UNIVERSIDADE
FEDERAL RURAL
DE PERNAMBUCO

Jéssica Alves de Souza

Utilização de IA Generativa para a Geração e Validação de Questões com Base na Teoria da Resposta ao Item

Recife

Agosto de 2025

Jéssica Alves de Souza

Utilização de IA Generativa para a Geração e Validação de Questões com Base na Teoria da Resposta ao Item

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.

Universidade Federal Rural de Pernambuco – UFRPE
Departamento de Estatística e Informática
Curso de Bacharelado em Sistemas de Informação

Orientador: Gabriel Alves de Albuquerque Junior

Recife
Agosto de 2025

Utilização de IA Generativa para a Geração e Validação de Questões com Base na Teoria da Resposta ao Item

Jéssica Alves de Souza [1], **Gabriel Alves de Albuquerque Junior** [1]

[1]Departamento de Estatística e Informática – Universidade Federal Rural de Pernambuco
Rua Dom Manuel de Medeiros, s/n, - CEP: 52171-900 – Recife – PE – Brasil,

jessica.alvess@ufrpe.br, gabriel.alves@ufrpe.br

Resumo. *Este trabalho teve como objetivo investigar a utilização da IA generativa na geração de exercícios educacionais e na simulação de respostas de estudantes, com o intuito de aplicar a Teoria de Resposta ao Item (TRI) aos resultados da simulação e verificar se os exercícios gerados correspondem ao nível de dificuldade solicitado, calibrando o simulador. Para isso, foi desenvolvida uma arquitetura de serviços web em FastAPI que orquestra chamadas a Grandes Modelos de Linguagem (LLMs) por meio de templates de prompts contextualizados, permitindo a criação de questões objetivas e dissertativas no estilo ENEM, além da simulação de respostas de mil estudantes fictícios cujas habilidades foram amostradas a partir de uma distribuição gaussiana. A reestimação dos parâmetros a, b e c do modelo logístico de três parâmetros (ML-3P) via bootstrap foi avaliada quantitativamente por meio de métricas de erro. Qualitativamente, as questões produzidas seguiram o padrão ENEM e atenderam às regras predefinidas. No teste quantitativo de seis questões com o Prompt A para determinar sua dificuldade, obteve-se 50 % de acerto. Nas classificações incorretas, as estimativas permaneceram coerentes com a faixa correta, com valores próximos aos seus limites. Por fim, a simulação de respostas forneceu subsídios quantitativos valiosos para aprimorar os prompts e a aplicação dos modelos, aproximando mais os resultados dos dados reais.*

Abstract. *This work aimed to investigate the use of generative AI in the creation of educational exercises and the simulation of student responses, with the goal of applying Item Response Theory (IRT) to the simulation results and verifying whether the generated exercises match the requested difficulty level, thereby calibrating the simulator. To this end, a web services architecture was developed in FastAPI to orchestrate calls to Large Language Models (LLMs) via contextualized prompt templates, enabling the creation of multiple-choice and open-ended questions in the ENEM style, as well as the simulation of responses from one thousand fictitious students whose abilities were sampled from a Gaussian distribution. The re-estimation of the three-parameter logistic model (3PL) parameters a, b, and c via bootstrap was evaluated quantitatively using error metrics. Qualitatively, the generated questions adhered to the ENEM format and met all predefined rules. In a quantitative test of six questions using Prompt A to determine difficulty, 50% of classifications were correct. In the incorrect cases, the estimates remained consistent with the correct difficulty range, with values close to their category boundaries. Finally, the response simulations provided valuable quantitative insights for refining both the prompts and the model application, bringing the results closer to real-world data.*

1. Introdução

A elaboração de questões de múltipla escolha, apesar de parecer simples, exige rigor metodológico para garantir sua validade e confiabilidade. Itens mal estruturados podem aumentar os erros nas avaliações, enquanto a violação de boas práticas de escrita compromete a qualidade das provas e introduz erros sistemáticos que afetam os resultados. Esses aspectos evidenciam a necessidade de critérios claros e consistentes na formulação e validação de instrumentos avaliativos [Romão and Sá 2019].

Para apoiar essa avaliação rigorosa, a Teoria da Resposta ao Item (TRI) surge como uma ferramenta essencial. Ela permite analisar cada questão detalhadamente, verificando dificuldade, capacidade de discriminação e consistência, além de identificar itens problemáticos e possibilitar a redução do número de perguntas sem comprometer a confiabilidade da avaliação [Edelen and Reeve 2007].

Partindo dessa perspectiva, a IA generativa tem se mostrado uma ferramenta promissora para aprimorar o processo educacional. Ela possibilita personalizar o aprendizado, automatizar tarefas repetitivas, como a correção de exercícios, e liberar tempo para que os educadores se dediquem a atividades de maior valor. Além disso, pode apoiar a inovação pedagógica, facilitar a tradução de materiais e criar experiências interativas, onde estudantes interagem com tutores virtuais [Baidoo-Anu 2023].

Diante desse cenário, este trabalho propõe investigar o uso do prompt engineering como estratégia para otimizar o ensino e a aprendizagem, com ênfase na geração e simulação automatizada de questões. A partir da construção de prompts bem estruturados, busca-se explorar o potencial da IA generativa na produção de materiais instrucionais relevantes e adaptados a diferentes contextos, avaliando a qualidade e a consistência dos conteúdos gerados. Assim, pretende-se demonstrar como a integração entre IA e técnicas de prompt engineering pode contribuir para uma educação mais eficiente.

1.1. Objetivos

O objetivo geral deste trabalho é investigar o uso de Grandes Modelos de Linguagem (LLMs, do inglês *Large Language Models*) na geração automática de exercícios educacionais e na simulação de respostas de estudantes fictícios, com ênfase na avaliação da qualidade, consistência e dificuldade dos itens produzidos a partir da Teoria da Resposta ao Item (TRI). Para isso, propõe-se o desenvolvimento de serviços web que integrem LLMs, aplicando engenharia de prompts em cada etapa, de modo a apoiar tanto a geração quanto a validação dos exercícios.

Objetivos específicos

1. Desenvolver uma API RESTful para gerenciar o fluxo de geração de exercícios e simulação de respostas com LLMs;
2. Elaborar e refinar templates de prompts para:
    - geração de questões objetivas e dissertativas no estilo ENEM;
    - simulação de respostas com base em diferentes níveis de habilidade dos estudantes.
3. Produzir conjuntos de exercícios variando parâmetros como nível de dificuldade, tópico e número de questões;

4. Simular respostas de estudantes com habilidades amostradas de uma distribuição gaussiana e reestimar os parâmetros a, b e c do modelo TRI em itens com parâmetros previamente conhecidos;
5. Avaliar a qualidade e consistência dos exercícios gerados por meio de métricas quantitativas, como erro absoluto médio (MAE), erro quadrático médio (MSE) e intervalos de confiança;
6. Verificar se os exercícios atendem aos níveis de dificuldade solicitados a partir das simulações de respostas.

1.2. Organizacão do trabalho

A Seção 2 discute o referencial teórico, abordando temas relevantes para este trabalho, como prompt engineering. A Seção 3 apresenta os trabalhos relacionados, destacando o uso da IA generativa na educação. A Seção 4 descreve a abordagem proposta, detalhando a API desenvolvida, os prompts e as métricas utilizadas. A Seção 5 apresenta os resultados obtidos, analisando os achados do trabalho. Por fim, a Seção 6 traz as considerações finais e sugestões para trabalhos futuros.

2. Referencial teórico

Esta seção apresenta os principais conceitos que fundamentam este trabalho, abordando desde os fundamentos da Teoria de Resposta ao Item (TRI) até as práticas modernas de uso de modelos de linguagem para geração e validação de exercícios educacionais.

2.1. Teoria de Resposta ao Item (TRI)

A Teoria da Resposta ao Item (TRI) é um modelo utilizado para analisar testes e questionários, com o objetivo de compreender como as respostas aos itens estão relacionadas à habilidade dos indivíduos. Essa habilidade interna, denominada traço latente (representada por θ), influencia diretamente a chance de acerto: quanto maior a habilidade, maior a probabilidade de resposta correta. A relação entre habilidade e probabilidade é descrita por fórmulas matemáticas, como a Curva Característica do Item (CCI), que ilustra o aumento da probabilidade de acerto conforme cresce θ. A TRI permite uma análise mais precisa e individualizada tanto dos itens quanto dos participantes, sendo mais informativa e robusta que os modelos clássicos de avaliação [Pasquali and Primi 2003].

Na Teoria da Resposta ao Item (TRI), o Modelo Logístico de 3 Parâmetros (ML3P) é amplamente utilizado para analisar itens com respostas dicotômicas (certo/errado). Esse modelo descreve a probabilidade de um indivíduo j, com traço latente θ_j (habilidade), responder corretamente ao item i por meio da equação (1) [Araujo et al. 2009].

$$P(U_{ij} = 1 \mid \theta_j) = c_i + (1 - c_i)\,\frac{1}{1 + \exp\left[-D\,a_i(\theta_j - b_i)\right]} \tag{1}$$

onde:

- θ_j é o traço latente do respondente, representando sua habilidade.
- a_i é o **parâmetro de discriminação**, que indica o quão bem o item diferencia indivíduos com diferentes níveis de habilidade. Valores mais altos de a_i produzem curvas mais inclinadas, ou seja, melhor discriminação. Valores negativos são indesejados.

- b_i é o **parâmetro de dificuldade**, representando o nível de habilidade necessário para que a probabilidade de acerto seja $(1 + c_i)/2$. Quanto maior o valor de b_i, mais difícil é o item.
- c_i é o **parâmetro de acerto casual**, ou seja, a probabilidade de indivíduos com baixa habilidade acertarem o item por sorte.
- D é uma constante de escala. Quando $D = 1$, usamos a forma padrão da função logística; já com $D = 1{,}7$, a curva se aproxima da forma da ogiva normal.

O modelo de 3 parâmetros (ML-3P) é mais flexível e realista, pois considera simultaneamente a dificuldade, a capacidade de discriminação dos itens e a chance de acerto por adivinhação. Já os modelos logísticos de 2 e 1 parâmetro são simplificações: o primeiro desconsidera o chute ($c_i = 0$), e o segundo, além disso, assume que todos os itens têm o mesmo valor de discriminação (a_i constante). A escolha do modelo ideal depende dos objetivos da análise e das características dos dados.

Graficamente, a Figura 1 apresenta diferentes curvas características do item (CCI) obtidas a partir do modelo logístico de três parâmetros (3PL). Cada curva representa um item de múltipla escolha com parâmetros distintos, permitindo visualizar como pequenas variações em *discriminação* (a), *dificuldade* (b) e *acerto ao acaso* (c) afetam o comportamento da probabilidade de acerto em função da habilidade do estudante (θ).

De forma mais detalhada:

- O Item 1 trata-se de um item fácil, pois sua curva está deslocada para a esquerda, sendo respondido corretamente mesmo por estudantes com baixa habilidade. Contudo, a discriminação é baixa ($a = 0{,}6$), de modo que o item não separa bem estudantes de níveis próximos de proficiência.
- O Item 2 por estar centralizado no eixo da habilidade, ele representa um item de dificuldade média, útil para diferenciar estudantes em torno de $\theta = 0$, com boa capacidade discriminativa.
- O Item 3 é de dificuldade intermediária-alta. Sua curva é mais inclinada, indicando alta discriminação, o que significa que consegue separar com maior precisão estudantes cujas habilidades estão próximas de $b = 0{,}8$.
- O Item 4 trata-se de um item difícil, pois apenas estudantes com alto nível de habilidade apresentam probabilidade elevada de acerto. Sua discriminação é muito alta, tornando-o adequado para diferenciar entre estudantes de desempenho avançado.

Assim, o conjunto de CCIs ilustra diferentes combinações possíveis de parâmetros e como essas interações moldam o comportamento probabilístico do acerto nos itens, desde questões muito fáceis e pouco discriminativas até itens difíceis e altamente discriminativos.

Por fim, parâmetros do modelo TRI apresentam limites teóricos amplos, mas na prática são usados valores mais restritos. O parâmetro de discriminação a pode, em teoria, assumir qualquer valor real ($-\infty < a < +\infty$), mas normalmente fica entre $-2{,}80$ e $+2{,}80$. O parâmetro de dificuldade b também é, teoricamente, livre ($-\infty < b < +\infty$), porém costuma variar entre -3 e $+3$. Já o parâmetro de chute c varia de 0 a 1 em teoria, mas na prática valores acima de $0{,}35$ são considerados pouco plausíveis, de modo que é adotado $0 \leq c \leq 0{,}35$ [Baker 2001].

