UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO

Ancoragem Curricular via RAG: um Middleware para Mitigação de Alucinações na Geração de Questões Alinhadas à BNCC

Relatório Técnico relativo ao Trabalho de Conclusão Curso do Bacharelado em Sistemas de Informação na modalidade Empresa

---

Aluno
Eduardo Estevão Nunes Cavalcante

Orientador
Cícero Garrozi
Departamento de Estatística e Informática - UFRPE

6 de julho de 2026

Eduardo Estevão Nunes Cavalcante

Ancoragem Curricular via RAG: um Middleware para Mitigação de Alucinações na Geração de Questões Alinhadas à BNCC

Relatório Técnico apresentado ao Curso de Bacharelado em Sistemas de Informação da Universidade Federal Rural de Pernambuco, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Universidade Federal Rural de Pernambuco – UFRPE

Departamento de Estatística e Informática

Curso de Bacharelado em Sistemas de Informação

Orientador: Cícero Garrozi

Recife

Julho de 2026

Dados Internacionais de Catalogação na Publicação
Sistema Integrado de Bibliotecas da UFRPE
Bibliotecário(a): Suely Manzi – CRB-4 809

C377a Cavalcante, Eduardo Estevão Nunes.
Ancoragem curricular via RAG: um middleware para mitigação de alucinações na geração de questões alinhadas à BNCC / Eduardo Estevão Nunes Cavalcante. – Recife, 2026.
31 f.; il.

Orientador(a): Cícero Garrozi.

Trabalho de Conclusão de Curso (Graduação) – Universidade Federal Rural de Pernambuco, Bacharelado em Sistemas da Informação, Recife, BR-PE, 2026.

Inclui referências.

1. Sistemas de informação - Estudo e ensino. 2. Inteligência artificial. 3. Middleware . 4. Base Nacional Comum Curricular 5. Modalidade (Linguagem). I. Garrozi, Cícero, orient. II. Título

CDD 004

Eduardo Estevão Nunes Cavalcante

Ancoragem Curricular via RAG: um Middleware para Mitigação de Alucinações na Geração de Questões Alinhadas à BNCC

Artigo apresentado ao Curso de Bacharelado em Sistemas de Informação da Universidade Federal Rural de Pernambuco, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Aprovada em: 06 de Julho de 2026.

BANCA EXAMINADORA

Cícero Garrozi (Orientador)
Departamento de Estatística e Informática
Universidade Federal Rural de Pernambuco

André Kunio de Oliveira Tiva
Departamento de Tecnologia da Informação
Centro Universitário Maurício de Nassau

Resumo

A elaboração de avaliações alinhadas à Base Nacional Comum Curricular (BNCC) é um processo complexo e custoso para educadores e empresas de tecnologia educacional (*EdTechs*). Embora os Grandes Modelos de Linguagem (LLMs) apresentem potencial para a Geração Automática de Questões (AQG), sua aplicação direta no domínio normativo brasileiro esbarra em dois obstáculos, a alucinação de dados estruturados, em que o modelo inventa ou associa de forma incorreta códigos curriculares, e a transferência da complexidade da Engenharia de *Prompt* para o usuário final. Este trabalho apresenta a concepção e a implementação de um *middleware* de ancoragem curricular baseado no paradigma *Retrieval-Augmented Generation* (RAG) para o ecossistema educacional Gradepen, da Machlev Smart Applications. A solução atua como uma camada de abstração no *backend*, que infere a intenção do usuário, realiza uma recuperação determinística sobre a hierarquia curricular oficial modelada em JSON e orquestra a construção de um *prompt* enriquecido em XML. Esse *prompt* incorpora exemplos *Few-Shot*, o nível cognitivo da Taxonomia de Bloom Revisada e regras psicométricas para a formulação de distratores, antes de consultar uma API de LLM de propósito geral. Diferentemente da formulação mais difundida de RAG, a recuperação é determinística, baseada em correspondência exata sobre o vocabulário fechado e padronizado da BNCC. Implantada em ambiente de homologação, a solução opera de forma nativa no *backend*, o que dispensa serviços externos e reduz a complexidade e o custo computacional da operação. A abordagem busca assegurar que os códigos e os conteúdos curriculares das questões geradas existam, de fato, na base oficial, reduzindo a incidência de alucinações normativas e tornando os recursos de IA acessíveis a docentes sem exigir conhecimento técnico especializado, ainda que a revisão docente permaneça como etapa do fluxo.

Palavras-chave: Geração Automática de Questões. *Retrieval-Augmented Generation*. *Middleware* de Ancoragem Curricular. BNCC. Engenharia de *Prompt*. Grandes Modelos de Linguagem.

Abstract

The design of assessments aligned with the Brazilian National Common Curricular Base (BNCC) is a complex and costly process for educators and educational technology companies (*EdTechs*). Although Large Language Models (LLMs) show potential for Automatic Question Generation (AQG), their direct application to the Brazilian normative domain faces two obstacles: the hallucination of structured data, in which the model fabricates or incorrectly associates curricular codes, and the transfer of the complexity of Prompt Engineering to the end user. This work presents the design and implementation of a curricular grounding *middleware* based on the Retrieval-Augmented Generation (RAG) paradigm for the Gradepen educational ecosystem, developed by Machlev Smart Applications. The solution acts as an abstraction layer in the *backend* that infers the user's intent, performs a deterministic retrieval over the official curricular hierarchy modeled in JSON, and orchestrates the construction of an enriched prompt in XML. This prompt incorporates Few-Shot examples, the cognitive level of the Revised Bloom's Taxonomy, and psychometric rules for the formulation of distractors, before querying a general-purpose LLM API. Unlike the more widespread formulation of RAG, the retrieval is deterministic, based on exact matching over the closed and standardized vocabulary of the BNCC. Deployed in a staging environment, the solution runs natively in the *backend*, dispensing with external services and reducing the computational complexity and cost of the operation. The approach seeks to ensure that the curricular codes and contents of the generated questions actually exist in the official base, reducing the incidence of normative hallucinations and making AI resources accessible to teachers without requiring specialized technical knowledge, while teacher review remains a step in the workflow.

Keywords: Automatic Question Generation. Retrieval-Augmented Generation. Curricular Grounding *Middleware*. BNCC. Prompt Engineering. Large Language Models.

Sumário

1 Introdução

O presente trabalho descreve o desenvolvimento de uma solução aos desafios da Geração Automática de Questões (AQG, do inglês *Automatic Question Generation*) no contexto educacional brasileiro, especialmente diante das exigências estabelecidas pela Base Nacional Comum Curricular (BNCC). Esta seção apresenta o panorama tecnológico relacionado ao uso de Grandes Modelos de Linguagem (LLMs, do inglês *Large Language Models*) na educação, discute os desafios de alinhamento pedagógico enfrentados por plataformas educacionais e descreve uma proposta de solução baseada em *middleware*, Engenharia de *Prompt* e *Retrieval-Augmented Generation* (RAG), concebida para conciliar escalabilidade operacional e aderência às diretrizes curriculares nacionais.

1.1 Contexto

Historicamente, a elaboração de materiais didáticos e avaliações representa um dos processos mais custosos e intelectualmente exigentes no ecossistema educacional. A construção manual de questões e exercícios consiste em uma tarefa complexa, que demanda tempo, treinamento, experiência pedagógica e recursos significativos, atuando frequentemente como um gargalo à escalabilidade de tecnologias educacionais [1]. No cenário brasileiro, essa exigência operacional foi intensificada pela homologação da Base Nacional Comum Curricular (BNCC). Este documento normativo[2] descreve as competências e habilidades, essenciais e obrigatórias, na construção de currículos e de instrumentos de avaliação.

Com o objetivo de reduzir custos e atender à crescente demanda por conteúdos alinhados às diretrizes educacionais oficiais, a área de Geração Automática de Questões (AQG) vem sendo pesquisada há décadas. O desenvolvimento histórico desses *frameworks* computacionais demonstra uma evolução contínua, desde sistemas baseados em regras determinísticas e manipulação gramatical, até a chegada de abordagens fundamentadas em Inteligência Artificial Generativa e Grandes Modelos de Linguagem (LLMs - *Large Language Models*) [3, 4].

A popularização recente de modelos fundacionais, impulsionada por arquiteturas como os Transformers, presentes em ferramentas como ChatGPT, GPT-4 e Gemini, transformou significativamente a educação digital. Essas tecnologias reduziram barreiras técnicas relacionadas à produção de conteúdo, permitindo a geração rápida de textos, soluções e explicações com qualidade comparável à de especialistas humanos [5]. No contexto das EdTechs (empresas de tecnologia educacional), tais ferramentas passaram a desempenhar um papel estratégico na escalabilidade da produção de conteúdo [3], viabilizando a automatização da criação de avaliações alinhadas às exigências pedagógicas estabelecidas pela BNCC.

1.2 Problema e solução proposta

Apesar da acessibilidade e do potencial produtivo dos LLMs, a qualidade, veracidade e conformidade pedagógica dos conteúdos gerados dependem fortemente das instruções fornecidas à máquina, prática conhecida como Engenharia de *Prompt*. Educadores e autores de materiais didáticos, em geral, não possuem o conhecimento técnico necessário para elaborar *prompts* complexos que

evitem respostas genéricas, desvios das diretrizes pedagógicas (como a Taxonomia de Bloom) ou alucinações da IA [5, 4].

No cenário educacional brasileiro, a aplicação direta desses modelos de propósito geral enfrenta um obstáculo adicional, *a geração de códigos ou descritores curriculares inexistentes ou associados de forma incorreta às habilidades pretendidas*. Como os LLMs não dispõem internamente à complexa hierarquia de códigos da BNCC, quando instruídos apenas em linguagem natural, eles tendem a inventar ou a associar incorretamente unidades temáticas, objetos de conhecimento e códigos de habilidades [6, 3]. No fluxo operacional de uma empresa de tecnologia educacional, essa ineficiência na interação e a geração de artefatos academicamente inválidos resultam em perda de produtividade e na necessidade constante de revisão manual minuciosa, limitando a escalabilidade do negócio.

