# Desenvolvimento de Quizbots Educacionais com Low-Code

**Vinícius Silva, José Dantas Neto, Samuel Carneiro, Murilo Maciel, Anderson Gomes, Elvys Soares**

[1]Instituto Federal de Alagoas (IFAL) – Maceió/AL – Brasil

{vrs20, jdsn1, slsc2, mfpm1}@aluno.ifal.edu.br,
{anderson,elvys.soares}@ifal.edu.br

***Abstract.*** *Artificial Intelligence (AI) is becoming increasingly present in academic spaces, assisting teachers and supporting learning processes. However, general AI tools do not always work perfectly in the school environment, as their use carries the risks of spreading misinformation and superficial answers, which can confuse students who do not yet master the content and cannot identify such inconsistencies. This work proposes the development of educational quizbots using low-code strategies to assist students in reviewing academic content through interactive questionnaires. The development of the solution involved the creation of quizbot prototypes on the Appsmith and Telegram platforms, the generation of a bank of 8,000 multiple-choice questions with the Google Gemini API, the validation of a sample of questions by teachers, and the performance of usability tests with students. Validation with teachers indicated an estimated validity rate of 96.9% for the questions produced, while usage tests with students showed that 95% found the system easy to use, and 100% stated that the tool helps them memorize the content.*

***Resumo.*** *A Inteligência Artificial (IA) vem se tornando cada vez mais presente nos espaços acadêmicos, auxiliando professores e apoiando os processos de aprendizagem. No entanto, ferramentas de IA gerais nem sempre funcionam perfeitamente no ambiente escolar, pois sua utilização traz riscos de disseminação de desinformação e respostas superficiais, o que pode confundir o estudante que ainda não domina o conteúdo e não consegue identificar tais inconsistências. Este trabalho propõe o desenvolvimento de quizbots educacionais utilizando estratégias low-code para auxiliar os estudantes na revisão de conteúdos acadêmicos por meio de questionários interativos. O desenvolvimento da solução envolveu a criação de protótipos de quizbots nas plataformas Appsmith e Telegram, a geração de um banco de 8.000 questões de múltipla escolha com a API do Google Gemini, a validação de uma amostra das perguntas por docentes e a realização de testes de usabilidade com estudantes. A validação com os professores apontou uma taxa estimada de validade de 96,9% para as questões produzidas, enquanto os testes de uso com alunos mostraram que 95% acharam o sistema fácil de utilizar e 100% afirmaram que a ferramenta ajuda a memorizar os conteúdos.*

## 1. Introdução

A Inteligência Artificial (IA) vem se tornando cada vez mais presente nos espaços acadêmicos, sendo utilizada em atividades de ensino, aprendizagem e produção de tra-

balhos acadêmicos. Nesse contexto, ferramentas de IA podem auxiliar professores, especialmente em situações de alta demanda de estudantes, e apoiar os processos de aprendizagem dos usuários que as utilizam [Holmes et al. 2023].

No cenário promissor dessa arquitetura, encontram-se os *quizbots*, que são um método de ensino focado em guiar o usuário por meio de questionários interativos. No âmbito educacional, essa abordagem se saiu superior às formas tradicionalmente usadas, como os flashcards (cartões de memória digitais), pois incentiva a recuperação ativa do conhecimento e estimula o tempo que os estudantes passam se dedicando, por conta própria, aos estudos [Ruan et al. 2019].

Para realizar a criação dessas ferramentas de forma ágil, as plataformas de desenvolvimento *low-code* surgem como uma alternativa viável. Essa abordagem baseia-se na utilização de interfaces visuais pré-programadas e componentes do tipo "arrastar e soltar", o que reduz a necessidade de conhecimento em programação e incentiva a criação de softwares, permitindo que profissionais de outras áreas, como educadores, participem ativamente do desenvolvimento dessas soluções [Juhás et al. 2022].

Quanto à utilização de sistemas de IA de propósito geral, estes podem trazer sérios riscos de disseminação de desinformação, preconceitos e respostas superficiais, que podem confundir o estudante caso não haja um direcionamento claro. No contexto educacional, se um aluno utiliza a ferramenta justamente para aprender um conteúdo que ainda não domina, ele pode não possuir o conhecimento prévio necessário para identificar tais inconsistências [Kasneci et al. 2023].