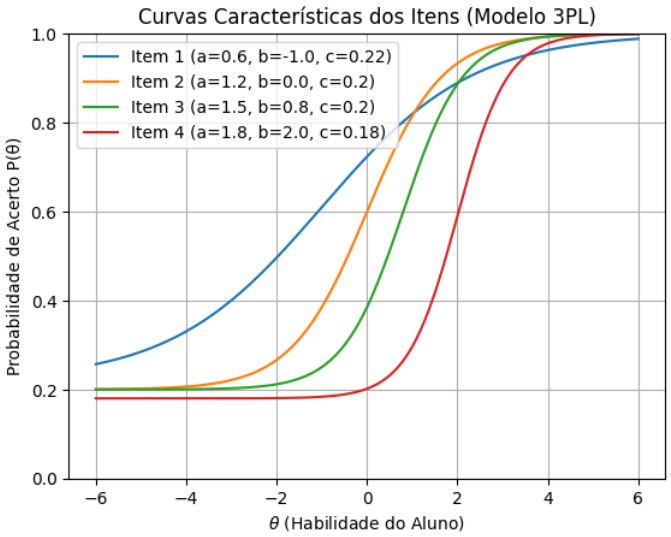

Figura 1. Curvas características de itens simulados no modelo 3PL com diferentes combinações de parâmetros a, b e c. Fonte: Elaborado pela autora, 2025.

2.2. Métricas de avaliação

As métricas de avaliação são ferramentas fundamentais para analisar o desempenho de modelos de aprendizado de máquina. Elas permitem quantificar a diferença entre os valores previstos pelo modelo e os valores reais. No contexto de regressão ou classificação, essas métricas ajudam a identificar se o modelo está fazendo boas previsões, sendo essenciais para guiar decisões como ajustes de parâmetros, escolha de modelos e validação de resultados. Neste trabalho, adotam-se as seguintes notações para todas as métricas: x_i representa o valor real (observado) e $\hat{x}_i$ representa o valor previsto pelo modelo para a mesma instância i.

Erro Médio de Viés (MBE) O MBE (do inglês *Mean bias error*) é uma métrica utilizada para quantificar o viés médio entre os valores previstos. Ele é calculado pela média das diferenças entre as previsões e os valores observados, conforme equação (2). Um valor de MBE negativo indica que o modelo tende a superestimar os resultados, enquanto um valor positivo sugere subestimação. Essa métrica é útil para identificar a presença de viés sistemático nas previsões e pode indicar a necessidade de calibragens adicionais [Piotrowski et al. 2022].

$$\mathrm{MBE}_x = \frac{1}{m}\sum_{i=1}^{m}\left(x_i - \hat{x}_i\right) \tag{2}$$

Erro Absoluto Médio (MAE) O MAE (do inglês *Mean absolute error*) é uma métrica de avaliação cujo melhor valor é 0 e não possui um limite superior definido. Ele é útil quando os outliers nos dados de treinamento representam informações corrompidas, pois a norma L1 suaviza esses erros, evitando penalizações excessivas. No entanto, caso o conjunto de teste também contenha muitos outliers, o desempenho do modelo pode parecer

inferior quando avaliado por essa métrica [Chicco et al. 2021]. Conforme a Equação (3), o MAE é definido como a média dos valores absolutos das diferenças entre os valores observados x_i e os valores previstos $\hat{x}_i$.

$$\mathrm{MAE}_x = \frac{1}{m}\sum_{i=1}^{m}\left|x_i - \hat{x}_i\right| \tag{3}$$

Erro Quadrático Médio (MSE) O MSE (do inglês *Mean square error*) é uma métrica que calcula a média dos quadrados das diferenças entre os valores observados e previstos, conforme equação (4). Ele é ótimo para atribuir maiores pesos a essas diferenças, Por elevar os erros ao quadrado, penaliza fortemente previsões distantes, sendo útil na identificação de outliers. Quanto mais perto de 0, melhor é o resultado, sendo 0 o valor perfeito [Chicco et al. 2021].

$$\mathrm{MSE}_x = \frac{1}{m}\sum_{i=1}^{m}\left(x_i - \hat{x}_i\right)^2 \tag{4}$$

Raiz do Erro Quadrático Médio (RMSE) O RMSE (do inglês *Root Mean Squared Error*) é uma métrica usada para medir o quão distantes estão as previsões feitas por um modelo em relação aos valores reais. Ele é calculado extraindo-se a raiz quadrada do MSE, conforme a Equação (5), o que permite comparar tudo na mesma escala dos valores originais. Quanto mais próximo de zero for o RMSE, melhor a qualidade da predição, sendo 0 a previsão perfeita. Valores mais altos indicam erros maiores [Vujović et al. 2021].

$$\mathrm{RMSE}_x = \sqrt{\mathrm{MSE}_x} \tag{5}$$

Erro Percentual Absoluto Médio (MAPE) O MAPE (do inglês *Mean absolute percentage error*) é uma métrica utilizada para avaliar modelos de regressão com base na diferença percentual entre valores observados x_i e previstos $\hat{x}_i$, como mostra a equação (6).

$$\mathrm{MAPE}_x = \frac{1}{m}\sum_{i=1}^{m}\left|\frac{x_i - \hat{x}_i}{x_i}\right| \tag{6}$$

Seu valor ideal é 0%, e não há limite superior. É indicado para análises em termos relativos, mas apresenta limitações: exige dados positivos e tende a superestimar o erro em previsões baixas, sendo inadequado quando há grandes discrepâncias [Chicco et al. 2021].

2.3. Intervalo de Confiança

O intervalo de confiança (IC) de 95% representa uma faixa de valores plausíveis para a média de uma população, estimada a partir de uma amostra. Como a média populacional

verdadeira é desconhecida, o IC fornece uma estimativa de onde esse valor provavelmente se encontra. Isso significa que, se múltiplas amostras fossem extraídas da mesma população e um IC de 95% fosse calculado para cada uma delas, esperar-se que cerca de 95% desses intervalos contivessem o valor real da média populacional. A largura do intervalo depende diretamente da variabilidade dos dados e do tamanho da amostra: quanto maior a variabilidade ou menor a amostra, mais amplo tende a ser o IC. Embora o valor verdadeiro da média seja fixo, o intervalo reflete a incerteza decorrente do processo de amostragem, assumindo que esta foi aleatória [O'Brien and Yi 2016]. Dessa forma, o intervalo de confiança contribui tanto para estimar a precisão dos resultados quanto para compreender a importância estatística e o tamanho do efeito das diferenças entre os grupos analisados.

2.3.1. Bootstrap

Em contextos em que os dados apresentam assimetrias ou em que a distribuição da população não é conhecida, o método bootstrap surge como uma abordagem robusta para estimar propriedades estatísticas. Essa técnica consiste em gerar diversas amostras com reposição a partir da amostra original e, a partir delas, calcular a estatística de interesse (como a média). O conjunto resultante de estatísticas compõe a distribuição bootstrap, a qual pode ser utilizada para estimar o erro padrão, o viés e os intervalos de confiança associados ao estimador. Embora o centro dessa distribuição reflita a média amostral e não o valor verdadeiro da população, sua dispersão e assimetria tendem a reproduzir bem as propriedades da distribuição amostral. Assim, o bootstrap é especialmente útil para quantificar a incerteza de estimativas sem depender de pressupostos fortes sobre a forma da distribuição populacional, sendo frequentemente aplicado em situações onde métodos analíticos tradicionais falham ou são inviáveis devido à complexidade dos dados ou à presença de vieses [Hesterberg 2011].

2.4. Inteligência artificial generativa

A inteligência artificial generativa (IA generativa) é uma tecnologia capaz de criar novos conteúdos, como textos, imagens, sons e vídeos, a partir de padrões aprendidos por meio de grandes volumes de dados. Utilizando modelos como redes neurais e aprendizado profundo, a IA generativa consegue analisar dados existentes e gerar resultados inovadores que simulam a criatividade humana. Um exemplo popular dessa tecnologia é o ChatGPT, que, a partir de simples comandos, é capaz de criar textos complexos e detalhados. Além de suas aplicações em áreas como educação, marketing e saúde, a IA generativa levanta importantes questões sobre ética, segurança e a qualidade das informações produzidas. Embora seja uma ferramenta poderosa para automação e inovação, seu uso também traz preocupações sobre a transparência e confiabilidade dos dados gerados, sendo fundamental que seu desenvolvimento e aplicação sejam conduzidos de forma responsável [Kalota 2024].

2.4.1. Processamento de Linguagem Natural (PLN)

O Processamento de Linguagem Natural (PLN) é a área da computação que estuda como fazer os computadores entenderem e utilizarem a linguagem humana, tanto falada quanto

escrita. Para isso, combina conhecimentos de várias áreas, como ciência da computação, linguística e inteligência artificial. O mesmo é usado em tarefas como tradução automática, processamento e resumo de textos, reconhecimento de fala e etc.

Para entender a linguagem, os sistemas de PLN podem analisar diferentes níveis linguísticos:

- Fonético: como as palavras são pronunciadas;
- Morfológico: partes das palavras com significado (como prefixos e sufixos);
- Lexical: o significado das palavras;
- Sintático: a estrutura das frases;
- Semântico: o significado das frases;
- Discursivo: a estrutura de textos maiores;
- Pragmático: o conhecimento de mundo e contexto.

Os sistemas de PLN podem usar esses níveis para entender desde palavras isoladas até textos completos.Esses avanços são essenciais para o funcionamento de tecnologias atuais, como os LLMs, que dependem do PLN para gerar e interpretar textos de forma automática [Chowdhary 2020].

2.5. Arquitetura Transformer

A arquitetura Transformer é um tipo de modelo de rede neural amplamente utilizado em tarefas de processamento de linguagem natural. Ela segue uma estrutura de codificador e decodificador: o codificador transforma a entrada em uma representação interna contínua, enquanto o decodificador usa essa representação para gerar a saída, palavra por palavra. Tanto o codificador quanto o decodificador são compostos por camadas empilhadas que combinam mecanismos de autoatenção e redes neurais totalmente conectadas. Uma das inovações centrais do Transformer é o uso da atenção multi-cabeça, que permite ao modelo considerar diferentes partes do texto simultaneamente. Além disso, a arquitetura aplica conexões residuais e normalização em cada subcamada para manter a estabilidade do treinamento. Essa estrutura tornou-se a base para muitos modelos modernos de linguagem, como os LLMs, por permitir o processamento paralelo de sequências longas com alta eficiência [Vaswani et al. 2017].

2.5.1. Modelos de Linguagem de Grande Escala (LLMs)

Baseada na arquitetura transformer, Modelos de Linguagem de Grande Escala, ou LLMs (do inglês *Large Language Models*), são sistemas estatísticos capazes de prever sequências de palavras com base em padrões extraídos de grandes volumes de texto gerado por humanos. Ao receberem um fragmento textual, esses modelos estimam quais palavras provavelmente virão a seguir, segundo a distribuição estatística observada no corpus de treinamento. Diferentemente dos humanos, que aprendem a falar porque nascem cercado de pessoas que usam a linguagem e vivem no mesmo mundo que elas, aprendendo a se comunicar ao interagir com esse grupo e com o ambiente em comum, os LLMs não possuem intenção comunicativa, crenças ou conhecimento real. Sua "resposta correta" é aquela que estatisticamente mais se ajusta ao que aparece em textos públicos. Ainda assim, sua versatilidade permite aplicações em diálogos, geração de textos, resolução de problemas, tradução de idiomas, elaboração de resumos jornalísticos, criação de roteiros

e solução de enigmas lógicos, desde que sejam guiados por técnicas como prompt engineering. Apesar de sua semelhança superficial com interações humanas, é importante lembrar que LLMs apenas produzem sequências de palavras prováveis, sem compreender de fato o conteúdo gerado [Shanahan 2024].

Ao decorrer de 2023, Diversos LLMs foram desenvolvidos e lançados, alcançando ampla popularidade. Entre os mais notáveis estão o ChatGPT, da OpenAI, o LLaMA, da Meta AI, e o Dolly 2.0, da Databricks. O ChatGPT, por exemplo, já acumula mais de 180 milhões de usuários ao redor do mundo [Yao et al. 2024].

2.6. Prompt Engineering

Prompt engineering é a prática de criar e refinar comandos textuais para otimizar a interação com modelos de inteligência artificial. Essa técnica busca maximizar a precisão e relevância das respostas geradas pela IA, explorando um entendimento profundo de como o modelo funciona e é treinado. Através de ajustes nos prompts, como a segmentação de perguntas complexas e a estruturação cuidadosa das entradas, é possível guiar o modelo a fornecer respostas mais adequadas e detalhadas. À medida que a IA evolui, o domínio dessa competência se torna essencial para diversos profissionais que desejam explorar o potencial máximo dessas ferramentas tecnológicas [Lund 2023].

Existem diversas técnicas de *prompt engineering* que aprimoram a interação com modelos de linguagem, permitindo maior controle sobre os resultados gerados. Neste trabalho, foram aplicadas especificamente algumas dessas técnicas, conforme definidas por [Sahoo et al. 2024]:

Zero-Shot Prompting: Essa técnica permite que o modelo resolva uma tarefa sem necessidade de treinamento prévio ou exemplos. A IA utiliza apenas as instruções fornecidas no prompt, aplicando seu conhecimento pré-existente para gerar respostas adequadas. É especialmente útil para tarefas simples ou bem definidas.

Few-Shot Prompting: Diferentemente do zero-shot, essa técnica apresenta ao modelo alguns exemplos de entrada e saída relacionados à tarefa. Esses exemplos ajudam a IA a entender o padrão desejado e a melhorar seu desempenho, especialmente em tarefas mais complexas.