Para solucionar este gargalo, este trabalho propõe o desenvolvimento e a integração de uma funcionalidade em escala corporativa denominada “Ancoragem Curricular via RAG”. O projeto teve como premissa fundamental a criação de uma solução estruturalmente alinhada à BNCC. O objetivo foi garantir que os usuários do ecossistema Gradepen pudessem produzir questões formalmente ancoradas nas diretrizes nacionais sem precisarem dominar comandos avançados de Inteligência Artificial.

A solução atua como uma camada de abstração (um *middleware*) na plataforma da empresa, estruturando-se sob o paradigma de *Retrieval-Augmented Generation* (RAG) [7]. O sistema permite que o usuário (professor ou conteudista) digite a menor quantidade possível de texto sobre o tema desejado. Em segundo plano, a ferramenta extrai a intenção do usuário, busca as correspondências exatas em uma base de dados estruturada local da BNCC e preenche uma estrutura otimizada de parâmetros. Esse *prompt* enriquecido, que conterá o contexto curricular oficial, restrições pedagógicas e regras de psicometria, é então processado por uma API de LLM avançada. Como resultado, o sistema retorna artefatos educacionais precisos, com maior aderência às diretrizes curriculares da BNCC e menor incidência de alucinações normativas, tornando-os aptos para aplicação em contextos educacionais reais.

1.3 Justificativa

O desenvolvimento deste *middleware* justifica-se, primordialmente, tanto pelo aumento da eficiência operacional quanto pela maior consistência normativa dos materiais produzidos pela empresa. No contexto do mercado brasileiro de *EdTechs*, materiais avaliativos que não apresentam alinhamento adequado à Base Nacional Comum Curricular (BNCC) tendem a reduzir sua aplicabilidade em instituições de ensino que exigem conformidade com as diretrizes educacionais nacionais. Nesse cenário, a adoção de uma arquitetura RAG acoplada a uma API de LLM de propósito geral configura-se como uma estratégia tecnológica relevante, uma vez que contribui para a mitigação de alucinações algorítmicas ao ancorar a geração de conteúdo em dados curriculares oficiais e estruturados [6, 7].

Adicionalmente, a automação parametrizada da geração de conteúdo reduz significativamente o tempo e o esforço cognitivo despendidos por educadores na elaboração manual de avaliações e na construção de distratores plausíveis. Essa otimização operacional possibilita que os docentes direcionem maior atenção para atividades de maior valor pedagógico, como o acompanhamento dos alunos, a mediação do processo de aprendizagem e a análise diagnóstica dos resultados [5].

Por fim, ao reduzir a necessidade de conhecimento especializado em Inteligência Artificial por parte dos usuários finais, a solução amplia a acessibilidade dessas tecnologias no ecossistema Gradepen. Modelos generativos demandam estruturação adequada de intenção, contexto e formato para alcançar maior qualidade pedagógica [4, 8]. Ao incorporar práticas de Engenharia de *Prompt* diretamente no sistema, incluindo definição taxonômica, injeção de contexto estruturado e de regras de formatação, a ferramenta contribui tanto na padronização quanto no aumento da validade acadêmica dos materiais desenvolvidos. Dessa forma, a solução apresenta potencial de aplicação em larga escala no contexto educacional brasileiro.

2 Trabalhos Relacionados

Esta seção apresenta uma revisão da literatura sobre a evolução das abordagens tecnológicas para a geração automática de artefatos educacionais. O objetivo é traçar a trajetória da área, abrangendo a transição da comunidade científica de modelos locais aos LLMs acessados via API, as dificuldades encontradas na aplicação dessas tecnologias frente às diretrizes curriculares brasileiras e a adoção de estratégias de injeção de conhecimento estruturado, como o paradigma RAG, que fundamenta a solução proposta neste trabalho. Foram priorizados estudos recentes sobre AQG no contexto educacional, com ênfase em trabalhos voltados à língua portuguesa e ao cenário brasileiro, dada a centralidade da BNCC no problema abordado.

2.1 A transição de modelos locais para grandes modelos de linguagem

A automação da criação de avaliações constitui um objeto de investigação de longa data, motivada pela necessidade de reduzir a carga de trabalho docente e viabilizar a escalabilidade de tecnologias educacionais [1]. Embora as abordagens pioneiras da área se fundamentassem em regras determinísticas e em manipulação gramatical [1, 3], a última década concentrou esforços no treinamento ou no ajuste fino (*fine-tuning*) de modelos neurais executados localmente, de menor porte em relação aos grandes modelos de linguagem atuais. No contexto do ensino de tecnologia, [9] explorou a geração automática de exercícios de programação, combinando *templates* e modelos especializados em código, como o CodeT5. Para a língua portuguesa, [10] desenvolveram o *Bequizzer*, uma ferramenta de geração automática de questionários baseada no ajuste fino de modelos da família T5, comparando essa técnica com abordagens de engenharia de *prompt*.

A principal limitação dessas abordagens reside no custo computacional do treinamento e na dependência de bases de dados extensas e bem anotadas, fatores que restringem sua acessibilidade em ambientes educacionais e tornam sua execução inviável em infraestruturas convencionais [1, 9, 3]. Os estudos evidenciam, ainda, que modelos de menor porte apresentam dificuldade em generalizar para novos contextos. Mesmo que produzam textos sintaticamente corretos, os materiais gerados tendem a apresentar incoerências ou a desviar do escopo solicitado, *carecendo da profundidade semântica e da complexidade pedagógica esperadas em situações reais de ensino* [10, 9, 11, 3]. Essa relação desfavorável entre esforço e resultado contribuiu para o deslocamento do interesse da comunidade em direção aos LLMs de grande porte acessados via API, que apre-

sentam maior capacidade de geração, mas transferem o desafio tecnológico para a formulação das instruções — prática conhecida como Engenharia de *Prompt*.

2.2 Engenharia de *Prompt* e o desafio do alinhamento à BNCC

Com a adoção de APIs de LLMs, a literatura passou a investigar como otimizar o desempenho pedagógico dessas ferramentas. Estudos indicam que LLMs modernos são capazes de gerar questões com nível cognitivo comparável ao de educadores humanos [5]. Esse nível costuma ser aferido pela Taxonomia de Bloom, que organiza os objetivos de aprendizagem em uma hierarquia de complexidade crescente de processos, como recordar e compreender até analisar, avaliar e criar, desde que os modelos sejam instruídos por meio de *prompts* estruturados, baseados em exemplos (*Few-Shot*) ou em raciocínio passo a passo (*Chain-of-Thought*) [8].

Contudo, a transposição dessa tecnologia para o contexto educacional brasileiro enfrenta as exigências normativas específicas da Base Nacional Comum Curricular (BNCC). [12] destaca que a elaboração manual de questões de Matemática alinhadas à BNCC constitui uma atividade exaustiva para os docentes, e que a aplicação direta de IA generativa apresenta fragilidades na clareza textual e na calibração do nível de dificuldade. Em estudo sobre o uso de LLMs de propósito geral para a geração de itens em Língua Portuguesa destinados a avaliações brasileiras em larga escala, [13] concluíram que as tecnologias atuais, isoladamente, ainda não atendem por completo aos requisitos de qualidade e de alinhamento normativo exigidos.

Buscando contornar essas limitações, [14] investigou o uso de *Meta-Prompting*, técnica em que *prompts* são utilizados para gerar outros *prompts*, na criação de exercícios de Língua Portuguesa. O estudo observou que, embora a técnica tenha promovido ganhos qualitativos na clareza e na contextualização das questões, as diferenças de qualidade das questões em relação a *prompts* convencionais não foram estatisticamente significativas em nenhum dos critérios avaliados. Esses resultados sugerem que o refinamento isolado da linguagem do *prompt* não é suficiente para suprir a ausência de conhecimento estruturado do modelo sobre os códigos e as diretrizes curriculares brasileiras.

2.3 Injeção de conhecimento estruturado e middlewares de *prompt*

Para mitigar as alucinações e as falhas de alinhamento dos LLMs diante de currículos complexos, pesquisas recentes passaram a integrar bases de dados estruturadas ao processo de geração. Em [6] propuseram o *KG-Quizzer*, um *framework* que integra Grafos de Conhecimento (KGs) a LLMs para a geração de questionários em português. Os autores demonstraram que a recuperação e a injeção de informações estruturadas (triplas factuais) no *prompt* aumentam a precisão factual e a relevância das questões geradas.

Para além da injeção de dados, parte da literatura recente argumenta que a integração escalável dessas soluções em ambientes de produção requer uma camada de orquestração arquitetural, frequentemente designada *prompt middleware*. [15] propuseram um *framework* dessa natureza, capaz de mapear interações simples do usuário em interfaces gráficas para *prompts* estruturados baseados em *templates*. Em linha semelhante, [16] investigou *middlewares* de *prompt* dinâmicos, sugerindo

que o refinamento contextual em tempo de execução permite parametrizar tarefas complexas, equilibrando a facilidade de uso com a robustez das instruções enviadas aos modelos.

A partir desse panorama, delimita-se a lacuna que o presente trabalho visa preencher. Os estudos discutidos aqui convergem em quatro constatações:

- O treinamento de modelos próprios mostrou-se custoso e pedagogicamente limitado [10, 9].
- O acesso direto a LLMs de propósito geral esbarra na dificuldade de educadores em formular *prompts* tecnicamente estruturados [17].
- Técnicas centradas apenas na reformulação textual, como o *Meta-Prompting*, apresentam limitações frente às exigências da BNCC [14].
- A injeção de dados estruturados se destaca como caminho promissor para a precisão pedagógica, e arquiteturas de *middleware* oferecem um meio adequado para abstrair essa complexidade do usuário final [6, 15].