Assim, este trabalho propõe o desenvolvimento de quizbots educacionais utilizando estratégias *low-code* para auxiliar estudantes na revisão de conteúdos acadêmicos por meio de sessões interativas de perguntas e respostas. O desenvolvimento da solução envolveu (i) A avaliação de plataformas low-code para criação da aplicação; (ii) O projeto e a implementação dos protótipos dos quizbots; (iii) A geração de um banco de questões utilizando a API Google Gemini; (iv) A validação das questões por docentes; e (v) A realização de testes de usabilidade com estudantes.

Como resultados deste trabalho, destaca-se a criação de um banco com aproximadamente 8.000 questões de múltipla escolha, abrangendo quatro disciplinas e seus respectivos conteúdos. Durante a validação pedagógica, docentes analisaram uma amostra de 192 questões — representando 95% de confiança e 5% de margem de erro — e identificaram apenas três inconsistências, o que sugere um elevado nível de precisão do material produzido. Os estudantes que participaram dos testes também apresentaram avaliações bastante positivas. Numa escala de 0 a 5 pontos, a facilidade de uso recebeu nota média de 4,15, com 95% de aprovação. Resultados semelhantes foram observados na avaliação da metodologia baseada em *quizzes*, que alcançou uma média de 4,35. Já o impacto percebido na retenção dos conteúdos obteve média de 4,45, sendo reconhecido por todos os participantes como um fator que contribuiu para a aprendizagem.

Dessa forma, os resultados obtidos demonstram que a combinação entre inteligência artificial e tecnologias *low-code* constitui uma abordagem viável para o desenvolvimento de ferramentas educacionais capazes de apoiar e potencializar o processo de aprendizagem. Portanto, as principais contribuições deste trabalho são:

- Uma proposta de desenvolvimento de quizbots com o apoio de Inteligência Arti-

ficial e tecnologias low-code (Seção 3);
- A criação e validação de um conjunto de 8000 perguntas relativas a conteúdos do ensino médio (Seção 3 e Seção 4);
- A demonstração de que é possível criar quizbots especializados em conteúdos do ensino médio brasileiro para auxílio ao processo de ensino-aprendizagem (Seção 4).

O restante deste trabalho está organizado da seguinte forma: a Seção 2 apresenta os trabalhos relacionados; a Seção 3 descreve as etapas de desenvolvimento de um quizbot com tecnologias low-code; a Seção 4 os softwares desenvolvidos, a validação deles por docentes e estudantes e discute o trabalho; e a Seção 5 apresenta as conclusões e trabalhos futuros.

## 2. Trabalhos Relacionados

O uso de inteligência artificial generativa, LLMs e chatbots na educação é um tema discutido recorrentemente no ambiente acadêmico. Em [Holmes et al. 2023], a IA é tratada como uma forma eficiente de apoiar o aprendizado. Neste livro, os autores focam em como essas tecnologias podem ajudar na produção de materiais didáticos e ampliar o acesso a diversos recursos educacionais úteis. Enquanto nossa pesquisa se aproxima disso ao utilizar a API do Gemini para gerar questões objetivas, ela se diferencia por ter como objetivo a revisão dos conteúdos do Ensino Médio e por avaliar os materiais produzidos com docentes e discentes.

Em [Ruan et al. 2019], os autores apresentam o QuizBot: um sistema baseado em conversação para auxiliar a aprendizagem por meio de quizzes interativos. Seus resultados se demonstraram positivos no engajamento e na memorização do conteúdo, semelhante ao método de estudo de flashcards. Diferentemente de [Ruan et al. 2019], nossa pesquisa foca no desenvolvimento de quizbots com tecnologias low-code que utilizam um banco de questões gerado por IA.