Chain-of-Thought Prompting (CoT): Essa técnica incentiva o modelo a realizar um raciocínio lógico passo a passo antes de gerar a resposta final. Usando comandos como ”Vamos pensar passo a passo”, o modelo pode estruturar sequências de pensamento que aumentam a precisão em problemas mais complexos.

Directional Stimulus Prompting: Direciona a IA a seguir um tom, estilo ou estrutura específicos com base em exemplos ou instruções fornecidas. Isso é útil para adaptar as respostas ao contexto desejado, como ao criar exercícios ou feedback personalizados.

Role Prompting: Define um contexto ou papel para a IA, como ”Você é um professor responsável por criar exercícios educativos”. Essa técnica ajuda o modelo a ajustar sua abordagem às expectativas da tarefa.

Essas técnicas demonstram a flexibilidade e a eficiência do prompt engineering ao criar interações otimizadas com modelos de IA, permitindo aplicações variadas e personalizadas. À medida que novas técnicas surgem, a prática continua evoluindo como uma

competência fundamental para maximizar os benefícios da inteligência artificial generativa.

2.7. Interface de Programação de Aplicações (API)

Uma API (Application Programming Interface) permite a comunicação programática entre dois ou mais componentes de software. O termo ”interface”indica que as APIs funcionam como uma ponte entre diferentes sistemas, estabelecendo limites e permitindo o acesso a recursos. Nesse contexto, os desenvolvedores de software desempenham os papéis tanto de consumidores quanto de criadores dessas APIs. A estabilidade das APIs é crucial, pois mudanças frequentes podem afetar o código dos consumidores, tornando-as contratos que devem ser bem pensados e definidos [De Souza et al. 2004]. A API possuem múltiplos papéis, sendo eles:

APIs como contratos: Estabelecem uma interface definida entre pelo menos duas partes, funcionando como contrato. Permitindo que equipes trabalhem de forma independente, e garantindo que o contrato acordado nas reuniões de revisão seja cumprido.

APIs como limites organizacionais: Definem os limites entre diferentes componentes de software, que são implementados por equipes distintas. Elas servem como fronteiras externas dos componentes e, consequentemente, definem as interfaces entre as equipes de desenvolvimento, fazendo com que os times interajam sem precisar conhecer detalhes de implementação.

APIs como mecanismo de comunicação: São um meio de comunicação entre as equipes, permitindo que elas discutam e coordenem seu trabalho. Por exemplo, as equipes de servidor e cliente coordenam suas tarefas por meio das APIs, com uma focada na criação de APIs e a outra no consumo e criação de interfaces.

Neste trabalho, a API é utilizada como contrato para disponibilizar os serviços de geração de exercícios e simulação de respostas. Essa estrutura padronizada facilita a integração dos modelos de linguagem com outros componentes da aplicação, promovendo reutilização e organização do código.

3. Trabalhos Relacionados

Liu et al. [Liu et al. 2025] investigaram o uso de grandes modelos de linguagem (LLMs) como substitutos ou complementos a respondentes humanos na calibração estatística de itens educacionais, por meio de um estudo comparativo envolvendo seis modelos (GPT-3.5, GPT-4, Llama 2, Llama 3, Gemini-Pro e Cohere Command R Plus). Utilizando questões de Álgebra Universitária e a Teoria da Resposta ao Item (TRI), os autores analisaram a capacidade dos LLMs em simular padrões de resposta humanos e avaliaram a correlação entre parâmetros estimados com dados sintéticos e reais. Modelos como o GPT-3.5 alcançaram alta correlação com as estimativas humanas (Spearman = 0,87), com melhoria via técnicas de reamostragem (até Spearman = 0,93). Contudo, os autores destacam que os LLMs apresentam uma distribuição de proficiência mais estreita, o que limita sua capacidade de representar a variabilidade humana. Em contraste, nosso trabalho amplia essa abordagem ao incluir a geração controlada de exercícios, simulação de respostas com perfis variados de habilidade e validação dos parâmetros por meio de testes estatísticos e métricas de erro.

Santos et al. [Santos et al. 2023] exploraram o uso do ChatGPT no ensino da Matemática por meio de testes com questões de livros didáticos do Ensino Médio, analisando sua capacidade de resolver problemas passo a passo e gerar materiais como listas e planos de aula. A análise foi qualitativa, focada nas respostas geradas e na utilidade percebida da ferramenta. Diferentemente dessa abordagem descritiva, nosso trabalho emprega a TRI para simular e avaliar respostas binárias de estudantes fictícios, aplicando métricas quantitativas como erro absoluto médio (MAE), erro quadrático médio (MSE) e intervalos de confiança dos parâmetros estimados.

Já o artigo feito por [Xiao et al. 2023] apresenta uma solução para a geração de materiais de leitura em inglês nas escolas chinesas, onde esse tipo de recurso é escasso e muitas vezes desatualizado. Os autores defendem que textos envolventes, aliados a exercícios adaptados ao nível de cada aluno, podem aumentar o interesse pela aprendizagem e acelerar a proficiência. Eles propõem um exercício de compreensão composto por uma passagem longa e coerente seguida de questões de múltipla escolha.

Como método base, utilizam o GPT-2 medium com fine-tuned em dois conjuntos de dados (leitura suplementar e exercícios de interpretação) e aplicam PPLM para controlar o tópico por meio de palavras-chave. Essa configuração gera passagens coerentes e alinhadas aos temas. Posteriormente, exploram o ChatGPT em modos zero-shot e one-shot, onde definem parâmetros como tamanho, gênero textual, nível de dificuldade e tópicos, ou fornecem um texto de referência para orientar a geração. Com o ChatGPT, obtêm ganhos expressivos em fluência, naturalidade e adequação ao público-alvo, superando o GPT-2 + PPLM em quase todas as métricas avaliadas. Além disso, o artigo descreve um sistema visual interativo que permite ao usuário configurar o nível CEFR, o tamanho das passagens, o gênero, o número de questões e o número de alternativas, possibilitando a criação automática de material de leitura personalizado para cada estudante. Esse trabalho apresenta abordagem similar, com o uso de LLM para geração de conteúdo educacional, mas diverge nos modelos utilizados, na ferramenta e no propósito, que seria mais direcionado para exercícios de compreensão em inglês.

Em linha semelhante, Baidoo et al. [Baidoo-Anu 2023] avaliaram os benefícios e limitações do ChatGPT na educação matemática por meio de análises empíricas sobre a geração de conteúdos educacionais. Embora tenham relatado bons resultados em tarefas básicas, observaram dificuldades da IA em resolver exercícios mais complexos e riscos relacionados ao uso passivo por estudantes. Apesar das contribuições práticas, o estudo não inclui validação estatísticas nem simulação de desempenho, como propomos neste trabalho.

Logacheva et al. [Logacheva et al. 2024] utilizaram o GPT-4 para gerar exercícios de programação personalizados com base em interesses dos estudantes. A avaliação, feita por especialistas e alunos, destacou o engajamento e a clareza das tarefas, embora apontasse divergências quanto à dificuldade percebida. Já o presente trabalho produziu um conjunto mais diversificado de questões, variando quantidade, nível de dificuldade, idioma e seguindo o estilo ENEM, e avaliou qualitativamente se a IA atendeu a esses requisitos, mas não incluiu personalização conforme interesses específicos dos usuários. Já na etapa de simulação de respostas, aplicamos quantitativamente a TRI por meio de simulações de acertos e erros e métricas estatísticas para estimar e validar os parâmetros.

O sistema KAQG proposto por [Chen and Shiu 2025], cria automaticamente questões de prova com dificuldade controlada usando grafos de conhecimento e geração aumentada por recuperação (RAG). Primeiro, o material didático é transformado em um grafo onde entidades e relações são bem estruturadas. Depois, o componente de RAG faz buscas nesse grafo para encontrar fatos relevantes e os envia para o modelo de linguagem. Com esses dados, o modelo gera perguntas ajustadas pelos critérios da Teoria da Resposta ao Item (dificuldade, discriminação e chance de chute) e pelos níveis de Bloom. Nos testes, o TRI de cada questão produzida correspondeu aos benchmarks de especialistas. Entre os ganhos estão o controle preciso dos parâmetros, a integração automática de conhecimento externo via RAG e a capacidade de gerar muitas perguntas de forma escalável. Nosso trabalho adota uma abordagem similar, combinando geração de questões e validação de sua dificuldade, mas difere por algumas diferentes técnicas de geração e validação.

Ogbonna e Opara [Ogbonna and Opara] aplicaram o modelo logístico de três parâmetros (ML-3P) para estimar erros padrão (EP) de itens de um teste real de matemática, utilizando máxima verossimilhança marginal. Propuseram o uso desses erros como critério de seleção de itens com melhor estabilidade. Nosso trabalho complementa essa abordagem ao aplicar o modelo ML-3P em dados sintéticos gerados via IA, incluindo etapas de simulação de respostas e análise estatística com bootstrap, MAE, MSE, MAPE e intervalos de confiança.

O estudo de Benedetto et al. [Benedetto et al. 2024] propôs o uso de *prompt engineering* para simular respostas de estudantes com diferentes níveis de proficiência, utilizando o GPT-3.5. A abordagem envolveu a criação de um *reference prompt* (RP) para gerar saídas padronizadas, que foram testadas em múltiplos conjuntos de dados e avaliadas por monotonicidade da acurácia e aderência à dificuldade das questões. Embora o método tenha se mostrado promissor, os autores observaram instabilidade nas estimativas de dificuldade e limitações na generalização para outros modelos. Nosso trabalho, por sua vez, aplica diretamente a TRI para calibrar os parâmetros a, b e c, realiza validações com reamostragem e analisa provas completas simuladas, oferecendo uma abordagem mais robusta e interpretável para a geração e validação de exercícios educacionais.

De modo geral, os trabalhos analisados evidenciam o potencial dos LLMs no apoio ao processo educacional, seja por meio da geração de conteúdo, da simulação de respostas ou da avaliação automatizada. Entretanto, grande parte dessas iniciativas concentra-se em análises predominantemente qualitativas ou em validações parciais. Conforme apresentado na Tabela 1, o presente estudo diferencia-se ao propor uma abordagem end-to-end, que integra a geração controlada de exercícios, a simulação de respostas a partir de perfis variados de habilidade e a validação estatística fundamentada na Teoria da Resposta ao Item, empregando métricas quantitativas para assegurar a consistência e a relevância dos itens produzidos.

4. Materiais e Métodos

Esta seção descreve os métodos, ferramentas e recursos utilizados no desenvolvimento de uma solução que combina técnicas de engenharia de prompts com o modelo de linguagem do GPT e do DeepSeek, com o objetivo de otimizar o processo de ensino e aprendizagem por meio da geração de conteúdos educacionais personalizados.

Tabela 1. Trabalhos Relacionados

Trabalho	Método Principal	Tipo de Questões	Validação
Este trabalho	LLMs + TRI	Ciências Humanas (ENEM 2023)	Métricas de erro; aplicação da TRI para estimar dificuldade; avaliação qualitativa das perguntas geradas quanto ao atendimento aos requisitos
Liu et al. (2025)	LLMs + TRI	Álgebra Universitária (múltipla escolha)	Correlação entre parâmetros simulados e reais (Spearman até 0,93); análise da distribuição de proficiências
Santos et al. (2023)	ChatGPT	Matemática (livros didáticos do Ensino Médio)	Qualitativa: análise das respostas geradas e percepção de utilidade
Xiao et al. (2023)	GPT-2 + PPLM; ChatGPT	Leitura e compreensão em inglês (múltipla escolha)	Avaliação de fluência, naturalidade e adequação ao público-alvo; comparação entre GPT-2 + PPLM e ChatGPT
Baidoo et al. (2023)	ChatGPT	Matemática (conteúdos educacionais)	Qualitativa: análise de benefícios e limitações; observação de dificuldades em exercícios complexos
Logacheva et al. (2024)	GPT-4	Programação (exercícios personalizados)	Avaliação qualitativa por especialistas e estudantes quanto a clareza, engajamento e dificuldade percebida
Chen et al. (2025)	RAG + LLM + TRI	Questões de prova com dificuldade controlada	Comparação com benchmarks de especialistas; avaliação de parâmetros de dificuldade, discriminação e acerto ao acaso
Ogbonna & Opara (2024)	TRI (3PL)	Matemática (itens reais de teste)	Estimativa de erros padrão via máxima verossimilhança marginal; seleção de itens mais estáveis
Benedetto et al. (2024)	Prompt Engineering + GPT-3.5	Questões de múltipla escolha (simulação de respostas)	Avaliação de monotonicidade da acurácia e aderência à dificuldade; observação de instabilidade e limitações de generalização

4.1. Visão Geral da Abordagem

A proposta deste trabalho consiste na criação de uma aplicação voltada para auxiliar educadores na produção de exercícios personalizados. A solução integra estratégias de engenharia de prompts com o modelo de linguagem o4-mini para a geração de conteúdo educacional. Esse modelo foi escolhido devido à familiaridade com sua utilização, ao seu baixo custo operacional e por apresentar desempenho otimizado, maior velocidade de resposta e menor custo em relação a versões anteriores, características que o tornam especialmente adequado para aplicações interativas baseadas em IA generativa [OpenAI 2024]. Além disso, o modelo DeepSeek-Chat foi utilizado para simular as respostas de estudantes fictícios com base em perfis de habilidade previamente definidos. A escolha desse modelo se deu por sua boa capacidade de raciocínio orientado por instruções e, igualmente, pelo custo acessível de operação, o que o torna uma alternativa viável para ambientes de validação automatizada de atividades educacionais.