O trabalho aqui proposto situa-se nessa intersecção, articulando as duas frentes identificadas na literatura: a injeção de conhecimento estruturado e a arquitetura de *prompt middleware*. Integrado ao ecossistema Gradepen, o *middleware* de ancoragem curricular desenvolvido abstrai do usuário a complexidade da interação com o modelo. Em segundo plano, adota o paradigma *Retrieval-Augmented Generation* (RAG), técnica que combina a recuperação de informações de uma base externa com a capacidade generativa do modelo, operando de forma conceitualmente análoga à injeção de grafos de conhecimento, mas consumindo dinamicamente uma base estruturada local da BNCC. Dessa forma, o sistema orquestra requisições à API de um LLM de propósito geral, ancoradas em unidades temáticas, objetos de conhecimento e códigos de habilidades oficiais, buscando conciliar a escalabilidade exigida pelo contexto corporativo com a aderência normativa apontada pela literatura como fator crítico de qualidade.

3 A empresa e sua atuação

Nesta seção, é caracterizada a organização parceira onde o presente trabalho foi desenvolvido, detalhando seu histórico e posição no mercado de tecnologia educacional.

3.1 Caracterização da Empresa

A Machlev Smart Applications Serviço e Venda de Sistemas Ltda ME é uma microempresa brasileira inserida no setor terciário da economia, com foco principal no desenvolvimento de programas de computador sob encomenda e no segmento de tecnologia educacional (*EdTech*). Fundada formalmente em 16 de maio de 2017, a empresa possui forte ligação com o ecossistema de inovação da cidade do Recife, Pernambuco [18]. Em seu histórico geográfico, a organização iniciou suas atividades sendo incubada pela INCUBATEP, a incubadora de empresas do Instituto de Tecnologia de Pernambuco (ITEP). Posteriormente, transferiu suas operações para a região da Encruzilhada/Espinheiro e,

consolidando sua expansão comercial e tecnológica, estabeleceu sua sede atual no histórico Bairro do Recife, inserindo-se diretamente na área de influência do Porto Digital, um dos principais parques tecnológicos da América Latina.

O principal produto desenvolvido e operado de forma exclusiva pela Machlev (Machine Learning and Vision) é o ecossistema *Gradepen* [19]. Trata-se de uma plataforma educacional híbrida, composta por um sistema de gerenciamento *web* e um aplicativo móvel (disponível para iOS e Android) [20, 21], cujo objetivo é reduzir o gargalo operacional enfrentado por docentes na elaboração, aplicação e correção de avaliações físicas. Embora a fundação oficial da empresa como pessoa jurídica date de 2017, as raízes tecnológicas do *software* remontam a 2012 com a criação do sistema pioneiro "Questões na Web", que evoluiu estruturalmente até o lançamento da primeira versão do aplicativo Gradepen na App Store, em julho de 2013.

Na área específica de atuação, o Gradepen oferece ferramentas para a criação de provas discursivas e objetivas, com algoritmos antifraude que embaralham automaticamente a ordem das questões e das alternativas. O núcleo de inovação da empresa reside na aplicação de Visão Computacional e de Reconhecimento Óptico de Marcadores (OMR, do inglês *Optical Mark Recognition*), tecnologia focada na identificação automática das marcações e preenchimentos feitos pelos alunos nos cartões de resposta. Esse processamento é executado diretamente na borda (*Edge Computing*), ou seja, ocorre no próprio dispositivo do usuário, o que viabiliza a correção de gabaritos pela câmera do celular mesmo em ambientes escolares com conectividade restrita.

No que tange aos mercados e clientes, a empresa atende principalmente a professores, coordenadores pedagógicos e instituições de ensino de diversos níveis. O sistema adota um modelo de negócios híbrido do tipo *freemium*. A elaboração e diagramação das provas na plataforma *web* são disponibilizadas de forma totalmente gratuita aos docentes, assim como a correção de testes curtos (até cinco questões com cinco alternativas). A escalabilidade para a correção automatizada de exames maiores, contudo, é monetizada por meio de microtransações (compra de pacotes de créditos) [20]. A estrutura organizacional e o desenvolvimento de *software* mantêm um perfil proprietário e de código fechado, visando à preservação da propriedade intelectual dos algoritmos.

3.2 Atuação Profissional e Contexto do Projeto

O autor integra a equipe de Engenharia de Software da Machlev Smart Applications, que opera com metodologias ágeis, como o Scrum, possibilitando entregas incrementais e contínuas de funcionalidades para o ecossistema Gradepen.

Na organização, o autor ocupa a posição de Estagiário de Inteligência Artificial. Em linhas gerais, suas atividades envolvem a manutenção da plataforma web e o desenvolvimento de funcionalidades com foco em IA, integrando novas tecnologias para enriquecer a experiência dos educadores.

Dentro dessa equipe, a área específica de atuação do autor concentrou-se na exploração e na implementação de ferramentas de Inteligência Artificial Generativa. Dado que a comunidade de professores do Gradepen apresentava demanda crescente por novos conteúdos, mas encontrava barreiras técnicas na elaboração de *prompts* adequados, iniciou-se a investigação de métodos para automatizar a geração desses materiais. Foi nesse contexto corporativo que se idealizou o projeto

de geração de *prompts* educacionais, culminando na arquitetura atual, baseada no consumo da API de um LLM de propósito geral. O trabalho executado pelo autor abrangeu desde a prototipação e a validação de modelos menores via *fine-tuning* até a arquitetura final em produção, atuando como principal responsável pela integração de IA na funcionalidade descrita neste trabalho.

4 Desenvolvimento realizado na empresa

Esta seção detalha o desenvolvimento de software realizado pelo autor no ecossistema Gradepen. Descrevem-se a problemática técnica enfrentada, a solução implementada e as tecnologias e decisões arquiteturais que fundamentaram o *middleware* de ancoragem curricular desenvolvido.

4.1 A problemática e a solução proposta

4.1.1 O contexto operacional e o gargalo de produção

A plataforma Gradepen atende a uma comunidade crescente de educadores e autores de materiais didáticos que demandam continuamente a inserção de novos conteúdos avaliativos. A elaboração manual de itens de avaliação constitui um processo intelectualmente oneroso e de difícil escalabilidade, pois cada item requer clareza textual, calibração adequada do nível de dificuldade e a formulação de distratores plausíveis, capazes de mapear diagnósticos de aprendizagem distintos.

No mercado brasileiro de tecnologia educacional, essa complexidade é acentuada pelas exigências da BNCC. Para que um artefato avaliativo possua aplicabilidade prática em instituições de ensino, ele precisa estar formalmente ancorado em uma matriz multidimensional que envolve Unidades Temáticas, Objetos de Conhecimento e Habilidades. Cada habilidade é identificada por um código alfanumérico padronizado que sintetiza sua posição nessa matriz. No código `EF07MA01`, por exemplo, “EF” indica a etapa (Ensino Fundamental), “07” o ano de escolaridade, “MA” o componente curricular (Matemática) e “01” a ordem da habilidade naquele recorte, organização regular que se estende por toda a estrutura curricular [2]. A dimensão dessa matriz ajuda a explicar o esforço envolvido. Considerando apenas o Ensino Fundamental, sem incluir a Educação Infantil e o Ensino Médio, a BNCC organiza dez componentes curriculares ao longo dos nove anos de escolaridade, o que resulta em 1.443 objetos de conhecimento e 2.173 habilidades, cada uma identificada por um código único. A Figura 1 ilustra um recorte dessa estrutura para um único componente e ano. Localizar manualmente o descritor correto em meio a esse volume, para cada item de avaliação produzido, é uma tarefa que exige tempo e conhecimento especializado.

Esse processo manual de correspondência curricular representa uma oportunidade de automação relevante, na medida em que a elaboração alinhada à BNCC demanda treinamento de conteudistas e revisão humana cuidadosa, especialmente quando se busca ampliar o volume de conteúdos avaliativos disponíveis.

MATEMÁTICA – 7º ANO

UNIDADES TEMÁTICAS	OBJETOS DE CONHECIMENTO	HABILIDADES
Números	Múltiplos e divisores de um número natural	**(EF07MA01)** Resolver e elaborar problemas com números naturais, envolvendo as noções de divisor e de múltiplo, podendo incluir máximo divisor comum ou mínimo múltiplo comum, por meio de estratégias diversas, sem a aplicação de algoritmos.
	Cálculo de porcentagens e de acréscimos e decréscimos simples	**(EF07MA02)** Resolver e elaborar problemas que envolvam porcentagens, como os que lidam com acréscimos e decréscimos simples, utilizando estratégias pessoais, cálculo mental e calculadora, no contexto de educação financeira, entre outros.
	Números inteiros: usos, história, ordenação, associação com pontos da reta numérica e operações	**(EF07MA03)** Comparar e ordenar números inteiros em diferentes contextos, incluindo o histórico, associá-los a pontos da reta numérica e utilizá-los em situações que envolvam adição e subtração. **(EF07MA04)** Resolver e elaborar problemas que envolvam operações com números inteiros.
	Fração e seus significados: como parte de inteiros, resultado da divisão, razão e operador	**(EF07MA05)** Resolver um mesmo problema utilizando diferentes algoritmos. **(EF07MA06)** Reconhecer que as resoluções de um grupo de problemas que têm a mesma estrutura podem ser obtidas utilizando os mesmos procedimentos. **(EF07MA07)** Representar por meio de um fluxograma os passos utilizados para resolver um grupo de problemas. **(EF07MA08)** Comparar e ordenar frações associadas às ideias de partes de inteiros, resultado da divisão, razão e operador. **(EF07MA09)** Utilizar, na resolução de problemas, a associação entre razão e fração, como a fração 2/3 para expressar a razão de duas partes de uma grandeza para três partes da mesma ou três partes de outra grandeza.

Figura 1: Recorte da estrutura curricular da BNCC para a Matemática do 7º ano, evidenciando a organização hierárquica em unidades temáticas, objetos de conhecimento e habilidades, cada uma com seu código normativo (Brasil 2018).