[Kasneci et al. 2023] discutem o potencial de chatbots na educação, ressaltando a capacidade de personalização do aprendizado e do suporte individual a cada estudante. Contudo, os autores destacam problemas com alucinações, isto é, a geração de informações incorretas pela IA. Essa limitação é trabalhada diferentemente em nossa pesquisa através da realização de uma etapa de validação das questões produzidas pela IA, buscando garantir a qualidade das questões utilizadas nos quizzes. Este trabalho também se diferencia pelo uso de quizbots ao invés de chatbots.

[Juhás et al. 2022] falam sobre plataformas low-code de um modo geral e do potencial dessa tecnologia para simplificar o desenvolvimento de softwares. Os autores destacam que essas ferramentas reduzem a programação tradicional por meio de interfaces visuais Drag & Drop e de módulos reutilizáveis, agilizando o desenvolvimento e deixando-o acessível. Porém, diferentemente deste trabalho, [Juhás et al. 2022] não mencionam o uso dessa tecnologia em conjunto com IAs ou na área da educação.

[Lima 2025] investigam métodos para avaliar a eficácia do EducaChat como ferramenta de apoio ao ensino. Os autores enfatizam a importância do feedback dos usuários e também destacam os impactos dos chatbots no processo de aprendizagem, como [Kasneci et al. 2023]. Mas, além disso, propõem estratégias para medir sua utilidade em contextos educacionais. [Lima 2025] se aproximam da nossa pesquisa por

também apresentar uma proposta que utiliza a IA como auxílio para o aprendizado, porém se diferenciam por se tratar de um chatbot, e não um quizbot.

## 3. Desenvolvendo um Quizbot com Low-Code

Nesta seção, apresentaremos a metodologia de desenvolvimento da solução proposta. Em linhas gerais, apresentamos a arquitetura da solução por Quizbots (Seção 3.1). Em seguida, mostramos duas implementações utilizando tecnologias *low-code* distintas, com Appsmith (Seção 3.2) e Telegram (Seção 3.3). Por fim, apresentamos a estrutura das questões criadas, cuja base é lida por ambas as implementações de *frontend* (Seção 3.4).

### 3.1. Arquitetura da Solução

A arquitetura de software da solução proposta é dividida entre um módulo de interface gráfica (*i.e. frontend*) e um repositório estruturado de dados, resultante de prompts feitos ao Google Gemini. O *frontend*, desenvolvido sob uma plataforma *low-code*, gerencia o fluxo de navegação do estudante de forma hierárquica: inicialmente, são apresentadas as disciplinas disponíveis e, após a seleção de uma matéria, o sistema então apresenta os conteúdos disponíveis. Ao optar por um conteúdo disponível, a aplicação executa uma requisição no banco de dados — estruturado em forma de planilha — selecionando aleatoriamente um lote de 10 questões de múltipla escolha geradas previamente pela API do Google Gemini.

A avaliação de duas plataformas *low-code* com paradigmas distintos — Appsmith (interface web gráfica) e Telegram (interface conversacional) — justifica-se pela necessidade de analisar como diferentes arquiteturas impactam a usabilidade e o engajamento dos estudantes. Enquanto o Appsmith oferece uma estrutura robusta e customizável, o Telegram reduz a fricção tecnológica ao utilizar um ecossistema de mensageria já familiar aos discentes. Essa abordagem comparativa permite mapear se a eficácia na aprendizagem é maximizada pela riqueza visual de um sistema dedicado ou pela conveniência de uma ferramenta de uso diário. De qualquer forma, tentou-se utilizar o mesmo fluxo de informações (*e.g.,* escolha do assunto, escolha do tópico e escolha da resposta para cada pergunta), respeitando as diferenças tecnológicas de *frontend*.

As aplicações desenvolvidas estão disponíveis em repositórios públicos. [1] [2]

### 3.2. Appsmith

O Appsmith é uma plataforma *open-source* e *low-code* diferenciada pelo alto nível de customização. No desenvolvimento do aplicativo, priorizou-se a utilização de componentes nativos da interface, enquanto as regras de negócio de maior complexidade foram implementadas via *scripts* em JavaScript gerados por Inteligência Artificial.