Para operacionalizar essa integração, foi desenvolvida uma API utilizando o framework FastAPI, uma ferramenta moderna e de alto desempenho baseada em Python,

projetada para facilitar a criação de aplicações web assíncronas com suporte a tipagem estática [Ramírez 2019]. Essa API atua como intermediária entre os usuários e o modelo de IA, viabilizando a geração automatizada de conteúdos educacionais alinhados às necessidades específicas dos alunos.

O processo metodológico inicia-se com a definição dos prompts, responsáveis por orientar o modelo de IA na geração dos conteúdos educacionais. Na sequência, os resultados produzidos são avaliados com base em critérios previamente estabelecidos, com o objetivo de verificar sua conformidade com os parâmetros definidos. Caso sejam identificadas inconsistências ou desvios, os prompts são ajustados com o intuito de aprimorar o desempenho do modelo.

4.2. Funcionalidades Propostas

O sistema foi concebido com duas funcionalidades principais:

4.2.1. Simulação de Respostas

Retorna respostas simuladas(0 ou 1) para questões específicas. Com base em perfis de habilidade gerados segundo uma distribuição gaussiana no intervalo de –3 a 3, com 1000 amostras, a ferramenta produz as respostas de cada aluno fictício às respectivas questões.

Para este estudo, foram selecionadas 20 questões de Ciências Humanas do ENEM 2023 (caderno azul, códigos 47 a 68), previamente filtradas para excluir itens com imagens, em particular as questões 56 e 65. Essas questões foram enviadas à API, na qual a IA generativa simula as respostas de cada estudante segundo seu perfil de habilidade.

Para viabilizar a simulação de respostas para 1000 alunos em 20 questões, observou-se falhas de execução atribuídas ao limite de tokens por chamada da API. Como solução, a geração foi parcelada em 50 chamadas assíncronas, cada uma processando 20 perfis de habilidade, o que garantiu respostas completas sem erros de token e permitiu manter a performance de forma escalável.

Foram utilizados três prompts distintos para a simulação de respostas. No prompt A, as instruções são mais genéricas: cada questão recebe um nível de dificuldade b em escala de -3 a $+3$ e adota-se uma chance fixa de acerto por palpite de 20%, gerando respostas corretas (1) ou incorretas (0) sem um modelo probabilístico explícito.

Já o prompt B oferece um contexto muito mais específico, voltado para questões de Ciências Humanas e alinhado à BNCC. Nele, cada item é descrito pelos parâmetros originais da TRI (a de discriminação, b de dificuldade e c de chute) e a probabilidade de acerto do estudante é calculada pelo modelo logístico de três parâmetros (ML-3P). Além disso, o prompt define faixas de proficiência para o traço latente θ (por exemplo, $\theta > 2$ indica domínio avançado) e exige que as respostas sigam estritamente a mesma ordem das perguntas, pois é necessário para fazer os testes estatísticos posteriormente. Com isso, abandona-se o chute genérico de 20% em favor de um cálculo probabilístico fundamentado, esperando simulações mais fiéis ao comportamento esperado dos alunos.

O Prompt C calcula automaticamente a dificuldade de cada questão com base em regras explicitadas no próprio prompt e utiliza a função logística da TRI para gerar a probabilidade de acerto, já com a dificuldade ajustada. Para tornar a simulação mais realista,

inclui erros sistemáticos (40 % de chance de escolher um distrator plausível e 15 % de chance de escolher uma opção absurda) e viés cognitivo (preferência por alternativas centrais B–D). Por fim, exige que a saída seja um JSON contendo a lista de escolhas (A–E), a lista binária de acertos (0/1), o percentual de acerto de cada questão e a dificuldade calculada.

4.2.2. Geração de Exercícios

Permite criar automaticamente questões objetivas e dissertativas a partir de exemplos do ENEM de 2023 sobre um determinado tema. É possível definir o número de questões, a quantidade de alternativas, o tipo de questão e o nível de dificuldade.

Para este estudo, utilizaram-se 180 questões do ENEM de 2023, previamente filtradas para excluir aquelas que contêm imagens. As questões foram classificadas em três níveis de dificuldade: fácil, média e difícil, com base no parâmetro B dos microdados, garantindo segmentação equilibrada a partir dessa divisão:

- Fácil, se $b < -1,0$;
- Média, se $-1,0 \leq b \leq 1,0$;
- Difícil, se $b > 1,0$.

Com base nessa classificação, a IA generativa recebe a lista de três questões de exemplo de acordo com o nível de dificuldade selecionado e gera uma nova lista de exercícios sobre o tema escolhido, seguindo as especificações do prompt.

4.3. Desenvolvimento da API

A API foi um dos pilares da solução desenvolvida. Com ela, foi possível a comunicação os modelos de IA. O framework escolhido foi o *FastAPI*, por sua agilidade no desenvolvimento e compatibilidade com aplicações modernas.

4.3.1. Estrutura da API

A organização do código seguiu boas práticas de desenvolvimento, dividindo os arquivos em camadas específicas:

Controllers: Responsáveis por receber as requisições e repassá-las aos serviços correspondentes.

Models: Define a estrutura dos dados trocados entre o sistema e a IA. Cada funcionalidade possui modelos distintos para *prompts*, *requests* e *responses*.

Services: Onde ocorre a lógica principal. Aqui, os dados recebidos são processados e enviados ao modelo GPT-4o, que retorna os conteúdos gerados.

Routes: Define os endpoints disponíveis na API, mapeando as funcionalidades e permitindo a comunicação com os usuários.

4.3.2. Execução da API

Para execução da aplicação em ambiente local, utilizou-se o servidor Uvicorn, uma implementação ASGI leve e de alto desempenho [Christie 2023]. Sua integração com o FastAPI facilita o desenvolvimento de aplicações reativas e com recarga automática durante o processo de desenvolvimento.

```
uvicorn main:app --reload
```

A flag `--reload` permite que o servidor atualize automaticamente em caso de alterações no código.

4.3.3. Funcionamento da API

A API recebe requisições HTTP do tipo POST. A partir disso, o serviço correspondente interpreta os dados, aciona o modelo de IA generativa e retorna a resposta em formato JSON (JavaScript Object Notation), um padrão leve e amplamente utilizado para a troca de informações entre sistemas distintos [JSON.org 2024]. Essa estrutura modular facilita tanto a adição de novas funcionalidades quanto a manutenção do sistema.

A API indica o resultado de cada solicitação por meio de códigos HTTP padronizados, permitindo ao cliente interpretar rapidamente se a operação foi bem-sucedida ou se ocorreu algum tipo de erro. A seguir, apresentam-se os principais códigos de resposta utilizados:

Códigos de resposta HTTP

- 200 OK– Sucesso.
- 422 Unprocessable Entity – Requisição bem formada, porém com semântica inválida (falha de validação de dados).
- 500 Internal Server Error – Erro interno do servidor durante o processamento.

4.3.4. Rotas Criadas

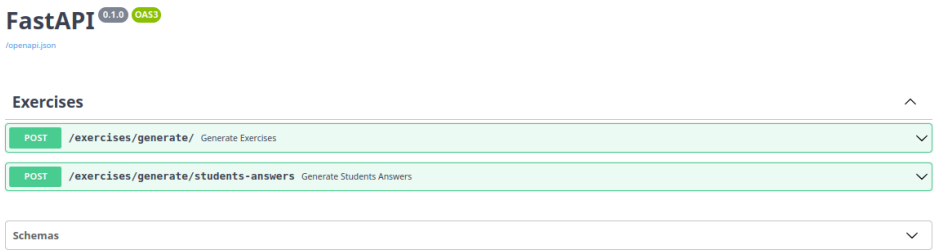

Figura 2. Interface Swagger exibida na execução local da API desenvolvida com FastAPI. Fonte: Elaborado pela autora, 2025.

Criação de Exercícios Personalizados

Método: `POST`
Rota: `/exercises/generate/`

Entrada esperada Recebe o tópico selecionado e os parâmetros para geração das questões, como idioma, nível de dificuldade e número de questões.

Listing 1. Entrada para a geração de exercícios

```
{
  "topic": {
    "name": "string",
    "number_of_questions": 0,
    "multiple_choice_qty": 0,
    "multiple_choice_options": 0,
    "open_ended_qty": 0
  },
  "idiom": "string",
  "difficulty": "string"
}
```

Saída esperada Recebe uma lista de questões contendo o tópico selecionado, o enunciado, o tipo de questão, a resposta correta e as alternativas.

Listing 2. Saída para a geração de exercícios

```
{
  "topic": "string",
  "questions": [
    {
      "type": "string",
      "question": "string",
      "options": [
        "string"
      ],
      "answer": "string"
    }
  ]
}
```

Simulacão de Respostas

Método: `POST`
Rota: `/exercises/generate/students-answers`

Entrada esperada Para o primeiro prompt, recebe-se a lista de exercícios contendo o título da questão, as alternativas e o ID para realização da simulação.

Listing 3. Entrada para a simulação de respostas do primeiro prompt

```
{
  "questions": [
    {
      "id": "string",
```

```
      "question": "string",
      "options": [
        "string"
      ]
    }
  ]
}
```

Já para o segundo prompt, acrescenta-se apenas o objeto dos parâmetros TRI, para que a IA calcule a probabilidade de acerto.

Listing 4. Entrada para a simulação de respostas do segundo prompt

```
{
  "questions": [
    {
      "id": "string",
      "question": "string",
      "options": [
        "string"
      ],
      "parameters": {
        "parameter_a": "string",
        "parameter_b": "string",
        "parameter_c": "string"
      }
    }
  ]
}
```

Saída esperada Retorna a lista de alunos com as respostas simuladas, a habilidade de cada um e o ID de cada questão.

Listing 5. Saída da simulação de respostas

```
{
  "responses": [
    {
      "student_id": "string",
      "ability": 0,
      "answers": [
        0
      ]
    }
  ]
}
```

4.3.5. Integração com IA generativa

A comunicação com os serviços de IA é feita por meio da biblioteca `openai` em Python, conforme mostrado na Listagem 6.

Listing 6. Exemplo Genérico de Integração com IA utilizado no projeto

```python
def call_service(request: Request):
    completion = openai.chat.completions.create(
        model="o4-mini",
        response_format={ "type": "json_object" },
        messages=[
            {"role": "system", "content": get_prompt(request)},
            {"role": "user", "content": str(request)}
        ]
    )
    response = json.loads(completion.choices[0].message.content)
    return response
```

Configuração de Credenciais Para utilizar os serviços de IA, foi necessário adquirir chaves de API da OpenAI e da DeepSeek. Estas credenciais foram armazenadas em um arquivo `.env` com as variáveis `OPENAI_API_KEY` e `DEEPSEEK_API_KEY`.

4.4. Desenvolvimento dos Prompts

Os prompts são parte essencial para orientar a geração de conteúdo pela IA. Quanto mais bem elaborados, mais precisos e coerentes tendem a ser os resultados, uma vez que a formulação adequada das instruções influencia diretamente na qualidade da resposta.

Para estruturar a interação com o4-mini para a geração de exercícios educacionais, utilizou-se o prompt completo listado no Apêndice A (Listing 7). Já para a simulação de respostas pelos alunos, foram utilizados os prompts listados no Apêndice B (Listing 10)

4.4.1. Estrutura dos Prompts

A estrutura dos prompts foi organizada em três arquivos principais:

examples: Define os formatos esperados de entrada e saída, com exemplos práticos que auxiliam a IA a compreender o tipo de resposta desejada. Esses exemplos funcionam como modelo para orientar a geração.

instructions: Reúne as instruções que guiam a IA na produção dos conteúdos. Elas incluem regras específicas, como o número de itens, o nível de dificuldade e outras informações relevantes para a tarefa.

prompt: É o arquivo que consolida os exemplos e as instruções, permitindo que a IA gere a resposta final no formato desejado, como JSON, por exemplo.

4.4.2. Técnicas de Prompting Utilizadas

Para este trabalho, utilizou-se um conjunto de técnicas de engenharia de prompts para orientar a LLM e aprimorar os resultados. O conceito de cada técnica foi apresentado na seção de referencial teórico.

Few-shot Prompting: Incluiu-se exemplos de questões fáceis, médias e difíceis do ENEM no prompt de geração de exercícios, permitindo que o modelo assimile o estilo, a complexidade e o padrão de alternativas.

Directional Stimulus Prompting: Utilizou-se esta técnica para direcionar o tom, o vocabulário e o formato de saída, mantendo o estilo JSON e evitando desvios em relação aos exemplos fornecidos.

Contextual Prompting: Foi utilizada para definir o papel da IA (por exemplo, "You are a specialist teacher creating educational exercises for high school students"), melhorando a compreensão da tarefa e favorecendo respostas mais alinhadas aos objetivos.

Chain-of-Thought Prompting: Utilizou-se esta técnica para apresentar uma sequência de passos obrigatórios (leitura dos exemplos, identificação da estrutura lógica, estilo e padrão de alternativas e inclusão de contexto histórico), guiando o raciocínio do modelo antes da geração.