4.1.2 As limitações dos LLMs de propósito geral no domínio normativo

Com a disseminação dos LLMs acessados via APIs comerciais, abriu-se a oportunidade de automatizar a geração de questões em larga escala. A plataforma já contava com recursos de geração de conteúdo apoiados por IA. O que se buscou, neste trabalho, foi estender essa capacidade com um tratamento específico para o alinhamento à BNCC, considerando duas dimensões pouco contempladas por uma abordagem de uso geral.

A primeira dimensão é a usabilidade, para obter itens com o rigor exigido em avaliações de larga escala, é necessário formular solicitações bastante estruturadas, o que nem sempre está ao alcance de professores e conteudistas sem familiaridade com técnicas de instrução de modelos. A segunda dimensão, de natureza computacional, decorre de uma característica geral dos LLMs de propósito geral. Por não disporem de representação interna determinística da hierarquia da BNCC, esses modelos podem produzir alucinações normativas, isto é, gerar códigos inexistentes ou associar de forma incorreta habilidades e objetos de conhecimento. Garantir aderência estrita aos descritores oficiais é, portanto, um requisito que demanda uma camada dedicada de validação.

4.1.3 O *middleware* de ancoragem curricular baseado em RAG

Para atender a esse requisito, o autor projetou e implementou o *middleware* de ancoragem curricular, que constitui o núcleo da funcionalidade de geração automática de questões alinhadas à BNCC. Estruturado sob o paradigma RAG, o *middleware* atua entre a interface do usuário final e

a API de um LLM de propósito geral. Em vez de permitir que o modelo gere a correspondência curricular de forma probabilística, o sistema recupera a hierarquia real de uma base de dados local estruturada da BNCC e a injeta no fluxo de geração, restringindo as escolhas do modelo a descritores oficiais existentes.

A contribuição do autor articula-se em dois componentes complementares. O primeiro é a camada de recuperação, validação e orquestração curricular, responsável por consultar uma base estruturada da BNCC e restringir a geração aos descritores oficiais. O segundo é a definição de um modo de operação especializado para a BNCC, que reúne instruções, blocos de contexto estruturado e regras psicométricas elaboradas pelo autor. Optou-se, deliberadamente, por integrar esse modo à capacidade de geração já existente na plataforma, concentrando o esforço de desenvolvimento na camada de ancoragem curricular, que é a contribuição original deste trabalho. A Figura 2 apresenta a interface dessa funcionalidade, na qual o usuário pode descrever o tema em linguagem natural para o preenchimento automático dos campos curriculares ou selecioná-los manualmente pela hierarquia da BNCC.

(a) Estado inicial, com o campo de extração automática vazio e a seleção por hierarquia recolhida.

(b) Após a extração a partir do tema “Álgebra para 9º ano”, com os campos curriculares preenchidos e validados.

Figura 2: Funcionalidade de contextualização de questões com a BNCC na plataforma Gradepen, em dois momentos do fluxo de uso. A interface sinaliza os campos confirmados e exibe o código oficial da habilidade recuperada (e.g., `EF09MA06`) (O autor, 2026).

4.2 Tecnologias utilizadas

Esta subseção apresenta as principais tecnologias e técnicas empregadas no desenvolvimento do *middleware*, descrevendo o papel de cada uma no projeto. A forma como elas se articulam no fluxo de geração é detalhada na Seção 4.3.

O ***Retrieval-Augmented Generation*** **(RAG)** é um paradigma que combina uma etapa de recuperação de informações de uma base externa com a capacidade generativa de um modelo de linguagem, de modo que a geração se fundamente em dados reais, e não apenas no conhecimento

interno do modelo [7]. Neste trabalho, o RAG sustenta toda a solução. Antes de gerar uma questão, o sistema recupera a hierarquia curricular oficial da BNCC e a insere no contexto enviado ao modelo. Esse paradigma foi escolhido porque ancora a geração nos descritores normativos reais e reduz a ocorrência de alucinações normativas, sem exigir o treinamento de um modelo próprio [7, 6].

Os **Grandes Modelos de Linguagem (LLMs)** são modelos de grande porte, baseados em arquiteturas neurais, capazes de interpretar e gerar texto em linguagem natural com boa fluência [3]. No projeto, um LLM de propósito geral, acessado via API, faz a inferência nas etapas de extração de intenção, seleção curricular e geração da questão. A opção por consumir um modelo já treinado, em vez de aplicar ajuste fino (*fine-tuning*) a um modelo local, deve-se à qualidade de geração em língua portuguesa e à ausência dos altos custos de infraestrutura e manutenção de um modelo próprio [9, 11].

A **Engenharia de *Prompt*** é a prática de formular instruções que orientam o comportamento de um LLM. No projeto, é por meio dela que a geração se especializa para o domínio da BNCC, com instruções de sistema que atribuem ao modelo o papel de especialista, blocos de contexto pedagógico e exemplos no formato *Few-Shot* (demonstrações inseridas no *prompt* para orientar o formato da resposta) [22, 4]. As instruções também incorporam o nível cognitivo da Taxonomia de Bloom Revisada [23, 8] e regras psicométricas para a elaboração de distratores. Essas técnicas foram usadas porque permitem controlar a qualidade pedagógica e o formato da saída sem alterar o modelo subjacente.

A **API REST** (*Representational State Transfer*) é um estilo de comunicação entre sistemas no qual as requisições trafegam sobre o protocolo HTTP, seguindo convenções arquiteturais restritas para o acesso a recursos [24]. No projeto, é por meio de uma API REST que o *backend* se comunica com o LLM, enviando os *prompts* construídos e recebendo as respostas estruturadas do modelo. Esse modelo de integração foi adotado por ser o padrão de acesso oferecido pelos provedores de LLMs e por se encaixar diretamente no ambiente da plataforma, sem a necessidade de dependências adicionais.

A **recuperação determinística por correspondência exata** difere da formulação mais comum do paradigma RAG [7], que tradicionalmente recupera dados por similaridade semântica entre vetores. No projeto, a recuperação curricular baseia-se em correspondência exata sobre as chaves da árvore da BNCC, complementada por uma rotina de correspondência aproximada por subcadeia de caracteres para os casos em que a disciplina e o ano não são identificados diretamente. A abordagem foi escolhida por se adequar ao vocabulário fechado e padronizado da BNCC [2], no qual a correspondência exata é mais previsível e econômica que uma busca aproximada. Essa arquitetura dispensa a infraestrutura de geração e armazenamento de vetores, mitigando alucinações ao basear-se exclusivamente em dados estruturados estáticos [6].

O **PHP** é a linguagem de execução no lado do servidor sobre a qual a plataforma foi construída em uma arquitetura monolítica [25]. Toda a lógica do *middleware*, desde a recuperação curricular até a montagem do contexto, foi implementada nativamente nesse ambiente. A decisão de integrar a solução ao *backend* existente, em vez de mantê-la isolada em um microsserviço externo, simplificou o processo de *deploy* e dispensou infraestrutura adicional, com ganho de coesão arquitetural.

O **JSON** (*JavaScript Object Notation*) é um formato leve de representação de dados estruturados em pares chave-valor [26]. No projeto, a complexa hierarquia da BNCC [2] foi mapeada e armaze-

nada localmente em uma base nesse formato, organizada em uma árvore de cinco níveis, que vão da disciplina até as habilidades, passando por ano, unidade temática e objeto de conhecimento. O formato foi adotado por permitir a navegação eficiente da estrutura curricular em memória, o que viabiliza a recuperação determinística sem latência ou consultas a serviços externos.

O **XML** (*eXtensible Markup Language*) é um formato de marcação que organiza dados em uma hierarquia estrita de elementos [27]. No projeto, ele estrutura o *prompt* da etapa de geração final, na qual as seções da instrução e o contexto curricular são organizados em elementos delimitados, uma prática recomendada na engenharia de *prompt* para isolar instruções de sistema e guiar a atenção do modelo [28]. Esse é também o formato no qual a questão é devolvida à plataforma, por se adequar à representação estruturada do enunciado, das alternativas e dos metadados de cada item gerado. Enquanto as etapas anteriores de extração e seleção operam em JSON, a transição para o XML na geração final favorece a organização rigorosa tanto da instrução de entrada quanto da resposta gerada.

4.3 Arquitetura de integração

Esta subseção descreve como as tecnologias apresentadas se articulam no fluxo de geração, desde a estruturação dos dados curriculares até a devolução da questão à plataforma.

4.3.1 Infraestrutura de *backend* e estruturação dos dados

A lógica de orquestração curricular foi encapsulada em um componente nativo do *backend* da plataforma, desenvolvido pelo autor, que concentra as operações de recuperação, validação e montagem do contexto enviado ao modelo.

Para suprir a ausência de conhecimento estruturado do modelo acerca da regulação educacional, a hierarquia da BNCC foi mapeada e armazenada localmente em uma base no formato JSON, com aproximadamente 682 KB, organizada na árvore de cinco níveis descrita anteriormente. Injetar a árvore completa em cada requisição seria inviável em termos de latência e de custo por *tokens*, e por isso o sistema realiza a filtragem prévia do escopo pedagógico diretamente em memória, antes de qualquer chamada ao modelo.

4.3.2 Recuperação curricular e aumento de contexto

A recuperação principal recebe a disciplina e o ano de escolaridade e percorre a árvore curricular em busca do ponto correspondente, retornando a subárvore completa daquele recorte, isto é, todas as unidades temáticas, objetos de conhecimento e habilidades daquela disciplina e ano. Antes da comparação, os valores de entrada passam por uma normalização (caixa baixa, remoção de espaços e do indicador ordinal) que torna a busca tolerante a variações de digitação, de modo que formas como “1º” e “1” ou “Ciências” e “ciências” resultem em correspondência. Conceitualmente, a operação equivale a uma consulta condicionada por disciplina e ano sobre uma estrutura hierárquica, e não a uma busca por proximidade de significado.