Com relação ao fluxo da aplicação, estabelecido na Seção 3.1, após a seleção do assunto e o sorteio das 10 questões, o estudante é direcionado para a tela inicial apresentada pela Figura 1a. Nela, são encontrados (i) o número da questão atual, (ii) o enunciado da questão, e (iii) as alternativas.

Ao responder a questão, o usuário é apresentado uma caixa de diálogo (Figura 1b) dependente da alternativa selecionada estar correta ou errada, caso esteja errada, essa

[1] https://github.com/viniciusrodrigues70/ifal-tutor-ai

[2] https://github.com/jose-dnt/quiz-appsmith

**Figura 1. Telas do quiz desenvolvido no Appsmith**




**(a) Tela do quiz na questão.** **(b) Tela do quiz ao errar a questão**

caixa mostra o comentário referente àquela alternativa e também apresenta a alternativa correta.

Ao fim do quiz, é mostrada a tela de resultado contendo a quantidade de acertos em relação a quantidade de perguntas, as opções de repetir o mesmo quiz ou de gerar um novo e, caso o usuário resolva encerrar o quiz, uma pesquisa de satisfação.

### 3.3. Telegram

O Telegram é um aplicativo de mensageria multiplataforma que se destaca no ecossistema de desenvolvimento de software por disponibilizar uma Interface de Programação de Aplicações (API) para bots que é aberta, robusta e gratuita. Essa característica viabilizou a construção de uma interface de usuário tanto acessível quanto familiar para os estudantes, mitigando barreiras de adoção tecnológica ao eliminar a necessidade de instalação de novos softwares nos dispositivos móveis ou computadores dos alunos da instituição.

O desenvolvimento do bot foi realizado na linguagem Python, utilizando a biblioteca *python-telegram-bot*. A arquitetura do código foi estruturada considerando a programação assíncrona, assegurando a concorrência do sistema e a estabilidade do serviço em caso de acessos simultâneos. A interface gráfica aproveita os componentes nativos da plataforma, empregando teclados em linha (*inline keyboards*) e requisições de retorno (*callback queries*) para a construção dos menus dinâmicos de seleção de disciplinas, tópicos e formatos de questionários. As regras de negócio de maior complexidade foram implementadas por meio de *scripts* customizados em Python, englobando: o gerenciamento de estado e progresso do estudante integrado ao banco de dados NoSQL MongoDB; a limpeza de *strings* para prevenir que caracteres especiais e notações matemáticas corrompessem a renderização visual das mensagens; o processamento de entradas via expressões regulares (*Regex*); e um algoritmo de ordenação aleatória para o embaralhamento das alternativas, garantindo o dinamismo na disposição da resposta correta a cada tentativa.