Zero-shot Prompting: Não foram fornecidos exemplos prévios. No primeiro raprompt de simulação de respostas, solicitou-se apenas: "Simule a resposta de um aluno para a questão...", sem exemplos para guiar.

4.5. Validação Estatística dos Parâmetros TRI

Para avaliar a capacidade do simulador de estimar os parâmetros a, b e c, foram geradas estimativas com base no modelo logístico de três parâmetros (ML-3P). As estimativas $\hat{a}_i$, $\hat{b}_i$ e $\hat{c}_i$, obtidas pela média dos limites inferior e superior do intervalo de confiança de cada questão, foram comparadas aos valores oficiais a_i, b_i e c_i dos microdados do ENEM 2023.

Com base nos erros obtidos entre os valores estimados e os reais, foram calculadas cinco métricas estatísticas sobre o conjunto de 20 questões de Ciências Humanas: erro médio de viés (MBE), erro absoluto médio (MAE), erro quadrático médio (MSE), raiz do erro quadrático médio (RMSE) e erro percentual médio (MAPE). As fórmulas dessas métricas encontram-se descritas na seção de referencial teórico.

4.5.1. Intervalo de Confiança de 95%

Para estimar a variabilidade dos parâmetros gerados pelo modelo TRI, foi empregada a técnica de reamostragem *bootstrap*, com 1000 amostras geradas com reposição a partir do conjunto original de respostas dos estudantes. Para cada amostra, os parâmetros a, b e c foram reestimados para todas as questões, totalizando 1000 repetições do processo.

Com base nas distribuições obtidas para cada parâmetro, foram calculados os intervalos de confiança de 95% utilizando a distribuição t de Student, por meio do ambiente *Google Colab*, com a linguagem Python. A função empregada foi:

```
stats.t.interval(0.95, len(dados) - 1,
```

```
loc=np.mean(dados),
scale=stats.sem(dados))
```

da biblioteca `scipy`. Nessa função, os argumentos representam, respectivamente: o nível de confiança, os graus de liberdade, a média amostral e o erro padrão da média.

O nível de confiança de 95% indica que existe 95% de probabilidade de que o valor verdadeiro do parâmetro esteja contido entre os limites inferior e superior do intervalo, correspondentes aos quantis de 2,5% e 97,5% da distribuição t.

Para obtenção das estimativas pontuais finais, cada parâmetro (a, b e c) de cada questão foi definido como o valor médio entre o limite inferior e o limite superior de seu respectivo intervalo de confiança. Dessa forma, além de avaliar a variabilidade das estimativas, obtém-se uma estimativa central que reflete a posição média da distribuição de bootstrap e que será utilizada na comparação com os parâmetros do ENEM de 2023.

4.6. Classificação dos Níveis de Dificuldade

Para facilitar a interpretação das estimativas do parâmetro de dificuldade b, adotamos duas convenções de limiar, ambas dividindo o domínio $[-3, +3]$ em três categorias qualitativas, mas com larguras diferentes:

- Critério estreito:
  - Fácil, se $b < -0,5$;
  - Média, se $-0,5 \leq b \leq 0,5$;
  - Difícil, se $b > 0,5$.
- Critério amplo:
  - Fácil, se $b < -1,0$;
  - Média, se $-1,0 \leq b \leq 1,0$;
  - Difícil, se $b > 1,0$.

Essa rotulagem permite agrupar qualitativamente os itens em classes de dificuldade intuitivas, mantendo a base estatística do TRI. O critério estreito destaca pequenas variações ao redor de zero, enquanto o critério amplo assegura que cada categoria cubra exatamente duas unidades de habilidade, oferecendo uma visão de sensibilidade distinta para análise.

5. Resultados

5.1. Simulação de respostas

Nesta seção são apresentadas as análises quantitativas para a simulação de respostas, nas quais os parâmetros estimados pelo modelo ML-3P e bootstrap são comparados aos valores reais do ENEM 2023. As métricas calculadas incluem o erro simples, o erro absoluto, o erro quadrático médio, raiz do erro quadrático médio, o erro percentual médio, o intervalo de confiança e a simulação da TRI para cada questão.

5.1.1. Análise Comparativa das Métricas para o Parâmetro a

No caso do parâmetro de discriminação a observa-se que os Prompts A e B Tabelas 2 e 3 apresentam viés de superestimação, enquanto o Prompt C Tabela 4 revela viés de

subestimação sem que nenhuma estimativa caia dentro do intervalo de confiança de 95%. As amplitudes dos IC95 variam de $0{,}02759$ a $0{,}23804$ no Prompt A de $0{,}03555$ a $0{,}15047$ no Prompt B e de $0{,}00009$ a $0{,}36936$ no Prompt C refletindo previsões mais estáveis no Prompt A , dispersão intermediária no Prompt B e picos de incerteza mais acentuados no Prompt C. O Prompt A também se destaca pela menor média de erro com $1{,}35377$.

Tabela 2. Prompt A: Métricas de Validação por Questão para o Parâmetro a

Questão	a_{orig}	A [IC95%]	MAE	MBE	MSE	MAPE
1	2,40144	5,95796 [5,92027; 5,99564]	3,55652	-3,55652	12,64880	148,09927
2	3,06915	2,06486 [2,04047; 2,08925]	1,00429	1,00429	1,00860	32,72209
3	2,88883	3,89588 [3,86001; 3,93174]	1,00705	-1,00705	1,01414	34,85996
4	3,29734	1,95357 [1,91382; 1,99331]	1,34378	1,34378	1,80573	40,75330
5	0,71667	2,69464 [2,67817; 2,71111]	1,97797	-1,97797	3,91237	275,99453
6	1,58000	2,13557 [2,11952; 2,15161]	0,55557	-0,55557	0,30865	35,16234
7	2,78931	4,54716 [4,50647; 4,58785]	1,75785	-1,75785	3,09004	63,02096
8	1,22125	3,26875 [3,14973; 3,38777]	2,04750	-2,04750	4,19226	167,65609
9	2,43245	4,66081 [4,60547; 4,71614]	2,22836	-2,22836	4,96557	91,60949
10	2,53325	2,86230 [2,83244; 2,89215]	0,32905	-0,32905	0,10827	12,98905
11	2,86400	2,46898 [2,45518; 2,48277]	0,39503	0,39503	0,15604	13,79277
12	1,70103	2,33371 [2,30116; 2,36625]	0,63268	-0,63268	0,40028	37,19364
13	1,40431	3,84971 [3,82022; 3,87920]	2,44540	-2,44540	5,97998	174,13534
14	2,59177	1,92071 [1,86109; 1,98033]	0,67106	0,67106	0,45032	25,89196
15	2,58516	4,49901 [4,45057; 4,54744]	1,91385	-1,91385	3,66280	74,03197
16	3,81637	2,28985 [2,24880; 2,33090]	1,52652	1,52652	2,33026	39,99927
17	1,20452	2,77406 [2,75587; 2,79224]	1,56954	-1,56954	2,46344	130,30377
18	2,42214	2,58718 [2,55475; 2,61960]	0,16504	-0,16504	0,02724	6,81360
19	1,53812	3,40268 [3,37815; 3,42720]	1,86456	-1,86456	3,47657	121,22299
20	2,59313	2,67692 [2,59063; 2,76321]	0,08379	-0,08379	0,00702	3,23123
Média			1,35377	-0,85970	2,60042	76,47418

5.1.2. Análise Comparativa das Métricas para o Parâmetro b

Em todos os três prompts (Tabelas 5, 6 e 7) verifica-se viés nas estimativas de b, porém com sinais opostos: os Prompts A e C apresentaram MBE positivo ($0{,}14371$ e $0{,}53978$), indicando subestimação, enquanto o Prompt B teve MBE negativo ($-0{,}16207$), sinalizando superestimação. Em termos de MAE, o Prompt B foi o mais preciso (MAE = $0{,}46825$), seguido pelo Prompt A (MAE = $0{,}51358$) e pelo Prompt C (MAE = $0{,}97717$). As amplitudes dos IC95% também diferem: no Prompt A variam de $0{,}00335$ a $0{,}01957$; no Prompt B, de $0{,}00372$ a $0{,}00824$; e no Prompt C, de $0{,}00000$ a $0{,}05049$. Apenas na questão 17 do Prompt A o valor de b ficou contido no respectivo intervalo de confiança. Quanto a MSE e MAPE, o Prompt A obteve $0{,}45558$ e $457{,}56189\%$, o Prompt B $0{,}32987$ e $844{,}68140\%$, e o Prompt C $1{,}32141$ e $457{,}36980\%$. Esses resultados indicam que o Prompt B sofre menor penalização quadrática média em comparação ao Prompt A, enquanto o Prompt C exibe picos de incerteza mais acentuados.

Tabela 3. Prompt B: Métricas de Validação por Questão para o Parâmetro a

Questão	a_{orig}	A [IC95%]	MAE	MBE	MSE	MAPE
1	2,40144	4,44971 [4,40627; 4,49315]	2,04827	-2,04827	4,19541	85,29341
2	3,06915	6,44763 [6,37239; 6,52286]	3,37848	-3,37848	11,41409	110,07852
3	2,88883	4,62503 [4,58818; 4,66187]	1,73620	-1,73620	3,01437	60,10028
4	3,29734	5,19883 [5,13110; 5,26655]	1,90149	-1,90149	3,61565	57,66724
5	0,71667	2,72746 [2,70937; 2,74554]	2,01079	-2,01079	4,04326	280,57335
6	1,58000	3,65212 [3,62759; 3,67664]	2,07212	-2,07212	4,29366	131,14652
7	2,78931	4,03818 [4,00875; 4,06760]	1,24887	-1,24887	1,55966	44,77326
8	1,22125	2,76264 [2,74486; 2,78041]	1,54139	-1,54139	2,37587	126,21372
9	2,43245	4,38529 [4,35301; 4,41757]	1,95284	-1,95284	3,81358	80,28284
10	2,53325	3,75953 [3,73387; 3,78518]	1,22628	-1,22628	1,50375	48,40718
11	2,86400	3,98664 [3,95722; 4,01605]	1,12264	-1,12264	1,26031	39,19815
12	1,70103	4,00365 [3,97384; 4,03345]	2,30262	-2,30262	5,30204	135,36593
13	1,40431	3,77373 [3,74774; 3,79972]	2,36942	-2,36942	5,61415	168,72485
14	2,59177	4,48757 [4,45192; 4,52322]	1,89580	-1,89580	3,59406	73,14692
15	2,58516	3,46416 [3,43915; 3,48916]	0,87900	-0,87900	0,77263	34,00157
16	3,81637	3,76894 [3,74320; 3,79468]	0,04743	0,04743	0,00225	1,24280
17	1,20452	4,21695 [4,18524; 4,24865]	3,01243	-3,01243	9,07470	250,09340
18	2,42214	3,94859 [3,92009; 3,97709]	1,52645	-1,52645	2,33005	63,02072
19	1,53812	4,12854 [4,09723; 4,15984]	2,59042	-2,59042	6,71025	168,41436
20	2,59313	3,87652 [3,84846; 3,90458]	1,28339	-1,28339	1,64709	49,49193
Média			1,80731	-1,80257	3,80684	100,36185

Tabela 4. Prompt C: Métricas de Validação por Questão para o Parâmetro a

Questão	a_{orig}	A [IC95%]	MAE	MBE	MSE	MAPE
1	2,40144	3,08184 [3,04168; 3,12200]	0,68040	-0,68040	0,46294	28,33300
2	3,06915	0,78043 [0,77180; 0,78906]	2,28872	2,28872	5,23824	74,57179
3	2,88883	2,00016 [1,97293; 2,02738]	0,88868	0,88868	0,78974	30,76245
4	3,29734	1,94225 [1,91999; 1,96450]	1,35510	1,35510	1,83628	41,09661
5	0,71667	9,99934 [9,99929; 9,99938]	9,28267	-9,28267	86,16787	1295,24956
6	1,58000	1,74283 [1,72331; 1,76235]	0,16283	-0,16283	0,02651	10,30570
7	2,78931	1,63571 [1,61771; 1,65371]	1,15360	1,15360	1,33079	41,35790
8	1,22125	0,78658 [0,77660; 0,79656]	0,43467	0,43467	0,18894	35,59222
9	2,43245	1,03128 [1,01776; 1,04479]	1,40118	1,40118	1,96329	57,60345
10	2,53325	1,84954 [1,82567; 1,87341]	0,68371	0,68371	0,46746	26,98944
11	2,86400	1,51134 [1,49404; 1,52863]	1,35267	1,35267	1,82970	47,22992
12	1,70103	1,41229 [1,39281; 1,43177]	0,28874	0,28874	0,08337	16,97442
13	1,40431	1,36918 [1,35423; 1,38413]	0,03513	0,03513	0,00123	2,50158
14	2,59177	0,48667 [0,48206; 0,49127]	2,10511	2,10511	4,43147	81,22268
15	2,58516	1,87867 [1,69399; 2,06335]	0,70649	0,70649	0,49913	27,32868
16	3,81637	0,81184 [0,80604; 0,81763]	3,00454	3,00454	9,02723	78,72756
17	1,20452	0,71396 [0,70619; 0,72173]	0,49056	0,49056	0,24065	40,72660
18	2,42214	1,23745 [1,22545; 1,24944]	1,18470	1,18470	1,40350	48,91109
19	1,53812	0,99514 [0,97889; 1,01139]	0,54298	0,54298	0,29483	35,30154
20	2,59313	0,75553 [0,73554; 0,77552]	1,83760	1,83760	3,37677	70,86417
		Média	1,49400	0,48141	5,98300	104,58252