Como a recuperação exata depende de a disciplina e o ano terem sido identificados corretamente, o sistema conta com um mecanismo complementar de correspondência aproximada. Sempre que algum dos campos curriculares está ausente, mas há ao menos um indício textual de unidade temática, objeto de conhecimento ou habilidade, essa rotina varre a árvore curricular procurando correspondências por subcadeia de caracteres. A cada candidato é atribuída uma pontuação ponderada que reflete a especificidade do indício, na qual uma correspondência de unidade temática vale um ponto, a de objeto de conhecimento dois, e a de habilidade três, de modo que os sinais mais específicos predominem sobre os mais genéricos. O caminho de maior pontuação preenche os campos ausentes, com uma salvaguarda, o ano de escolaridade só é inferido quando a pontuação acumulada atinge ao menos dois pontos, o que evita associações ambíguas a partir de um indício que, sozinho, abrangeria vários anos. Dessa forma, uma entrada que mencione apenas um conteúdo reconhecível, como "equações", pode ser resolvida com o recorte curricular correspondente sem qualquer chamada ao modelo.

Recuperado o recorte curricular, ocorre a etapa de aumento de contexto. A subárvore é inserida no *prompt* enviado ao modelo, acompanhada da instrução de selecionar a unidade temática, o objeto de conhecimento e a habilidade, copiando fielmente os textos presentes naquele recorte. Como o modelo passa a enxergar apenas descritores que de fato existem na BNCC e é instruído a não se afastar deles, a geração fica ancorada nesses dados, o que reduz substancialmente a possibilidade de fabricação de códigos inexistentes.

A formulação final das instruções apoia-se no construtor de *prompts* já existente na plataforma, responsável pela montagem padronizada de instruções de sistema e exemplos *Few-Shot*. A contribuição deste trabalho estendeu esse construtor com um modo de operação especializado para a BNCC, ativado quando há dados curriculares disponíveis para a requisição. As principais diferenças entre esse modo e o modo padrão estão sintetizadas na Tabela 1.

Tabela 1: Comparativo entre o modo de operação padrão do construtor de *prompts* e o modo especializado para a BNCC desenvolvido pelo autor

Componente	**Modo padrão**	**Modo especializado para a BNCC**
Instrução de sistema	Define o papel do assistente e as regras de formatação da plataforma.	Acrescenta uma instrução que atribui ao modelo o papel de especialista em *design* educacional e na BNCC.
Exemplos anteriores (*Few-Shot*)	Questões já cadastradas pelo usuário, usadas como referência de estilo e formato.	Mesmo mecanismo, sem alterações.
Contexto da questão	Tipo, nível de ensino, disciplinas, assuntos e instruções livres do usuário.	Acrescenta blocos estruturados com o contexto pedagógico e a habilidade da BNCC (código e descrição).
Orientação cognitiva e psicométrica	Sem indicação explícita do nível cognitivo nem regras para as alternativas incorretas.	Define o nível da Taxonomia de Bloom Revisada e regras para a elaboração dos distratores.

Fonte: O autor (2026).

Nesse modo, a instrução de sistema padrão é acrescida de uma instrução complementar que atribui ao modelo o papel de especialista em *design* educacional e insere blocos estruturados de

contexto pedagógico, além de regras de formatação de saída, como delimitadores para fórmulas matemáticas e marcações estruturadas. O modo orienta a geração pelo nível cognitivo pretendido, fundamentado na Taxonomia de Bloom Revisada [23]. Para itens de múltipla escolha, acrescenta regras psicométricas para a construção dos distratores a partir de erros lógicos recorrentes, como interpretação equivocada, inversão causal e generalização indevida.

4.3.3 Orquestração do *pipeline*

As etapas descritas articulam-se em um *pipeline* de quatro estágios, dos quais três envolvem chamadas ao modelo de linguagem e um é executado integralmente em memória, sem custo de API.

1. **Extração de campos base (1ª chamada).** O sistema recebe o texto livre digitado pelo usuário (e.g., *"Quero uma questão de matemática sobre equações de primeiro grau para o 7º ano, nível de aplicação"*) e faz uma primeira requisição ao modelo para inferir os metadados da intenção. Nessa etapa, o *prompt* restringe explicitamente os valores admissíveis de cada campo, e o modelo é instruído a retornar um valor nulo quando não houver inferência segura, para que não preencha por suposição. O retorno é um objeto JSON com os campos inferidos (disciplina, ano, nível da Taxonomia de Bloom e perfil do aluno, entre outros), que nesta fase ainda constituem estimativas, não necessariamente correspondentes ao texto oficial da BNCC.

2. **Recuperação local (sem consumo de *tokens*).** De posse da disciplina e do ano inferidos, a recuperação opera inteiramente sobre a base estruturada, sem qualquer chamada ao modelo, recortando a subárvore curricular correspondente. Quando a primeira chamada não identifica esses campos com segurança, a correspondência aproximada descrita anteriormente reconstrói o caminho provável antes da recuperação exata, de modo que mesmo entradas vagas resultem em um recorte curricular válido.

3. **Seleção curricular assistida (2ª chamada).** A subárvore recortada é enviada ao modelo, com a instrução de selecionar unidade temática, objeto de conhecimento e habilidade copiando fielmente os textos do recorte. Como as opções se restringem a descritores oficiais existentes, a possibilidade de fabricação de códigos é substancialmente reduzida.

4. **Geração final (3ª chamada).** Com as variáveis curriculares validadas, o construtor de *prompts*, operando no modo especializado, monta o *payload* definitivo em XML, reunindo instrução de sistema, exemplos *Few-Shot*, regras psicométricas e de formatação. A questão é então gerada e devolvida à plataforma no mesmo formato, adequado à estrutura do enunciado, das alternativas e dos metadados.

O Código 1 ilustra a estrutura da instrução empregada na primeira chamada, com o texto do usuário já interpolado.

Código 1: Estrutura do *prompt* de extração de campos base (1ª chamada).

```
1 Você é um especialista em educação brasileira e na BNCC.
2 Analise o texto abaixo e preencha os campos que conseguir
```

```
inferir com segurança.
IMPORTANTE: para cada campo, o valor retornado deve ser
EXATAMENTE um dos textos admitidos.
Use valor nulo apenas quando não for possível inferir
- não invente.
Retorne SOMENTE o objeto JSON, sem marcações.

Texto: "Quero uma questão de matemática sobre equações
de primeiro grau para o 7o ano, nível de aplicação"

Campos:
{
  "disciplina": "EXATAMENTE um da lista - nulo se
   não identificável",
  "ano": "ordinal do EF sem 'ano' (1o-9o) - nulo
   se ausente",
  "unidadeTematica": "... - nulo se incerto",
  "objetoConhecimento": "... - nulo se incerto",
  "habilidade": "código BNCC se mencionado - nulo
   se ausente",
  "perfilAluno": "inferido do texto; um de: Básico
   | Intermediário | Avançado | Bom domínio
   | Com dificuldades - nulo se não inferível",
  "nivelBloom": "inferido pelo verbo cognitivo; um de:
   Lembrar | Compreender | Aplicar | Analisar
   | Avaliar | Criar - nulo se não inferível",
  "tipoTextoBase": "inferido do tema e disciplina;
   um de: Narrativo | Argumentativo
   | Científico/Técnico | Instrucional | Jornalístico
   | Literário | Histórico/Documental | Publicitário
   | Sem texto-base - nulo se não inferível"
}
```

O modelo analisa o texto livre e infere, além dos campos curriculares, o perfil cognitivo do aluno, o nível taxonômico de Bloom e o tipo de texto-base mais adequado à disciplina. O Código 2 apresenta a resposta obtida para o exemplo em questão.

Código 2: Resposta do modelo à primeira chamada, com todos os campos inferidos.

```
{
  "disciplina": "Matemática",
  "ano": "7o",
  "unidadeTematica": "Álgebra",
  "objetoConhecimento": "Equações polinomiais do 1o grau",
  "habilidade": null,
  "perfilAluno": "Intermediário",
  "nivelBloom": "Aplicar",
  "tipoTextoBase": "Científico/Técnico"
}
```

Nota-se que o campo `habilidade` permanece nulo, pois o texto do usuário não mencionou um código BNCC específico. Esse campo será preenchido na segunda chamada, que recebe a subárvore curricular completa da disciplina e do ano recuperados.

O Código 3 ilustra a instrução da segunda chamada, com o recorte curricular já inserido no *prompt*. Esse recorte reúne a subárvore completa da disciplina e do ano recuperados, com todas as suas unidades temáticas, objetos de conhecimento e habilidades, de modo que a seleção do modelo ocorra sobre o conjunto íntegro de descritores válidos.

Código 3: Estrutura do *prompt* de seleção curricular (2ª chamada), com o recorte da BNCC inserido.

```
Você é um especialista na Base Nacional Comum Curricular (BNCC).

TAREFA
Dado o tema do usuário e a estrutura BNCC real abaixo, escolha
a unidade temática, o objeto de conhecimento e a habilidade
mais adequados. Use EXATAMENTE os textos que aparecem na
estrutura, sem alterações.

TEMA DO USUÁRIO
"Quero uma questão de matemática sobre equações de primeiro grau para o 7o ano, nível
    de aplicação"

ESTRUTURA BNCC - Matemática, 7o ano
{
  "Números": { ... },
  "Álgebra": {
    "Linguagem algébrica: variável e incógnita": [
      "(EF07MA13) Compreender a ideia de variável...",
      "(EF07MA15) Utilizar a simbologia algébrica..."
    ],
    "Equações polinomiais do 1o grau": [
      "(EF07MA18) Resolver e elaborar problemas que possam ser
       representados por equações polinomiais de 1o grau,
       redutíveis à forma ax + b = c, fazendo uso das
       propriedades da igualdade."
    ]
  },
  "Geometria": { ... },
  "Grandezas e medidas": { ... },
  "Probabilidade e estatística": { ... }
}

FORMATO DE SAÍDA
Retorne APENAS um JSON válido:
{ "unidadeTematica": "", "objetoConhecimento": "", "habilidade": "" }
```

Diante desse recorte, o modelo seleciona a habilidade pertinente copiando seu texto integralmente, como mostra o Código 4.