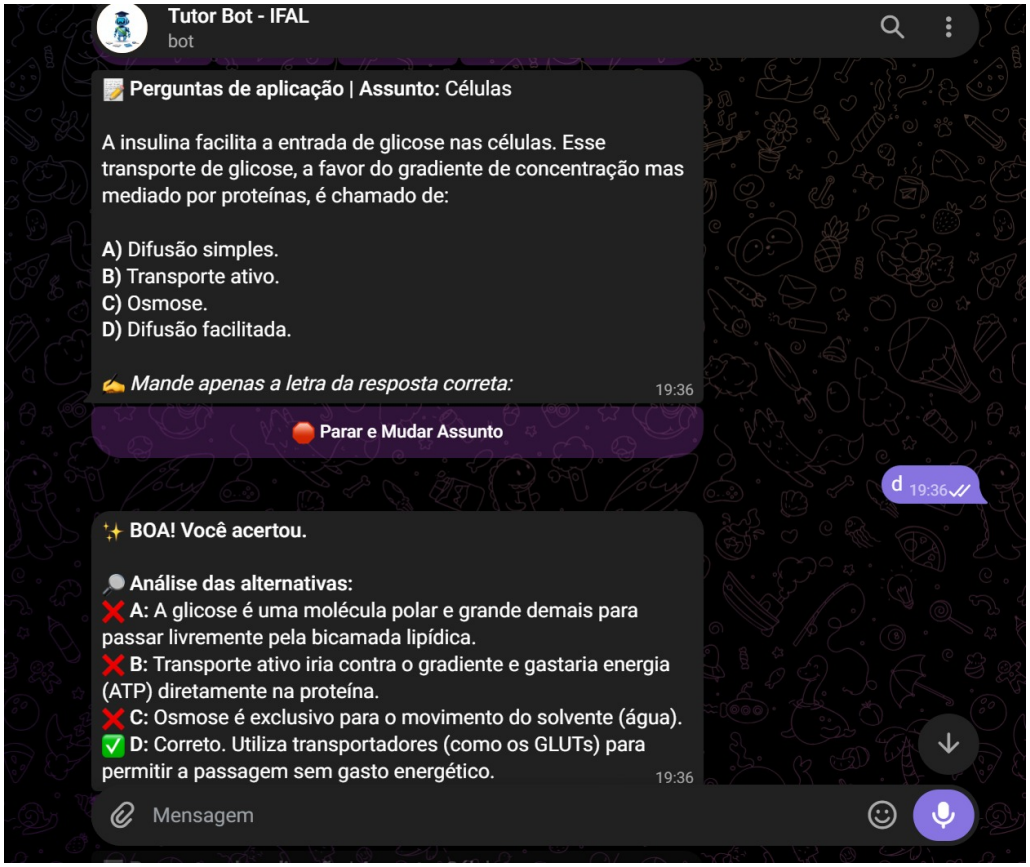



**Figura 2. Tela do Telegram**

### 3.4. Estrutura e Criação das Questões

O preenchimento do repositório de dados das aplicações baseia-se na geração automatizada de questões por meio da API do modelo de linguagem em larga escala (LLM) Google Gemini. Devido a restrições operacionais de cotas de requisição e limites de tokens por chamada do plano disponível, o processo de geração foi estruturado em lotes contendo 50 unidades por execução.

Para minimizar alucinações do modelo e assegurar a padronização, aplicou-se a engenharia de prompts através de um modelo de comando padronizado (prompt template) parametrizável e reutilizável. Cada ciclo de execução da IA é condicionado e delimitado por três variáveis de entrada: a disciplina de origem, o conteúdo programático específico e o tipo de pergunta.

Para abranger diferentes níveis, modelou-se uma taxonomia composta por 8 categorias distintas de perguntas: Conceituais Básicas, Intermediárias, Avançadas, Com Pegadinhas, Baseadas em Exemplos Práticos, Comparativas, De Aplicação e De Revisão Geral.

Cada item gerado é composto pelos seguintes atributos:

- **Enunciado:** O texto principal contendo a problematização;
- **Alternativas:** Quatro opções de escolha, indexadas sequencialmente de A a D;
- **Gabarito:** O identificador único correspondente à alternativa correta;

- **Feedback:** Comentários explicativos e justificativas teóricas associados a cada uma das alternativas.

A Tabela 1 ilustra o formato final de uma questão estruturada tal como é retornada pela API do Gemini antes de sua inserção no banco de dados.

**Tabela 1. Exemplo de questão criada pelo Google Gemini:**

| | |
|---|---|
| Matéria | História |
| Conteúdo | Idade Média |
| Tipo de pergunta | Perguntas conceituais básicas |
| Pergunta | Qual evento é tradicionalmente utilizado pelos historiadores para marcar o início da Idade Média em 476 d.C.? |
| Alternativa A | A queda de Constantinopla. |
| Alternativa B | A queda do Império Romano do Ocidente. |
| Alternativa C | A fundação do Império Carolíngio. |
| Alternativa D | O início das Cruzadas. |
| Alternativa correta | B |
| Comentário A | A queda de Constantinopla em 1453 marca o fim da Idade Média. |
| Comentário B | Correto. A deposição do último imperador romano de Roma marca a transição da Antiguidade para o Medievo. |
| Comentário C | O Império Carolíngio surgiu séculos depois, por volta do ano 800. |
| Comentário D | As Cruzadas começaram apenas no final do século XI. |

## 4. Resultados e Validação

Os resultados apresentados nesta seção referem-se às avaliações dos quizbots implementados nas aplicações Telegram e Appsmith por docentes e discentes, bem como às etapas anteriores de desenvolvimento.

