluções mais completas. Também é relevante estudar como professores incorporam a IA a rubricas, feedbacks e atividades de depuração. Essas pesquisas permitirão transformar as possibilidades mapeadas neste artigo em evidências sobre condições efetivas de aprendizagem com a LabIA Blocks.

8.1 Agenda de pesquisa e formação docente

A continuidade desta agenda envolve não apenas testar o desempenho dos estudantes, mas observar como professores planejam intervenções com diferentes níveis de IA. Estudos de aplicação podem comparar atividades equivalentes realizadas sem assistência, com perguntas, com pistas e com apoio de Tutor, analisando quais configurações favorecem maior autonomia, melhor qualidade de explicação e menor dependência de respostas prontas. Também interessa investigar que tipos de pergunta surgem em cada etapa do projeto e de que maneira o feedback automatizado influencia a depuração e a revisão do código. A formação docente constitui outro eixo necessário. Para utilizar a IA de modo educacionalmente responsável, o professor precisa dominar as possibilidades da ferramenta, reconhecer limites dos modelos generativos e transformar respostas automáticas em oportunidades de questionamento. Oficinas de formação podem explorar elaboração de missões, definição de níveis de ajuda, análise crítica de respostas do Tutor e construção de rubricas que valorizem autoria, teste e justificativa. Essa preparação tende a ser decisiva para que a inteligência artificial funcione como mediação pedagógica e não como mecanismo de terceirização do raciocínio.

REFERÊNCIAS

CHOA, Haeun; KOO, Dukhoi. AIBlockAgent: design and development of an AI agent to support block-based programming learning. The Journal of Korean Association of Computer Education, v. 29, n. 3, p. 113-122, 2026. DOI: https://doi.org/10.32431/kace.2026.29.3.011. Acesso em: 20 set. 2026.

DETERDING, Sebastian; DIXON, Dan; KHALED, Rilla; NACKE, Lennart. From game design elements to gamefulness: defining gamification. In: PROCEEDINGS OF THE 15TH INTERNATIONAL ACADEMIC MINDTREK CONFERENCE. Tampere: ACM, 2011. p. 9-15. DOI: https://doi.org/10.1145/2181037.2181040. Acesso em: 13 ago. 2026.

FREEMAN, Scott et al. Active learning increases student performance in science, engineering, and mathematics. Proceedings of the National Academy of Sciences, v. 111, n. 23, p. 8410-8415, 2014. DOI: https://doi.org/10.1073/pnas.1319030111. Acesso em: 13 ago. 2026.

GARCÍA-MÉNDEZ, Silvia; DE ARRIBA-PÉREZ, Francisco; SOMOZA-LÓPEZ, María del Carmen. A review on the use of large language models as virtual tutors. Science & Education, v. 34, p. 877-892, 2025. DOI: https://doi.org/10.1007/s11191-024-00530-2. Acesso em: 20 set. 2026.

GROVER, Shuchi; PEA, Roy. Computational thinking in K-12: a review of the state of the field. Educational Researcher, v. 42, n. 1, p. 38-43, 2013. DOI: https://doi.org/10.3102/0013189X12463051. Acesso em: 13 ago. 2026.

HUMBLE, Niklas. Risk management strategy for generative AI in computing education: how to handle the strengths, weaknesses, opportunities, and threats? International Journal of Educational

Technology in Higher Education, v. 21, art. 61, 2024. DOI: https://doi.org/10.1186/s41239-024-00494-x. Acesso em: 13 ago. 2026.

RESNICK, Mitchel et al. Scratch: programming for all. Communications of the ACM, v. 52, n. 11, p. 60-67, 2009. DOI: https://doi.org/10.1145/1592761.1592779. Acesso em: 13 ago. 2026.

SAIDELLES, Tiago; BRASIL, Miguel Augusto Bauermann. LabIA Blocks: programação em blocos, inteligência artificial e gamificação no ensino introdutório de jogos web. Aurum Editora (e-Books), p. 492-518, 2026. DOI: https://doi.org/10.63330/aurumpub.057-026. Acesso em: 20 set. 2026.

UNESCO. Guia para a IA generativa na educação e na pesquisa. Paris: UNESCO, 2024. ISBN 978-92-3-700028-1. Disponível em: https://unesdoc.unesco.org/ark:/48223/pf0000390241. Acesso em: 13 ago. 2026.

WEINTROP, David; WILENSKY, Uri. Comparing block-based and text-based programming in high school computer science classrooms. ACM Transactions on Computing Education, v. 18, n. 1, art. 3, p. 1-25, 2018. DOI: https://doi.org/10.1145/3089799. Acesso em: 13 ago. 2026.



This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International (CC BY-NC 4.0).