Código 4: Resposta do modelo à segunda chamada, com os descritores copiados fielmente da BNCC.

```
{
  "unidadeTematica": "Álgebra",
  "objetoConhecimento": "Equações polinomiais do 1o grau",
  "habilidade": "(EF07MA18) Resolver e elaborar problemas que
```

```
    possam ser representados por equações polinomiais de 1o grau,
    redutíveis à forma ax + b = c, fazendo uso das propriedades
    da igualdade."
}
```

A correspondência exata entre os textos da resposta e os da estrutura fornecida assegura a ancoragem curricular e evita que o modelo crie ou reformule descritores. O modelo apenas seleciona, dentro do conjunto oficial, os que melhor atendem ao tema solicitado.

Com os descritores curriculares ancorados e os campos pedagógicos inferidos, o *pipeline* dispõe de todos os metadados necessários para compor a chamada de geração final. As duas chamadas anteriores operam com *prompts* planos e respostas em JSON. A terceira é diferente, pois emprega um *payload* estruturado em XML e organizado em múltiplos turnos, no formato de conversação da API do modelo. Esse *payload* reúne uma instrução de sistema separada, turnos de demonstração no estilo *few-shot* e, ao final, o contexto pedagógico enriquecido com as especificações da BNCC, o nível da Taxonomia de Bloom e as regras psicométricas para a construção das alternativas. O Código 5 apresenta uma versão resumida desse *payload*.

Código 5: Estrutura do *payload* de geração final (3ª chamada), em formato XML multiturnos.

```
<request model="llm">
  <system-instruction>
    Você atua como especialista em design educacional e
    elaboração de itens para exames de larga escala, com
    profundo conhecimento da BNCC e da Taxonomia de Bloom.
    Gere uma questão inédita, original e pedagogicamente
    robusta, seguindo rigorosamente as especificações
    BNCC fornecidas a seguir.

    Seu objetivo é ajudar os professores a criar questões
    objetivas e discursivas para as avaliações.
    (...)                regras de formato omitidas por sigilo
    - Equações em LaTeX: <latex>\frac{1}{2}</latex>
    - Finalize cada texto com: <|endoftext|>
  </system-instruction>

  <contents>
    <!-- ====== FEW-SHOT: avaliação-exemplo ====== -->
    <turn role="user">=== Nova Avaliação ===</turn>

    <turn role="user">
      Tipo: Múltipla Escolha
      Nível: Fundamental
      Disciplinas: Matemática
      Assuntos: Expressões numéricas
      Enunciado:
    </turn>
    <turn role="model">
      Calcule o valor da expressão
      <latex>3 + 2 \times (5 - 1)</latex>.<|endoftext|>
    </turn>
    <turn role="user">Alternativas:</turn>
```

```
    <turn role="user">1. </turn>
    <turn role="model">
      item marcado: 11<|endoftext|>
    </turn>
    <turn role="user">2. </turn>
    <turn role="model">
      item desmarcado: 20<|endoftext|>
    </turn>
    (...)          demais exemplos few-shot omitidos

    <!-- == AVALIAÇÃO ATUAL: questões existentes == -->
    <turn role="user">=== Nova Avaliação ===</turn>
    (...)          questões prévias da mesma prova

    <!-- ====== QUESTÃO NOVA - CONTEXTO BNCC ====== -->
    <turn role="user">
      Tipo: Múltipla Escolha
      Nível: 7o ano, Fundamental
      Disciplinas: Matemática
      Assuntos: Equações polinomiais do 1o grau

      === CONTEXTO PEDAGÓGICO E ALUNO-ALVO ===
      Perfil do Aluno: Intermediário
      Tópico Principal: Álgebra
      Assunto Específico: Equações polinomiais do 1o grau

      === ESPECIFICAÇÕES BNCC ===
      Nível de Bloom: Aplicar
      Código BNCC: EF07MA18
      Descrição BNCC: (EF07MA18) Resolver e elaborar
        problemas que possam ser representados por equações
        polinomiais de 1o grau, redutíveis à forma
        ax + b = c, fazendo uso das propriedades da
        igualdade.
      Tipo de Texto Base: Científico/Técnico
        (a questão deve ser precedida por texto de apoio
         conciso, sem entregar a resposta)

      === REGRAS PARA AS ALTERNATIVAS ===
      (...) regras de alternativas omitidas por sigilo

      Enunciado:
    </turn>
  </contents>

  <generation-config>
    <stop-sequences>
      <seq><|endoftext|></seq>
    </stop-sequences>
  </generation-config>
</request>
```

É importante notar que, na plataforma, o enunciado e as alternativas são gerados em turnos separados. O modelo primeiro produz o texto do enunciado e, em seguida, cada alternativa é solicitada

individualmente por meio de um novo turno `user`, o que permite ao sistema acumular o contexto da conversa a cada iteração. O Código 6 ilustra essa sequência de respostas.

Código 6: Resposta iterativa do modelo à terceira chamada: enunciado seguido das alternativas geradas uma a uma.

```
 - Turno 1: modelo gera o enunciado -
<turn role="model">
  Em Matemática, uma equação do primeiro grau é uma
  igualdade que envolve uma ou mais variáveis
  (incógnitas), onde o maior expoente dessas variáveis
  é 1. Para resolver essas equações e encontrar o valor
  da incógnita, aplicamos as propriedades da igualdade,
  buscando isolar a variável por meio de operações
  inversas nos dois lados da equação.

  Com base nas propriedades das equações do primeiro
  grau, analise a seguinte situação-problema: Um
  fazendeiro comprou 3 sacas de ração e mais 5 kg
  avulsos. No total, ele adquiriu 50 kg de ração.
  Considerando que cada saca tem a mesma quantidade
  de ração, qual é o peso, em quilogramas, de cada
  saca?<|endoftext|>
</turn>

 - Turnos 2-6: alternativas geradas uma a uma -
<turn role="user">Alternativas:</turn>

<turn role="user">1. </turn>
<turn role="model">
  item desmarcado: 10Kg<|endoftext|>
</turn>

<turn role="user">2. </turn>
<turn role="model">
  item desmarcado: 12Kg<|endoftext|>
</turn>

<turn role="user">3. </turn>
<turn role="model">
  item marcado: 15Kg<|endoftext|>
</turn>

<turn role="user">4. </turn>
<turn role="model">
  item desmarcado: 42Kg<|endoftext|>
</turn>

<turn role="user">5. </turn>
<turn role="model">
  item desmarcado: 45Kg<|endoftext|>
</turn>
```

A resposta correta (`item marcado`: 15 kg) corresponde à solução da equação $3x + 5 = 50$, co-

erente com a habilidade EF07MA18 e com o nível *Aplicar* da Taxonomia de Bloom. Os distratores exploram erros típicos do conteúdo, como a confusão entre operações (10 kg e 12 kg), a subtração simples sem divisão (45 kg) e a inversão parcial do procedimento (42 kg), conforme as regras psicométricas do *prompt*. A transição de formato entre as chamadas é intencional. As duas primeiras usam *prompts* planos com saída em JSON, adequada à extração e à seleção de campos estruturados, enquanto a terceira adota o formato multiturnos da API, com instrução de sistema isolada e turnos alternados entre `user` e `model`. Os turnos de demonstração estabelecem o padrão de formatação sem regras explícitas adicionais, e o bloco final, com o contexto da BNCC, ancora a geração no currículo oficial e nas boas práticas de elaboração de itens.

A Figura 3 apresenta a questão resultante já renderizada na interface do Gradepen, como é exibida ao professor, com o enunciado, o texto de apoio e as alternativas formatados para uso em uma avaliação.

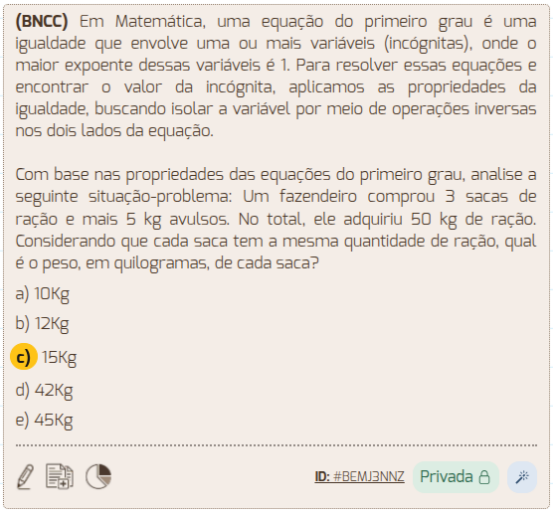

Figura 3: Questão gerada ao final do *pipeline*, renderizada na interface do Gradepen e ancorada na habilidade `EF07MA18` da BNCC (O autor, 2026).

A Figura 4 sintetiza o fluxo completo descrito nesta seção, das três chamadas ao modelo à etapa local de recuperação, evidenciando a ordem em que a extração, a recuperação curricular e a geração final se articulam.