### 4.1. Validação das perguntas

Com o auxílio da API do Gemini, foram geradas 8.000 questões objetivas para os quizzes das aplicações. Esse total foi distribuído igualmente entre quatro disciplinas (2.000 questões cada), sendo cada uma delas subdividida em cinco conteúdos, resultando em 400 questões por tópico.

A validação desse banco de dados contou com o apoio de sete docentes do campus. A avaliação baseou-se em uma amostra aleatória de 192 questões, o que garante um nível de confiança de 95% e uma margem de erro de 5% para o total de questões geradas. Os professores, abordados inicialmente de forma presencial, receberam por e-mail um arquivo PDF com as questões de suas respectivas áreas. Cada docente analisou individualmente 24 questões, avaliando a correção e a coerência dos enunciados e das alternativas.

Da amostra analisada, 6 questões apresentaram problemas de coerência e coesão que comprometeram a compreensão do enunciado. Essas falhas foram atribuídas a alucinações da inteligência artificial e à falta de especificidade nos prompts de geração. Entre as questões com inconsistências, cinco eram de matemática — devido à ausência de parênteses e de unidades de medida adequadas — e uma era de português, na qual a IA gerou mais de uma alternativa correta para um item sobre acentuação.

Com base nesses resultados, é estimada uma taxa de validade de 96,9% para o grupo de 8.000 questões geradas pela API do Gemini, o que é condizente com a Literatura [Santos 2025, Silva and Leite 2026]. A base de perguntas geradas e o arquivo contendo a validação pelos docentes se encontram disponíveis online.[3]

### 4.2. Satisfação dos Alunos

Após a validação das questões geradas, 20 alunos de diferentes séries do Ensino Médio no campus foram convidados a testar as aplicações, com o banco de questões, e as avaliarem em um formulário Google separado baseado na escala de Likert variando de "extremamente satisfeito(a)" a "nem um pouco satisfeito(a)" sobre a facilidade de uso da implementação testada, appsmith ou telegram, a metodologia de quizzes da implementação testada e a eficácia da implementação na assistência à memorização do conteúdo. Nesta etapa, 10 discentes avaliaram o quizbot Appsmith, e os outros 10 avaliaram o quizbot Telegram.

Destes discentes, 3 relataram as seguintes imprecisões nos quizzes: "uma questão de matemática em que o enunciado não condizia com a resposta", "A IA selecionou a alternativa errada ao invés da certa, mas assumiu estar errada com o comentário embaixo", e "Todas as questões corretas eram as maiores". As questões imprecisas mencionadas nas 2 primeiras respostas foram removidas do banco de questões.

Além disso, do total de alunos que testaram as aplicações, 19 responderam positivamente sobre a facilidade de uso das plataformas, 19 responderam positivamente acerca da metodologia usada de quizzes e todos responderam que o software testado ajudou a memorizar o conteúdo.

### 4.3. Discussão

Os resultados mostram que a estratégia de prompt utilizada para a geração de questões foi eficaz, porém, mostram também que é necessário mais especificações no prompt das questões de matemática. Além disso, a taxa de validação de 96,9% revela que a maior parte das questões produzidas pela API do Gemini apresentam qualidade suficiente para serem utilizadas em ambientes educacionais e materiais didáticos, exigindo apenas certas intervenções e especificações para corrigir as raras inconsistências.