Tabela 5. Prompt A: Métricas de Validação por Questão para o Parâmetro b

Questão	b_{orig}	B [IC95%]	MAE	MBE	MSE	MAPE
1	0,00606	0,23827 [0,23659; 0,23994]	0,23221	-0,23221	0,05392	3831,76568
2	0,94872	0,40304 [0,39850; 0,40757]	0,54569	0,54569	0,29777	57,51802
3	0,67105	0,31509 [0,31271; 0,31747]	0,35596	0,35596	0,12671	53,04523
4	1,19128	0,38325 [0,37346; 0,39303]	0,80804	0,80804	0,65292	67,82914
5	1,29861	0,34582 [0,34299; 0,34865]	0,95279	0,95279	0,90781	73,36999
6	1,18538	0,28151 [0,27682; 0,28620]	0,90387	0,90387	0,81698	76,25150
7	0,82081	0,31208 [0,30991; 0,31424]	0,50874	0,50874	0,25881	61,97963
8	2,63782	0,76020 [0,75330; 0,76709]	1,87763	1,87763	3,52548	71,18094
9	0,67281	0,26399 [0,26182; 0,26616]	0,40882	0,40882	0,16713	60,76307
10	0,06643	0,49659 [0,49251; 0,50067]	0,43016	-0,43016	0,18504	647,53876
11	0,24521	0,27691 [0,27388; 0,27994]	0,03170	-0,03170	0,00100	12,92769
12	0,11476	0,55328 [0,54736; 0,55920]	0,43852	-0,43852	0,19230	382,11921
13	0,08591	0,29025 [0,28801; 0,29249]	0,20434	-0,20434	0,04175	237,85357
14	-0,02033	0,46701 [0,45954; 0,47448]	0,48734	-0,48734	0,23750	2397,14707
15	0,39346	0,32079 [0,31862; 0,32295]	0,07268	0,07268	0,00528	18,47075
16	0,70936	0,56787 [0,56215; 0,57359]	0,14149	0,14149	0,02002	19,94615
17	0,31781	0,31526 [0,31236; 0,31816]	0,00255	0,00255	0,00001	0,80237
18	-0,62019	0,46717 [0,46223; 0,47211]	1,08736	-1,08736	1,18235	175,32692
19	0,12887	0,26152 [0,25900; 0,26403]	0,13265	-0,13265	0,01759	102,92931
20	0,08088	0,72992 [0,72341; 0,73643]	0,64904	-0,64904	0,42125	802,47280
Média			0,51358	0,14425	0,45558	457,56189

Tabela 6. Prompt B: Métricas de Validação por Questão para o Parâmetro b

Questão	b_{orig}	B [IC95%]	MAE	MBE	MSE	MAPE
1	0,00606	0,57722 [0,57525; 0,57918]	0,57116	-0,57116	0,32622	9425,00000
2	0,94872	0,76078 [0,75892; 0,76264]	0,18794	0,18794	0,03532	19,80985
3	0,67105	0,52896 [0,52707; 0,53085]	0,14209	0,14209	0,02019	21,17428
4	1,19128	0,79278 [0,79070; 0,79485]	0,39851	0,39851	0,15881	33,45183
5	1,29861	1,25975 [1,25627; 1,26323]	0,03886	0,03886	0,00151	2,99243
6	1,18538	0,64853 [0,64618; 0,65088]	0,53685	0,53685	0,28821	45,28927
7	0,82081	0,54106 [0,53894; 0,54318]	0,27975	0,27975	0,07826	34,08219
8	2,63782	1,40542 [1,40130; 1,40954]	1,23240	1,23240	1,51881	46,72040
9	0,67281	0,44227 [0,44034; 0,44420]	0,23054	0,23054	0,05315	34,26525
10	0,06643	0,70184 [0,69955; 0,70413]	0,63541	-0,63541	0,40375	956,51061
11	0,24521	0,57964 [0,57751; 0,58177]	0,33443	-0,33443	0,11184	136,38514
12	0,11476	0,70366 [0,70141; 0,70591]	0,58890	-0,58890	0,34680	513,15789
13	0,08591	0,59247 [0,59043; 0,59450]	0,50656	-0,50656	0,25660	589,63450
14	-0,02033	0,69800 [0,69578; 0,70022]	0,71833	-0,71833	0,51600	3533,34973
15	0,39346	0,66146 [0,65903; 0,66388]	0,26800	-0,26800	0,07182	68,11239
16	0,70936	0,67896 [0,67663; 0,68129]	0,03040	0,03040	0,00092	4,28555
17	0,31781	0,58056 [0,57862; 0,58249]	0,26275	-0,26275	0,06903	82,67361
18	-0,62019	0,69676 [0,69453; 0,69899]	1,31695	-1,31695	1,73436	212,34622
19	0,12887	0,57927 [0,57729; 0,58124]	0,45040	-0,45040	0,20286	349,49562
20	0,08088	0,71570 [0,71332; 0,71808]	0,63482	-0,63482	0,40300	784,89120
Média			0,46825	-0,16052	0,32987	844,68140

Tabela 7. Prompt C: Métricas de Validação por Questão para o Parâmetro b

Questão	b_{orig}	B [IC95%]	MAE	MBE	MSE	MAPE
1	0,00606	-0,17462 [-0,18075; -0,16848]	0,18068	0,18068	0,03264	2981,43564
2	0,94872	-0,02021 [-0,03193; -0,00849]	0,96893	0,96893	0,93883	102,13024
3	0,67105	-0,39113 [-0,40352; -0,37873]	1,06218	1,06218	1,12822	158,28552
4	1,19128	-0,32451 [-0,33457; -0,31444]	1,51579	1,51579	2,29760	127,24003
5	1,29861	2,99997 [2,99997; 2,99997]	1,70136	-1,70136	2,89463	131,01393
6	1,18538	-0,50983 [-0,52391; -0,49575]	1,69521	1,69521	2,87374	143,00984
7	0,82081	-0,58652 [-0,60194; -0,57110]	1,40733	1,40733	1,98058	171,45624
8	2,63782	1,68930 [1,67631; 1,70228]	0,94853	0,94853	0,89970	35,95867
9	0,67281	-0,69013 [-0,71537; -0,66488]	1,36294	1,36294	1,85759	202,57353
10	0,06643	-0,41886 [-0,43215; -0,40557]	0,48529	0,48529	0,23551	730,52838
11	0,24521	-0,50607 [-0,52343; -0,48871]	0,75128	0,75128	0,56442	306,38228
12	0,11476	-0,41257 [-0,42969; -0,39544]	0,52733	0,52733	0,27807	459,50244
13	0,08591	-0,82904 [-0,84996; -0,80812]	0,91495	0,91495	0,83713	1065,00989
14	-0,02033	0,10090 [0,09045; 0,11134]	0,12123	-0,12123	0,01470	596,28628
15	0,39346	2,94474 [2,93349; 2,95599]	2,55128	-2,55128	6,50903	648,42169
16	0,70936	-0,47671 [-0,48268; -0,47073]	1,18607	1,18607	1,40675	167,20213
17	0,31781	-0,49380 [-0,50854; -0,47905]	0,81161	0,81161	0,65870	255,37428
18	-0,62019	-0,96311 [-0,98604; -0,94017]	0,34292	0,34292	0,11759	55,29193
19	0,12887	-0,81936 [-0,84259; -0,79612]	0,94823	0,94823	0,89913	735,79964
20	0,08088	0,02063 [-0,00452; 0,04578]	0,06025	0,06025	0,00363	74,49308
			0,97717	0,53978	1,32141	457,36980

5.1.3. Análise Comparativa das Métricas para o Parâmetro c

Em todos os três prompts (Tabelas 8, 9 e 10) observa-se viés positivo nas estimativas de c, indicando subestimação em relação aos valores originais, porém com intensidades distintas: mais acentuado no Prompt B (MBE = 0,16569), intermediário no Prompt A (MBE = 0,09630) e mais ameno no Prompt C (MBE = 0,01089). Em termos de MAE, o Prompt A foi o mais preciso (MAE = 0,10653), seguido pelo Prompt C (MAE = 0,11703) e, por fim, pelo Prompt B (MAE = 0,16569).

As amplitudes dos IC95% também diferem: no Prompt A variam de $0{,}00035$ a $0{,}00848$; no Prompt B, de $0{,}00000$ a $0{,}00101$; e no Prompt C, de $0{,}00017$ a $0{,}01779$. Apenas a questão 20 do Prompt A e a questão 13 do Prompt C tiveram seu valor original de c dentro do respectivo intervalo de confiança; nos demais casos, todos os IC95% excluíram o valor real de c.

Quanto a MSE e MAPE, o Prompt A obteve $0{,}01569$ e $201{,}39484\%$, o Prompt B $0{,}03182$ e $98{,}81164\%$, e o Prompt C $0{,}01763$ e $377{,}33159\%$. Esses resultados indicam que o Prompt A apresenta a menor penalização quadrática média, enquanto o Prompt C sofre com picos de incerteza mais elevados.

5.1.4. Classificação de itens por grau de dificuldade

Na Tabela 11, utilizando o critério de limiares de $-0{,}5$ e $0{,}5$ para definir as categorias "fácil", "média" e "difícil", comparamos a classificação qualitativa de dificuldade original de cada item (fácil/média/difícil) com as estimativas geradas pelos Prompts A, B e C. Os

Tabela 8. Prompt A: Métricas de Validação por Questão para o Parâmetro c

Questão	$\mathbf{c}_{\text{orig}}$	C [IC95%]	MAE	MBE	MSE	MAPE
1	0,21265	0,00121 [0,00103; 0,00138]	0,21145	0,21145	0,04471	99,43334
2	0,17725	0,02143 [0,01992; 0,02293]	0,15583	0,15583	0,02428	87,91255
3	0,18912	0,05545 [0,05438; 0,05651]	0,13368	0,13368	0,01787	70,68264
4	0,15604	0,13400 [0,12976; 0,13824]	0,02204	0,02204	0,00049	14,12458
5	0,03594	0,01176 [0,01096; 0,01256]	0,02418	0,02418	0,00058	67,27880
6	0,13863	0,03619 [0,03449; 0,03788]	0,10245	0,10245	0,01049	73,89815
7	0,25733	0,05879 [0,05780; 0,05977]	0,19855	0,19855	0,03942	77,15579
8	0,30445	0,19008 [0,18665; 0,19350]	0,11438	0,11438	0,01308	37,56775
9	0,00278	0,08387 [0,08269; 0,08505]	0,08109	-0,08109	0,00658	2916,90647
10	0,17011	0,06391 [0,06230; 0,06551]	0,10621	0,10621	0,01128	62,43313
11	0,19053	0,01756 [0,01657; 0,01855]	0,17297	0,17297	0,02992	90,78360
12	0,11185	0,13276 [0,13027; 0,13525]	0,02091	-0,02091	0,00044	18,69468
13	0,21032	0,02664 [0,02581; 0,02747]	0,18368	0,18368	0,03374	87,33359
14	0,17587	0,06991 [0,06683; 0,07298]	0,10597	0,10597	0,01123	60,25189
15	0,17772	0,08197 [0,08085; 0,08308]	0,09576	0,09576	0,00917	53,87970
16	0,15459	0,11027 [0,10776; 0,11277]	0,04433	0,04433	0,00196	28,67262
17	0,22687	0,04634 [0,04514; 0,04753]	0,18054	0,18054	0,03259	79,57641
18	0,09595	0,08022 [0,07819; 0,08225]	0,01573	0,01573	0,00025	16,39396
19	0,18969	0,02903 [0,02808; 0,02997]	0,16067	0,16067	0,02581	84,69872
20	0,14645	0,14677 [0,14376; 0,14978]	0,00032	-0,00032	0,00000	0,21850
Média			0,10653	0,09630	0,01569	201,39484