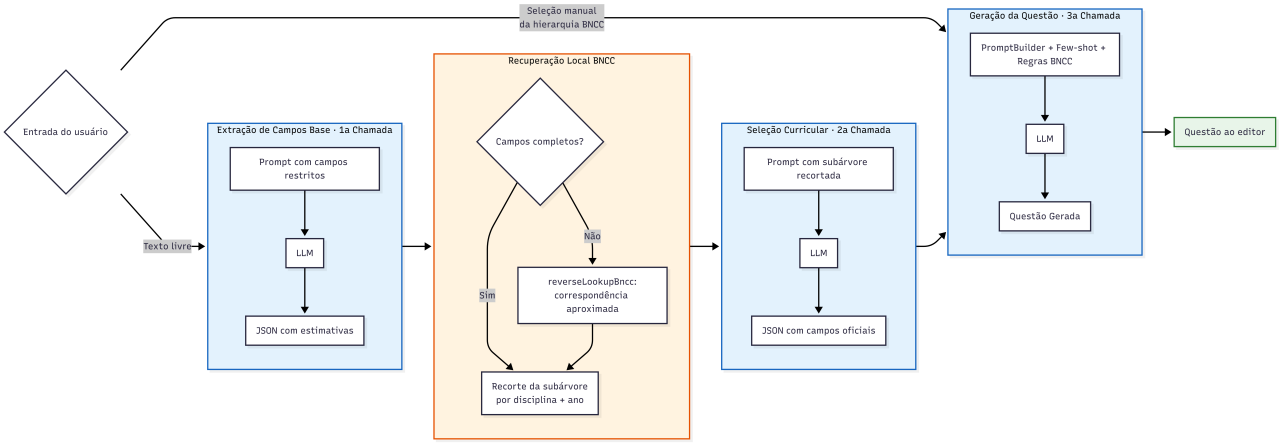

Figura 4: Fluxo lógico do *pipeline* de geração com ancoragem curricular (O autor, 2026).

4.4 Contribuição e impacto na empresa

A contribuição técnica descrita encontra-se implantada no ambiente de homologação da Machlev Smart Applications. O *middleware* de ancoragem curricular tem sido exercitado, desde o final de maio de 2026, em cenários de teste que antecedem sua eventual disponibilização aos usuários da plataforma.

Do ponto de vista da Engenharia de Software, a solução foi integrada de forma nativa ao *backend* da plataforma, reaproveitando a infraestrutura já existente, como o construtor de *prompts*, em vez de introduzir novos serviços ou componentes externos. Para a empresa, isso significa uma funcionalidade que opera sem servidores dedicados e com menos pontos de falha, integrada às rotinas de manutenção e *deploy* já praticadas pela equipe. Soma-se a esse ganho o fato de a recuperação curricular ocorrer localmente, o que evita o consumo de *tokens* faturáveis que ocorreria caso essa etapa fosse delegada ao modelo, contribuindo para um menor custo computacional da funcionalidade. A Tabela 2 resume os principais indicadores de operação da solução nesse ambiente.

Tabela 2: Indicadores de operação do *middleware* de ancoragem curricular em ambiente de homologação.

Indicador	**Valor**
Chamadas ao modelo por extração curricular	2
Etapas de recuperação executadas localmente, sem consumo de *tokens*	1
Tamanho da base curricular completa	aproximadamente 682 KB
Tamanho médio do recorte enviado ao modelo	algumas centenas de *bytes*
Consumo médio de *tokens* por extração	1562

Fonte: O autor (2026).

Do ponto de vista pedagógico, a ancoragem determinística assegura que os códigos curriculares presentes nas questões geradas existam, de fato, na base oficial da BNCC, ainda que a qualidade pedagógica do enunciado permaneça sujeita à revisão docente, etapa que é preservada no fluxo da plataforma. Ao incorporar técnicas de Engenharia de *Prompt* diretamente no sistema, a solução busca reduzir o esforço cognitivo exigido dos professores na elaboração de avaliações, permitindo que forneçam apenas o tópico desejado em linguagem natural. Por fim, a arquitetura desenvolvida não se restringe à BNCC. Como o alinhamento normativo decorre da base curricular estruturada que é consultada, e não da lógica do *middleware* em si, a mesma abordagem pode ser estendida a outros marcos avaliativos, como as matrizes de referência do ENADE ou do ENEM, bastando substituir a base consultada na etapa de recuperação e realizar as adaptações necessárias à estrutura de cada marco. Essa generalidade amplia o alcance potencial da funcionalidade dentro do portfólio do Gradepen, abrindo caminho para a geração de itens ancorados em diferentes referenciais oficiais.

5 Dificuldades encontradas

O desenvolvimento deste trabalho no ecossistema da Machlev Smart Applications trouxe dificuldades tanto técnicas quanto de alinhamento estratégico. Enfrentá-las foi o que mais contribuiu para meu amadurecimento profissional ao reorientar o projeto de uma perspectiva inicialmente acadêmica

para uma visão mais pragmática, voltada à Engenharia de Software e à entrega de valor.

A primeira dificuldade técnica esteve na escolha da abordagem de Inteligência Artificial. A concepção original previa aplicar ajuste fino (*fine-tuning*), técnica que consiste em treinar adicionalmente um modelo já existente sobre um conjunto de dados específico para adaptá-lo a uma tarefa, a um modelo de código aberto da plataforma Hugging Face, o *paraphrase-multilingual-MiniLM-L12-v2* [29, 30], voltado a tarefas de paráfrase e similaridade semântica em língua portuguesa. Essa escolha tinha um viés acadêmico, motivado pelo interesse em explorar o treinamento de redes neurais locais. Os primeiros testes, porém, mostraram que manter e treinar um modelo próprio exigiria infraestrutura de *hardware* robusta (GPUs), alta complexidade computacional e esforço contínuo de manutenção, o que destoava da agilidade e da entrega contínua exigidas no ambiente da empresa.

Diante disso, decidiu-se reorientar a arquitetura. Em vez do ajuste fino local, optou-se pelo consumo da API de Grandes Modelos de Linguagem (LLMs) ancorada pelo paradigma RAG [7]. Ainda que orquestrar um RAG com APIs comerciais possa parecer, à primeira vista, uma abordagem mais simples do que treinar os pesos de um modelo, ela se mostrou bem mais ágil de desenvolver, escalar e manter. Essa percepção prática vai ao encontro da literatura, que aponta a dificuldade de modelos locais menores em abstrair contextos pedagógicos complexos [10, 11]. A transição consolidou uma das principais lições do trabalho. A solução tecnologicamente mais sofisticada nem sempre é a melhor. A boa engenharia de software não está em adotar a tecnologia mais complexa, mas em escolher a ferramenta adequada ao contexto e às restrições do ambiente de negócio.

Outro desafio relevante foi integrar essas tecnologias de IA à infraestrutura já consolidada da empresa. O consumo de uma API de LLM teve um efeito positivo nesse ponto, pois tornou possível manter a lógica dentro da própria estrutura existente, sem depender de um serviço externo. Escrever a orquestração RAG e a busca reversa nativamente no *backend* foi um trabalho de codificação considerável, mas simplificou o *deploy* da funcionalidade e dispensou a manutenção de um componente apartado.

Por fim, atuar na Engenharia de IA exigiu adquirir novas competências em pouco tempo. A Engenharia de *Prompt* foi um aprendizado constante, foi preciso aprender a parametrizar a linguagem natural em instruções determinísticas, com controles rígidos sobre o formato de saída e a injeção de regras psicométricas para guiar o modelo e reduzir suas alucinações. O desenvolvimento deste *middleware* mostrou, na prática, que o trabalho do Engenheiro de Software e IA vai além de escrever algoritmos lógicos tradicionais, exigindo a habilidade de orquestrar a colaboração entre sistemas clássicos, bases de dados estruturadas e a natureza probabilística da Inteligência Artificial.

6 Impactos da formação no trabalho realizado

A base técnica e lógica para a execução deste projeto começou a ser construída antes do ingresso no ensino superior. Minha formação técnica de nível médio, no curso de Desenvolvimento de Sistemas da ETE Advogado José David Gil Rodrigues, deu-me um alicerce prático voltado à programação e ao desenvolvimento de sistemas.

No Bacharelado em Sistemas de Informação da UFRPE, essa base inicial amadureceu e ganhou uma dimensão de arquitetura de software. As disciplinas voltadas à Engenharia de Software e ao

desenvolvimento de sistemas me ajudaram a enxergar a solução não como um código que apenas funciona, mas como um software que precisa ser mantido e evoluído ao longo do tempo. Esse repertório sustentou decisões de projeto como a forma de integrar o *middleware* ao ecossistema da plataforma, com atenção à manutenção e à facilidade de *deploy*.

O contato com a área de Dados e Inteligência Artificial foi o que mais direcionou meus estudos, e acabou se tornando o foco da minha formação. As disciplinas ligadas a armazenamento e análise de dados, inteligência artificial e estatística me deram a bagagem teórica para atuar além da Engenharia de Software convencional. Foi esse embasamento que me permitiu assumir a integração de IA neste projeto. Entender algoritmos probabilísticos, estruturas de dados e o funcionamento dos Grandes Modelos de Linguagem (LLMs) foi o que tornou possível projetar a mecânica de RAG e a Engenharia de *Prompt*. Esse conhecimento me ajudou a tratar a IA não como uma caixa preta, mas como uma ferramenta estatística que precisa de restrições algorítmicas para não gerar alucinações, como a busca reversa que implementei na recuperação curricular.

As disciplinas de caráter prático e integrador, voltadas a projetos interdisciplinares e à construção de soluções mais complexas, desenvolveram em mim a capacidade de aplicar tecnologias recentes a problemas reais. No fim, foi a soma da mentalidade de desenvolvimento que trouxe da formação técnica com a visão de dados e IA construída na graduação que me permitiu entregar esta solução à empresa, consolidando minha atuação na fronteira entre a Engenharia de Software e a IA.

7 Conclusão

Este trabalho descreveu o desenvolvimento de um *middleware* de ancoragem curricular para a geração automática de questões alinhadas à BNCC, integrado ao ecossistema Gradepen, da Machlev Smart Applications. O ponto de partida foi um problema concreto do contexto educacional brasileiro, a dificuldade de fazer com que Grandes Modelos de Linguagem de propósito geral produzam questões fiéis aos descritores oficiais da BNCC sem que o usuário precise dominar técnicas avançadas de instrução de modelos.