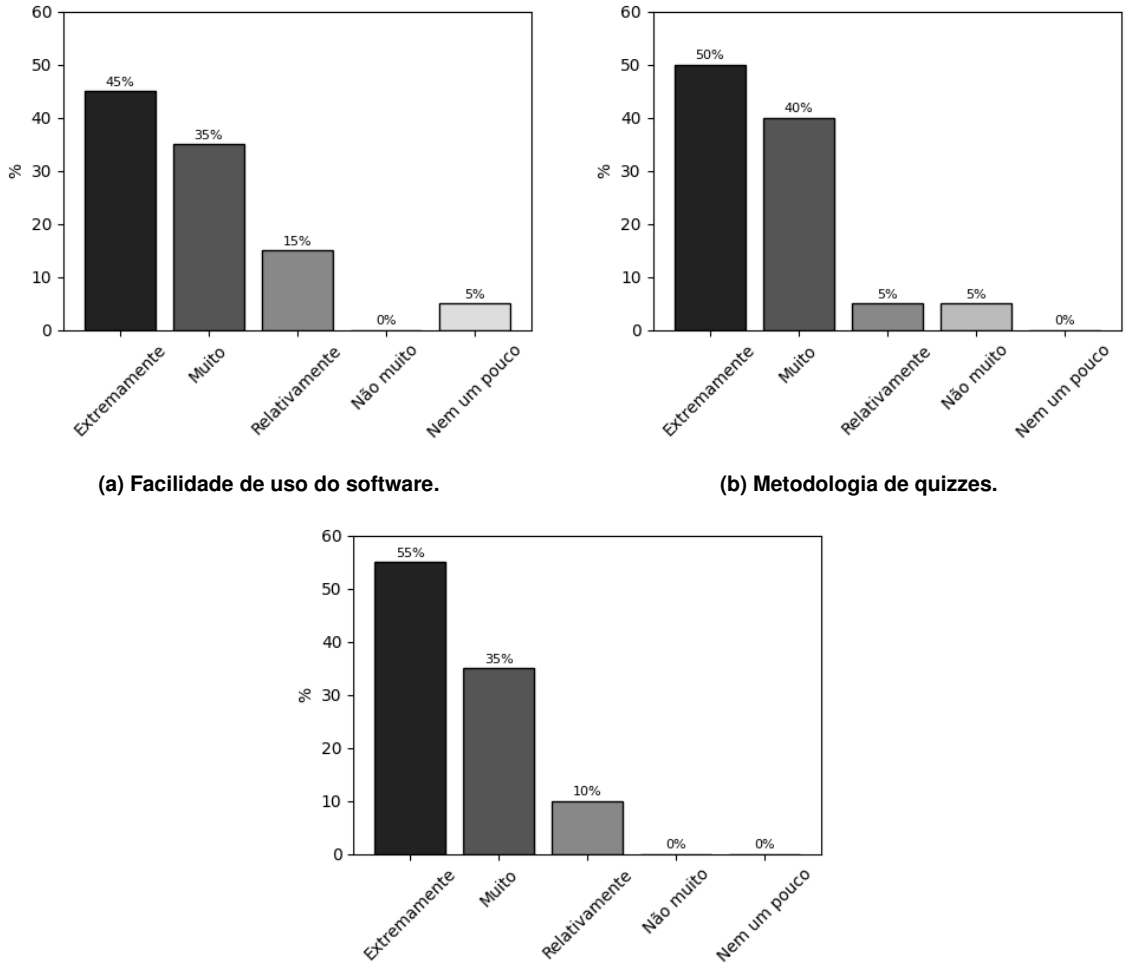
Semelhantemente, a avaliação dos estudantes revelou que os quizbots desenvolvidos são bem aceitos pelos usuários. Os índices de satisfação relacionados à facilidade de uso, à metodologia de quizzes e à ajuda na memorização indicam que o software foi capaz de alcançar os nossos objetivos de oferecer uma experiência acessível e proveitosa para a aprendizagem utilizando estratégias low-code.

Contudo, as imprecisões mencionadas pelos alunos e identificadas durante a validação dos professores mostram que ainda tem limitações no uso da inteligência artificial na educação. Problemas como a coerência entre enunciados e respostas, a escolha de alternativas incorretas e a repetição de padrões podem prejudicar o sistema.

Dessa forma, os resultados reforçam que o uso de inteligência artificial para a geração automatizada de questões educacionais é confiável e que é possível desenvolver um quizbot baseado em IA com estratégias low-code, ao mesmo tempo que mostram

[3]https://doi.org/10.6084/m9.figshare.33260844

**Figura 3. Avaliação dos alunos.**




**(a) Facilidade de uso do software.**

**(b) Metodologia de quizzes.**

**(c) Auxílio à memorização do conteúdo.**

como é necessário o aprimoramento contínuo dos prompts de geração e a implementação de formas de verificação ao longo da vida do sistema.