Tabela 9. Prompt B: Métricas de Validação por Questão para o Parâmetro c

Questão	$\mathbf{c}_{\text{orig}}$	C [IC95%]	MAE	MBE	MSE	MAPE
1	0,21265	0,00461 [0,00423; 0,00498]	0,20805	0,20805	0,04328	97,83447
2	0,17725	0,00010 [0,00006; 0,00013]	0,17716	0,17716	0,03138	99,94640
3	0,18912	0,00001 [0,00000; 0,00001]	0,18912	0,18912	0,03576	99,99736
4	0,15604	0,01366 [0,01315; 0,01416]	0,14239	0,14239	0,02027	91,24904
5	0,03594	0,00153 [0,00133; 0,00172]	0,03442	0,03442	0,00118	95,75682
6	0,13863	0,00000 [0,00000; 0,00000]	0,13863	0,13863	0,01922	100,00000
7	0,25733	0,00000 [0,00000; 0,00000]	0,25733	0,25733	0,06622	100,00000
8	0,30445	0,00006 [0,00003; 0,00009]	0,30439	0,30439	0,09265	99,98029
9	0,00278	0,00024 [0,00016; 0,00031]	0,00255	0,00255	0,00001	91,54676
10	0,17011	0,00001 [0,00000; 0,00002]	0,17010	-0,17010	0,02893	99,99412
11	0,19053	0,00000 [0,00000; 0,00000]	0,19053	-0,19053	0,03630	100,00000
12	0,11185	0,00001 [0,00000; 0,00001]	0,11185	-0,11185	0,01251	99,99553
13	0,21032	0,00005 [0,00001; 0,00008]	0,21028	-0,21028	0,04422	99,97860
14	0,17587	0,00000 [0,00000; 0,00000]	0,17587	-0,17587	0,03093	100,00000
15	0,17772	0,00001 [0,00000; 0,00001]	0,17772	-0,17772	0,03158	99,99719
16	0,15459	0,00000 [0,00000; 0,00000]	0,15459	-0,15459	0,02390	100,00000
17	0,22687	0,00005 [0,00002; 0,00008]	0,22682	-0,22682	0,05145	99,97796
18	0,09595	0,00001 [0,00000; 0,00001]	0,09595	-0,09595	0,00921	99,99479
19	0,18969	0,00003 [0,00001; 0,00004]	0,18967	-0,18967	0,03597	99,98682
20	0,14645	0,00001 [0,00000; 0,00001]	0,14645	-0,14645	0,02145	99,99659
Média			0,16519	0,16519	0,03182	98,81164

Tabela 10. Prompt C: Métricas de Validação por Questão para o Parâmetro c

Questão	c_{orig}	C [IC95%]	MAE	MBE	MSE	MAPE
1	0,21265	0,29072 [0,28771; 0,29372]	0,07807	-0,07807	0,00609	36,71056
2	0,17725	0,03429 [0,03073; 0,03784]	0,14297	0,14297	0,02044	80,65726
3	0,18912	0,30358 [0,29867; 0,30848]	0,11446	-0,11446	0,01310	60,51978
4	0,15604	0,32158 [0,31784; 0,32532]	0,16554	-0,16554	0,02740	106,08818
5	0,03594	0,00198 [0,00189; 0,00206]	0,03397	0,03397	0,00115	94,50473
6	0,13863	0,27212 [0,26630; 0,27794]	0,13349	-0,13349	0,01782	96,29229
7	0,25733	0,26617 [0,25976; 0,27258]	0,00884	-0,00884	0,00008	3,43528
8	0,30445	0,03620 [0,03385; 0,03855]	0,26825	0,26825	0,07196	88,10971
9	0,00278	0,17523 [0,16674; 0,18372]	0,17245	-0,17245	0,02974	6203,23741
10	0,17011	0,28580 [0,28044; 0,29116]	0,11569	-0,11569	0,01338	68,00894
11	0,19053	0,25451 [0,24774; 0,26127]	0,06398	-0,06398	0,00409	33,57739
12	0,11185	0,19468 [0,18786; 0,20149]	0,08283	-0,08283	0,00686	74,05007
13	0,21032	0,21213 [0,20368; 0,22058]	0,00181	-0,00181	0,00000	0,86059
14	0,17587	0,00354 [0,00217; 0,00490]	0,17234	0,17234	0,02970	97,98999
15	0,17772	0,06005 [0,05519; 0,06490]	0,11768	0,11768	0,01385	66,21371
16	0,15459	0,00239 [0,00149; 0,00329]	0,15220	0,15220	0,02316	98,45398
17	0,22687	0,03204 [0,02806; 0,03602]	0,19483	0,19483	0,03796	85,87737
18	0,09595	0,22020 [0,21130; 0,22909]	0,12425	-0,12425	0,01544	129,48932
19	0,18969	0,11300 [0,10511; 0,12088]	0,07670	0,07670	0,00588	40,43176
20	0,14645	0,02618 [0,02168; 0,03068]	0,12027	0,12027	0,01446	82,12359
Média			0,11703	0,01089	0,01763	377,33159

resultados mostram:

1. Prompt A: acertou em 10 de 20 questões (50%), apresentando boa concordância nos itens originalmente médios (10, 11, 13, 14, 15, 17 e 19), mas falhou em 7 dos 8 itens originalmente difíceis (2, 3, 4, 5, 6, 7 e 9) e em três médios (12, 18 e 20), indicando que tende a subestimar sistematicamente a dificuldade.
2. Prompt B: obteve 8 acertos (40%), mostrando-se mais consistente na identificação dos itens originalmente difíceis (acertou 8 dos 9, questões 2–8 e 16), porém errou a maioria dos itens médios (1, 10–15, 17, 19 e 20).
3. Prompt C: classificou corretamente 9 questões (45%), equilibrando acertos entre itens difíceis (corretos: 2, 5 e 8; errados: 3, 4, 6, 7, 9 e 16) e itens médios (corretos: 1, 12, 14, 17 e 20; errados: 10, 11, 13, 15, 18 e 19), mas ainda apresenta falhas tanto na superestimação de fáceis/médios quanto na subestimação de difíceis.

Na Tabela 12, utilizando o critério de limiares de $-1{,}0$ e $1{,}0$ para definir as categorias "fácil", "média" e "difícil", comparamos a classificação qualitativa de dificuldade original de cada item (fácil/média/difícil) com as estimativas geradas pelos Prompts A, B e C. Essa comparação entre diferentes convenções de limiar permite uma análise de sensibilidade, evidenciando como a amplitude da zona "média" impacta diretamente na distribuição de rótulos e ajuda a justificar a escolha do intervalo mais adequado ao conjunto de itens. Os resultados mostram:

1. Prompt A: acertou a categoria em 16 de 20 questões (80%), apresentando maior estabilidade, mas subestimou quatro itens originalmente difíceis (4, 5, 6 e 8).
2. Prompt B: obteve 18 acertos (90%), vencendo em consistência ao rotular corretamente todos os itens com dificuldade média e dois dos três originalmente difíceis (5 e 8), errando apenas nas questões 4 e 6.

Tabela 11. Classificação qualitativa da dificuldade original e estimada pelos Prompts A, B e C

Questão	b_{orig}	Prompt A	Prompt B	Prompt C
1	Média	Média	Difícil	Média
2	Difícil	Média	Difícil	Média
3	Difícil	Média	Difícil	Média
4	Difícil	Média	Difícil	Média
5	Difícil	Média	Difícil	Difícil
6	Difícil	Média	Difícil	Fácil
7	Difícil	Média	Difícil	Fácil
8	Difícil	Difícil	Difícil	Difícil
9	Difícil	Média	Média	Fácil
10	Média	Média	Difícil	Média
11	Média	Média	Difícil	Fácil
12	Média	Difícil	Difícil	Média
13	Média	Média	Difícil	Fácil
14	Média	Média	Difícil	Média
15	Média	Média	Difícil	Difícil
16	Difícil	Difícil	Difícil	Média
17	Média	Média	Difícil	Média
18	Fácil	Média	Difícil	Fácil
19	Média	Média	Difícil	Fácil
20	Média	Difícil	Difícil	Média

3. Prompt C: classificou corretamente 17 questões (85%), falhando em três originalmente difíceis (4, 6) e em um com dificuldade média (15).

Comparar os critérios de limiar (±0,5 e ±1,0) foi essencial para avaliar a estabilidade das estimativas de dificuldade dos três prompts. No critério mais estreito (–0,5 a 0,5), os Prompts A e B acertaram cerca de 50% das classificações e mantiveram as estimativas muito próximas das categorias originais, sem trocas drásticas (por exemplo, itens fáceis classificados como difíceis ou vice-versa). Ao adotar o critério mais amplo (–1,0 a 1,0), ambos passaram a ter acertos acima de 80%, melhorando o desempenho quantitativo com coerência. Já o Prompt C, embora tenha alcançado até 85% de acertos com a zona "média" ampliada, exibiu variabilidade excessiva, rotulando itens originalmente difíceis como fáceis em alguns casos no critério mais estreito.

Esses resultados indicam que testar diferentes limiares ajuda a escolher o critério mais apropriado e faz com que as classificações reflitam melhor o nível real das questões. Os Prompts A e B mostraram-se mais estáveis: ao aumentar o limite em 0,5, eles passaram a acertar uma parcela considerável das categorias e, mesmo nos erros, mantiveram estimativas próximas da faixa correta. Ainda assim, todos os prompts exigem ajustes para aproximar ainda mais as previsões dos valores originais, com atenção especial ao Prompt C, que tende a gerar variações excessivas na classificação.

Tabela 12. Classificação qualitativa da dificuldade original e estimada pelos Prompts A, B e C

Questão	b_{orig}	Prompt A	Prompt B	Prompt C
1	Média	Média	Média	Média
2	Média	Média	Média	Média
3	Média	Média	Média	Média
4	Difícil	Média	Média	Média
5	Difícil	Média	Difícil	Difícil
6	Difícil	Média	Média	Média
7	Média	Média	Média	Média
8	Difícil	Média	Difícil	Difícil
9	Média	Média	Média	Média
10	Média	Média	Média	Média
11	Média	Média	Média	Média
12	Média	Média	Média	Média
13	Média	Média	Média	Média
14	Média	Média	Média	Média
15	Média	Média	Média	Difícil
16	Média	Média	Média	Média
17	Média	Média	Média	Média
18	Média	Média	Média	Média
19	Média	Média	Média	Média
20	Média	Média	Média	Média

5.2. Geração de exercícios

A geração de conteúdos educacionais personalizados foi realizada por meio da aplicação das estratégias de prompting descritas na seção anterior. O sistema desenvolvido permitiu que o usuário configurasse a criação de exercícios com base em tópico, quantidade de questões, nível de dificuldade e tipo de pergunta (objetiva ou dissertativa). Esta seção apresenta exemplos representativos dos exercícios gerados, organizados por nível de dificuldade, com o objetivo de ilustrar a eficácia qualitativa da abordagem adotada.

5.3. Exemplos de Exercícios Gerados

Questões fáceis

- **Tema: Era Vargas**
  Entre 1937 e 1945, no período do Estado Novo, o governo de Getúlio Vargas estabeleceu o Departamento de Imprensa e Propaganda (DIP), que passou a centralizar a divulgação de notícias e produzir materiais de comunicação oficial. Essa estratégia tinha como objetivo principal:
  A) Uniformizar a opinião pública em torno do projeto político varguista.
  B) Garantir a plena liberdade de imprensa e o pluralismo de ideias.
  C) Descentralizar a difusão cultural para as federações estaduais.
  D) Incentivar a formação de partidos de oposição ao governo.
  E) Fortalecer a autonomia dos legislativos regionais.

- **Tema: Fisiologia Humana**
  Durante os Jogos Olímpicos do Rio de Janeiro de 2016, as altas temperaturas e a umidade elevados

levaram à intensificação do suor entre os atletas. Esse panorama pode sobrecarregar os mecanismos de termorregulação, sobretudo ao depender do processo de perda de calor mais sensível a variações de umidade do ar. Disponível em: https://rio2016.com. Acesso em: 15 ago. 2016 (adaptado). Com base nos princípios da fisiologia humana, o mecanismo de termorregulação que apresenta maior dependência da umidade relativa do ambiente é:
A) Transferência de calor por convecção.
B) Emissão de calor por radiação infravermelha.
C) Perda de calor por evaporação do suor.
D) Condução de calor para superfícies metálicas.
E) Troca de calor por ventilação pulmonar.

Questões Médias

- **Tema: Era Vargas**
  Durante o Estado Novo (1937–1945), o governo de Getúlio Vargas promoveu uma política de substituição de importações com o objetivo de reduzir a dependência do Brasil em relação a bens manufaturados estrangeiros. A implantação da Companhia Siderúrgica Nacional em 1941 foi um marco desse processo. Relacionando esse contexto à estratégia econômica da Era Vargas, constata-se que a política de substituição de importações foi marcada pela:
  A) Expansão da indústria de base para fortalecimento da autonomia econômica.
  B) Desvalorização da moeda para estimular a exportação de produtos agrícolas.
  C) Incentivo à importação de matérias-primas em detrimento ao consumo interno.
  D) Abertura irrestrita ao capital estrangeiro na composição do parque industrial.
  E) Prioridade no desenvolvimento de infraestrutura de transporte em países vizinhos.

- **Tema: Fisiologia Humana**
  Com o envelhecimento populacional e o aumento das doenças cardiovasculares crônicas, cresce a importância de compreendermos os mecanismos de autorregulação da pressão arterial. No sistema nervoso, os barorreceptores do seio carotídeo e do arco aórtico detectam variações da pressão. Quando a pressão arterial eleva-se abruptamente, ocorre aumento do estiramento desses receptores, desencadeando aumento do disparo de potenciais de ação ao tronco encefálico e ativação da via parassimpática cardíaca em detrimento da via simpática. Com base no texto, a ativação do sistema parassimpático cardíaco ocasiona, em curto prazo, a
  A) Liberação de renina pelo aparelho justaglomerular.
  B) Vasoconstrição arteriolar periférica.
  C) Elevação da contratilidade e do débito cardíaco.
  D) Diminuição da frequência cardíaca.
  E) Aumento da secreção de adrenalina pela medula adrenal.