A solução combinou duas frentes. A primeira foi uma camada de recuperação curricular baseada no paradigma RAG, que consulta uma base estruturada da BNCC e restringe as escolhas do modelo aos descritores que de fato existem no documento normativo, reduzindo a ocorrência de alucinações normativas. A segunda foi a extensão do construtor de *prompts* já existente na plataforma com um modo de operação especializado, responsável por incorporar contexto pedagógico, o nível cognitivo da Taxonomia de Bloom Revisada e regras para a elaboração de distratores. Uma característica que distingue a abordagem é o uso de recuperação determinística por correspondência exata, em vez de busca semântica por vetores, escolha viabilizada pelo vocabulário fechado e padronizado da BNCC, e que permite filtrar o escopo curricular localmente, sem custo de inferência.

Do ponto de vista da empresa, a contribuição foi implantada em ambiente de homologação e opera de forma nativa no *backend* da plataforma, reaproveitando a infraestrutura existente em vez de depender de serviços externos. Com isso, a funcionalidade dispensa servidores dedicados, simplifica o processo de *deploy* e mantém parte do processamento localmente, o que reduz a complexidade e o custo computacional da operação. No plano pedagógico, ao encapsular a complexidade da Engenharia de *Prompt*, o sistema permite que docentes gerem avaliações balizadas pela Taxonomia

de Bloom Revisada e com distratores mais consistentes sem treinamento especializado em IA, ainda que a revisão docente permaneça como etapa do fluxo.

Para além do caso específico, o trabalho sugere que *pipelines* de RAG podem ser incorporados a sistemas monolíticos legados aproveitando a infraestrutura já consolidada, sem a necessidade de microsserviços apartados ou reescritas amplas. Essa observação pode ser útil a outras equipes que enfrentem a integração de recursos de IA generativa em bases de código maduras.

Como trabalhos futuros, abre-se a possibilidade de uma avaliação quantitativa mais ampla, comparando a qualidade pedagógica e a aderência normativa das questões antes e depois da disponibilização da funcionalidade aos usuários, além da medição de indicadores de latência e de consumo de *tokens* em uso real. Uma segunda direção é a generalização da arquitetura para outros referenciais avaliativos, como as matrizes do ENEM e do ENADE, mediante o mapeamento de cada marco em uma estrutura equivalente. Há ainda espaço para um *pipeline* de validação automática pós-geração, no qual um segundo modelo atuaria como auditor da qualidade das questões, verificando a consistência dos distratores e a correção do gabarito. Por fim, a exploração de capacidades multimodais, para itens que envolvam imagens, gráficos e equações, representa uma fronteira promissora para o enriquecimento da plataforma.

Referências Bibliográficas

[1] G. Kurdi, J. Leo, B. Parsia, U. Sattler, and S. Al-Emari, "A systematic review of automatic question generation for educational purposes," *International Journal of Artificial Intelligence in Education*, vol. 30, no. 1, pp. 121–204, 2020.

[2] BRASIL. Ministério da Educação, *Base Nacional Comum Curricular: Educação Infantil e Ensino Fundamental*, Secretaria de Educação Básica, Brasília, 2018, acesso em: 7 jun. 2026. [Online]. Available: https://www.gov.br/mec/pt-br/escola-em-tempo-integral/BNCC_EI_EF_110518_versaofinal.pdf

[3] G. R. Rockembach and L. H. Thom, "The last decade of automatic question generation: A review of techniques, limitations, and applications in business process management education," *Revista Brasileira de Informática na Educação*, vol. 34, pp. 219–251, 2026.

[4] S. Sarsa, P. Denny, A. Hellas, and J. Leinonen, "Automatic generation of programming exercises and code explanations using large language models," in *Proceedings of the 2022 ACM Conference on International Computing Education Research V.1*. New York: ACM, 2022, pp. 27–43.

[5] J. Doughty, Z. Wan, A. Bompelli, J. Qayum, T. Wang, J. Zhang, Y. Zheng, A. Doyle, P. Sridhar, A. Agarwal *et al.*, "A comparative study of ai-generated (gpt-4) and human-crafted mcqs in programming education," in *Proceedings of the Australian Computing Education Conference (ACE 2024)*. New York: ACM, 2024, pp. 114–123.

[6] D. Medeiros, G. Leite, A. G. Regino, V. J. S. Chico, F. d. F. Rosa, and J. C. dos Reis, "Kg-quizzer: Refinamento de prompts via grafos de conhecimento para a geração automática de questionários em português," in *Anais do XVIII Seminário de Pesquisa em Ontologias no Brasil (ONTOBRAS 2025)*. Porto Alegre: CEUR-WS.org, 2025, pp. 38–50.

[7] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel *et al.*, "Retrieval-augmented generation for knowledge-intensive nlp tasks," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020, pp. 9459–9474.

[8] N. Scaria, S. D. Chenna, and D. Subramani, "Automated educational question generation at different bloom's skill levels using large language models: Strategies and evaluation," in *International Conference on Artificial Intelligence in Education*. Berlin: Springer, 2024, pp. 165–179.

[9] T. C. Freitas, "Automatic generation of programming exercises," Dissertação (Mestrado em Engenharia Informática), Universidade do Minho, Braga, 2024.

[10] V. J. S. Chico, J. F. Tessler, R. Bonacin, and J. C. dos Reis, "Bequizzer: Ai-based quiz automatic generation in the portuguese language," in *Natural Language Processing and Information Systems*. Berlin: Springer, 2024, pp. 237–248.

[11] M. M. Santos, A. P. Barros, E. Santos, J. G. da Silva, S. Isotani, I. I. Bittencourt, V. Macario, L. Rodrigues, and D. Dermeval, "Geração de questões com llms leves: Um estudo inicial sobre a percepção de educadores," in *Anais do XXXVI Simpósio Brasileiro de Informática na Educação (SBIE 2025)*. SBC, 2025, pp. 1636–1647.

[12] M. B. d. Melo, “Uso de grandes modelos de linguagem na geração automática de questões de matemática: Uma abordagem baseada em design science research,” Trabalho de Conclusão de Curso (Bacharelado em Engenharia de Computação), Instituto Federal da Paraíba, Campina Grande, 2025.

[13] J. V. d. C. M. Ferreira Nogueira, J. A. P. d. Castro, L. O. Larcher, R. V. J. Ferreira, B. T. Barbosa, and J. F. de Souza, “Uma análise de qualidade do uso de grandes modelos de linguagem para geração automática de itens avaliativos em português,” in *Anais do XXXVI Simpósio Brasileiro de Informática na Educação (SBIE 2025)*. SBC, 2025, pp. 234–246.

[14] W. S. Cataldo, “Estudo sobre o uso de meta-prompting em llms para a elaboração de exercícios: Na perspectiva de docentes,” Dissertação (Mestrado em Informática), Universidade Federal do Estado do Rio de Janeiro (UNIRIO), Rio de Janeiro, 2025.

[15] S. MacNeil, A. Tran, J. Kim, Z. Huang, S. Bernstein, and D. Mogil, “Prompt middleware: Mapping prompts for large language models to ui affordances,” in *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology (UIST ’22)*. ACM, 2022, também disponível em: https://arxiv.org/abs/2307.01142.

[16] I. Drosos, J. Williams, A. Sarkar, N. Wilson, S. Rintel, and P. Panda, “Dynamic prompt middleware: Contextual prompt refinement controls for comprehension tasks,” in *Proceedings of the 4th Annual Symposium on Human-Computer Interaction for Work (CHIWORK ’25)*. ACM, 2025.

[17] L. Kuusemets, K. Parve, K. Ain, and T. Kraav, “Assessing ai-generated (gpt-4) versus human created mcqs in mathematics education: A comparative inquiry into vector topics,” *International Journal of Education in Mathematics, Science, and Technology*, vol. 12, no. 6, pp. 1538–1558, 2024.

[18] Serasa Experian, “Machlev smart applications servico e venda de sistemas ltda - me - 27750739000177,” 2026, acesso em: 26 jun. 2026. [Online]. Available: https://empresas.serasaexperian.com.br/consulta-gratis/MACHLEV-SMART-APPLICATIONS-SERVICO-E-VENDA-DE-SISTEMAS-LTDA-ME-27750739000177

[19] Machlev Smart Applications, “Gradepen - crie, corrija e compartilhe avaliações e gabaritos,” 2026, acesso em: 26 jun. 2026. [Online]. Available: https://www.gradepen.com/

[20] Apple Inc., “Gradepen na app store,” 2026, acesso em: 26 jun. 2026. [Online]. Available: https://apps.apple.com/br/app/gradepen/id654152733

[21] Google LLC, “Gradepen - apps no google play,” 2026, acesso em: 26 jun. 2026. [Online]. Available: https://play.google.com/store/apps/details?id=com.gradepen

[22] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan *et al.*, “Language models are few-shot learners,” in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020, pp. 1877–1901.

[23] L. W. Anderson and D. R. Krathwohl, *A taxonomy for learning, teaching, and assessing: A revision of Bloom’s taxonomy of educational objectives*. Addison Wesley Longman, 2001.

[24] R. T. Fielding, “Architectural styles and the design of network-based software architectures,” Ph.D. dissertation, University of California, Irvine, 2000.

[25] The PHP Group, *PHP: A simple tutorial - Manual*, 2026. [Online]. Available: https://www.php.net/manual/en/tutorial.php

[26] ECMA International, “The json data interchange syntax,” ECMA International, Standard ECMA-404, 2017. [Online]. Available: https://www.ecma-international.org/publications-and-standards/standards/ecma-404/

[27] T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, and F. Maler, “Extensible markup language (xml) 1.0 (fifth edition),” World Wide Web Consortium (W3C), Tech. Rep., 2008. [Online]. Available: https://www.w3.org/TR/xml/

[28] Anthropic, *Use XML tags to delimit text*, Anthropic Documentation, 2024. [Online]. Available: https://docs.anthropic.com/en/docs/build-with-claude/prompt-engineering/use-xml-tags

[29] SentenceTransformers, “paraphrase-multilingual-minilm-l12-v2,” 2021, acesso em: 19 jun. 2026. [Online]. Available: https://huggingface.co/sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2

[30] N. Reimers and I. Gurevych, “Sentence-bert: Sentence embeddings using siamese bert-networks,” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, 2019, pp. 3982–3992.