Vale pontuar que os testes de usabilidade ficaram restritos a 20 alunos de uma única escola. Essa escolha atende ao caráter piloto do estudo, mas impede que os resultados obtidos sejam aplicados diretamente a outros contextos educacionais sem novas validações.

## 5. Conclusões

Após a execução deste trabalho, é possível afirmar que a estruturação de um Quizbot com base em inteligência artificial generativa e por meio de tecnologias *low-code* pode ser realizada com sucesso, facilitando a criação de materiais complementares para docentes que desejam desenvolver suas próprias plataformas de questionários interativos de maneira acessível, mesmo sem possuir conhecimento aprofundado em programação.

A relevância de uma ferramenta como o Quizbot está em sua contribuição para o processo de aprendizagem de estudantes do ensino médio. Recursos educacionais que

incentivam o estudo autônomo desempenham um papel importante na formação discente, com maior engajamento e participação ativa no processo de construção do conhecimento.

O nível de qualidade dos questionários gerados pelo Quizbot mostrou-se próximo do ideal após o processo de validação humana. Entretanto, ainda existem oportunidades de aprimoramento. Entre elas, destaca-se a ampliação da quantidade de conteúdos contemplados pela ferramenta, abrangendo diferentes áreas do conhecimento. Além disso, a implementação de mecanismos para seleção do nível de dificuldade dos questionários poderia proporcionar maior adaptabilidade às necessidades e aos conhecimentos prévios dos estudantes.

Como trabalhos futuros, pretende-se uma ampliação na base de questões utilizadas pelo Quizbot, resultando em uma maior diversificação dos conteúdos. Também é previsto um aprimoramento no processo de validação das questões para essa versão ampliada do banco, diminuindo a necessidade de intervenção manual. Por fim, estudos com períodos de teste mais longos possibilitarão uma análise mais aprofundada do impacto da utilização do quizbot no desempenho dos estudantes.

Dessa forma, conclui-se que os resultados obtidos atenderam aos objetivos estabelecidos inicialmente neste trabalho, além de evidenciarem a relevância de ferramentas que complementem o aprendizado de alunos do ensino médio. Nesse contexto, o Quizbot apresenta-se como uma alternativa viável, capaz de tornar o processo de aprendizagem mais interativo, personalizado e alinhado às demandas educacionais contemporâneas.

## Referências


Holmes, W., Miao, F., et al. (2023). *Guidance for generative AI in education and research*. Unesco Publishing.

Juhás, G., Molnár, L., Juhásová, A., Ondrišová, M., Mladoniczky, M., and Kováčik, T. (2022). Low-code platforms and languages: the future of software development. In *2022 20th International Conference on Emerging eLearning Technologies and Applications (ICETA)*, pages 286–293. IEEE.

Kasneci, E., Seßler, K., Schwaiger, S., Diaz, M., Schmucker, U., et al. (2023). Chatgpt for good? on the perspective of conversational ai in education. *Learning and Individual Differences*, 103:102274.

Lima, S. O. (2025). Inteligência artificial e educação: estratégias para avaliar a eficácia do chatbot educacional educachat. Dissertação (mestrado em ensino em ciência e saúde), Universidade Federal do Tocantins, Palmas.

Ruan, S., Jiang, L., Xu, J., Tham, B. J.-K., Qiu, Z., Zhu, Y., Murnane, E. L., Brunskill, E., and Landay, J. A. (2019). Quizbot: A dialogue-based adaptive learning system for factual knowledge. In *Proceedings of the 2019 CHI conference on human factors in computing systems*, pages 1–13.

Santos, T. (2025). Abordagens pedagógicas com o google gemini no ensino de matemática. *Revista Baiana de Educação Matemática*, 6(1):e202505–e202505.

Silva, J. B. C. d. and Leite, B. S. (2026). Análise do desempenho de chatbots na resolução de questões de química em provas do enem. *Educitec - Revista de Estudos e Pesquisas sobre Ensino Tecnológico*, 12(jan./dez.):e279626.