Questões difíceis

- **Tema: Era Vargas**
  No contexto da crise dos anos 1930 e da ascensão de Getúlio Vargas ao poder, o Estado Novo (1937-1945) centralizou o controle político e instituiu políticas trabalhistas corporativistas, destacando-se a criação do Ministério do Trabalho e a promulgação da Consolidação das Leis do Trabalho (CLT) em 1943, que oficializou direitos como férias remuneradas e jornada de 8 horas diárias, ao mesmo tempo em que cooptava o movimento sindical para impedir greves independentes. A adoção da CLT e da legislação trabalhista no período do Estado Novo teve como principal objetivo:
  A) Fortalecer a autonomia dos sindicatos e promover paralisações laborais frequentes.
  B) Unificar a legislação social e integrar o trabalhador à lógica do Estado, evitando greves independentes.
  C) Elevar a carga horária de trabalho industrial para enfrentar a crise econômica.

D) Descentralizar a regulamentação trabalhista e transferir a competência para os estados.
E) Promover a liberalização do mercado de trabalho para atrair investimentos estrangeiros.

- **Tema: Fisiologia Humana**
  Em meados do século XXI, o avanço das tecnologias de telemetria permitiu o monitoramento em tempo real da frequência cardíaca de atletas amadores durante corridas de rua. Estudos populacionais identificaram que, em corridas de longa duração, ocorrem modificações fisiológicas que visam manter a homeostasia cardiovascular e a perfusão tecidual. Nesse cenário, durante exercícios prolongados, verifica-se aumento do débito cardíaco e redistribuição do fluxo sanguíneo para os músculos atuantes. Esse fenômeno ocorre principalmente devido ao(a):
  A) Aumento do retorno venoso promovido pela bomba muscular.
  B) Vasoconstrição local nos músculos esqueléticos.
  C) Redução da resistência vascular pulmonar.
  D) Inibição do centro vasomotor no bulbo.
  E) Diminuição do volume sistólico pela taquicardia.

5.3.1. Questões geradas com idioma alemão

- **Tema: Fisiologia Humana**
  In den letzten Jahrzehnten hat die Prävalenz von Typ-2-Diabetes mellitus weltweit deutlich zugenommen, vor allem in ländlichen Gebieten mit begrenztem Zugang zu medizinischer Versorgung. Laut WHO stieg die Zahl der Betroffenen von etwa 108 Millionen im Jahr 1980 auf über 422 Millionen im Jahr 2014. Das Hormon Insulin spielt eine zentrale Rolle bei der Regulation des Blutzuckerspiegels. Welche der folgenden Wirkungen wird primär durch Insulin vermittelt?
  A) Hemmung der hepatischen Glukoneogenese.
  B) Stimulierung der Lipolyse im Fettgewebe.
  C) Förderung der Glykogenolyse in der Muskulatur.
  D) Erhöhung der glomerulären Filtrationsrate.
  E) Verstärkung der Reninfreisetzung in der Niere.

5.4. Considerações sobre as questões geradas

Os exemplos apresentados indicam que o sistema produz exercícios claros e adequados ao nível médio, cobrindo temas como Fisiologia humana e Era Vargas, e seguindo o estilo do ENEM com contexto introdutório e alternativas bem estruturadas. O mecanismo atende corretamente às configurações de tipo de questão, quantidade, idioma e nível de dificuldade, além de mostrar estabilidade em execuções sucessivas (conjuntos de exercícios com estrutura semelhante). Como pontos a aprimorar, foram observadas repetições de trechos em alguns enunciados e variações na profundidade do contexto histórico ou social, o que pode exigir revisão humana para aumentar a relevância. As técnicas de prompting, em especial a combinação de exemplos e instruções direcionadas, revelaram-se eficazes na produção de questões contextualizadas, apontando caminhos claros para ajustes futuros.

5.5. Validação de Nível de Dificuldade com Prompt A

Tabela 13. Estimativas de b (IC95%) e classificação de dificuldade

Questão	Dificuldade escolhida	Dificuldade estimada	B [IC95%]
1	FÁCIL	MÉDIA	$-0{,}12466$ [$-0{,}12635$; $-0{,}12297$]
2	FÁCIL	MÉDIA	0,13770 [0,13420; 0,14120]
3	MÉDIA	MÉDIA	0,38341 [0,38140; 0,38541]
4	MÉDIA	MÉDIA	$-0{,}00004$ [$-0{,}00333$; 0,00326]
5	DIFÍCIL	DIFÍCIL	0,65483 [0,65290; 0,65675]
6	DIFÍCIL	MÉDIA	0,35681 [0,35480; 0,35881]

Optou-se pelo Prompt A para gerar estas estimativas porque ele apresentou a segunda menor média de erro absoluto entre os três métodos, perdendo apenas para o Prompt B. Contudo, o Prompt B não pôde ser utilizado, pois exige o valor original de TRI de cada questão, informação indisponível em questões geradas artificialmente. Assim, o Prompt A mostrou-se a alternativa viável.

Na Tabela 13, que utilizou as questões da Seção 5.3 e o critério de dificuldade baseado nos limiares $-0{,}5$ e $0{,}5$, observa-se que, de seis itens avaliados, três foram classificados corretamente (questões 1, 3 e 5). Nos demais casos (2, 4 e 6), embora a classificação tenha diferido da original, os valores estimados permaneceram próximos aos limites de cada categoria. Não houve erro de ordem extrema, por exemplo, nenhuma questão originalmente fácil foi classificada como difícil ou vice-versa, o que indica que, mesmo nas "falhas", as estimativas de a mantiveram coerência com a faixa de dificuldade esperada.

6. Conclusões

Neste trabalho foi desenvolvida uma API em FastAPI que combina engenharia de prompts e LLMs para duas tarefas principais: gerar exercícios no estilo ENEM e simular respostas de estudantes, estimando parâmetros da TRI (a, b e c) para validar se cada questão atinge o nível de dificuldade desejado. Qualitativamente, as questões produzidas respeitaram idioma, número de itens, formato e temas definidos. Para verificar quantitativamente essa correspondência, realizamos um teste adicional utilizando o Prompt A, escolhido por apresentar baixo erro absoluto médio, para classificar seis itens gerados (Tabela 13). Com o resultado do Prompt A também foram gerados os intervalos de confiança para cada parâmetro, permitindo extrair o valor de b, responsável pela dificuldade, e compará-lo aos limiares $-0{,}5$ e $0{,}5$. Dos seis itens, três foram corretamente atribuídos às categorias "fácil", "média" e "difícil"; nos demais, embora tenham sido assinalados em faixa vizinha, não ocorreram inversões extremas. Esse resultado sugere que o Prompt A pode servir de base para aferir níveis de dificuldade, mas ainda demanda refinamento para garantir maior precisão em todas as faixas.

Na análise dos parâmetros TRI, observou-se que a foi superestimado nos Prompts A e B e subestimado no Prompt C, que b foi subestimado nos Prompts A e C, mas superestimado no Prompt B; e que c ficou sistematicamente subestimado em todos os cenários. Em termos de erro absoluto médio, o Prompt A foi o mais preciso para a (MAE = 1,35377) e para c (MAE = 0,10653), enquanto o Prompt B foi o mais preciso para b (MAE = 0,46825, com o menor MSE e RMSE relativo). Embora os erros absolutos não tenham

sido excessivos, o MAPE mostrou-se pouco confiável quando os denominadores se aproximam de zero, motivando a escolha de MAE e MSE como métricas principais. Esses resultados indicam que, apesar dos avanços proporcionados pelos prompts, ainda persiste variabilidade aleatória que não reflete perfeitamente a habilidade simulada do aluno.

Em termos de classificação qualitativa de dificuldade, realizamos dois testes com intervalos de limiares diferentes: estreito (–0,5 a 0,5) e amplo (–1,0 a 1,0). No primeiro teste, o Prompt C apresentou variabilidade excessiva, chegando a rotular itens originalmente difíceis como fáceis. Já os Prompts A e B atingiram 50% e 40% de acerto, respectivamente, mas mantiveram coerência entre as faixas mesmo quando erravam. No segundo teste, com limiar mais amplo, o Prompt B destacou-se como o mais consistente, pois 90% de suas estimativas de b coincidiram com as categorias originais. Ele também obteve MAE de 0,46825 em b, equivalente a 7,80% de erro em uma escala de –3 a +3. Assim, é pouco provável que um desvio de até 0,5 unidades desloque um item para fora de sua categoria original, somente se o valor estiver localizado nos extremos da faixa. Ainda assim, as métricas quantitativas indicam espaço para aprimoramento: reduzir a variabilidade aleatória nas estimativas e aproximar cada vez mais os valores simulados dos originais, com objetivo de garantir resultados mais confiáveis.

Este trabalho forneceu resultados iniciais promissores ao integrar LLMs via engenharia de prompts para geração e simulação de exercícios. No entanto, há amplo espaço para refinamento: a aplicação de técnicas de prompt tuning e a adoção de estratégias de RAG (Retrieval-Augmented Generation) podem aprimorar significativamente tanto a criação de exercícios quanto a simulação de respostas. Além disso, ao simular respostas de estudantes amostrando habilidades por distribuição gaussiana e reestimando-as via bootstrap, foi possível calcular intervalos de confiança que revelam a variabilidade das estimativas e permitem comparar diferentes cenários de forma mais confiável. Com base nisso, propomos as seguintes sugestões para estudos futuros:

- Aplicar técnicas de *prompt tuning* e incorporar RAG (*Retrieval-Augmented Generation*) para enriquecer o contexto e reduzir vieses nas estimativas.
- Experimentar LLMs maiores ou *fine-tuning* para aprimorar a estimação dos parâmetros da TRI, especialmente b.
- Realizar estudos com grupos de alunos e professores reais, ajustando a simulação de respostas com base em dados empíricos.
- aplicar o método em outras áreas do conhecimento, avaliando a flexibilidade dos prompts em diferentes disciplinas.

Referências

Araujo, E. A. C. d., Andrade, D. F. d., and Bortolotti, S. L. V. (2009). Teoria da resposta ao item. *Revista da Escola de Enfermagem da USP*, 43:1000–1008.

Baidoo-Anu, David e Ansah, L. O. (2023). Education in the era of generative artificial intelligence (ai): Understanding the potential benefits of chatgpt in promoting teaching and learning. *Journal of AI*, 7(1):52–62.

Baker, F. B. (2001). *The basics of item response theory*. ERIC.

Benedetto, L., Aradelli, G., Donvito, A., Lucchetti, A., Cappelli, A., and Buttery, P. (2024). Using llms to simulate students' responses to exam questions. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 11351–11368.

Chen, C. H. and Shiu, M. F. (2025). Kaqg: A knowledge-graph-enhanced rag for difficulty-controlled question generation. *arXiv preprint arXiv:2505.07618*.

Chicco, D., Warrens, M. J., and Jurman, G. (2021). The coefficient of determination r-squared is more informative than smape, mae, mape, mse and rmse in regression analysis evaluation. *Peerj computer science*, 7:e623.

Chowdhary, K. (2020). Natural language processing. *Fundamentals of artificial intelligence*, pages 603–649.

Christie, T. (2023). Uvicorn: An asgi web server implementation for python. Acessado em: abril de 2025.

De Souza, C. R., Redmiles, D., Cheng, L.-T., Millen, D., and Patterson, J. (2004). Sometimes you need to see through walls: a field study of application programming interfaces. In *Proceedings of the 2004 ACM conference on Computer supported cooperative work*, pages 63–71.

Edelen, M. O. and Reeve, B. B. (2007). Applying item response theory (irt) modeling to questionnaire development, evaluation, and refinement. *Quality of life research*, 16(Suppl 1):5–18.

Hesterberg, T. (2011). Bootstrap. *Wiley Interdisciplinary Reviews: Computational Statistics*, 3(6):497–526.

JSON.org (2024). Introducing json (javascript object notation). Acessado em: abril de 2025.

Kalota, F. (2024). A primer on generative artificial intelligence. *Education Sciences*, 14(2):172.

Liu, Y., Bhandari, S., and Pardos, Z. A. (2025). Leveraging llm respondents for item evaluation: A psychometric analysis. *British Journal of Educational Technology*, 56(3):1028–1052.

Logacheva, E., Hellas, A., Prather, J., Sarsa, S., and Leinonen, J. (2024). Evaluating contextually personalized programming exercises created with generative ai. In *Proceedings of the 2024 ACM Conference on International Computing Education Research-Volume 1*, pages 95–113.

Lund, B. (2023). The prompt engineering librarian. *Library Hi Tech News*, 40(8):6–8.

O'Brien, S. F. and Yi, Q. L. (2016). How do i interpret a confidence interval? *Transfusion*, 56(7):1680–1683.

Ogbonna, J. U. and Opara, I. M. Estimating standard errors of irtparameters of mathematics achievement test using three parameter model.

OpenAI (2024). Gpt-4o technical report. Acessado em: abril de 2025.

Pasquali, L. and Primi, R. (2003). Fundamentos da teoria da resposta ao item: Tri. *Avaliação Psicológica: Interamerican Journal of Psychological Assessment*, 2(2):99–110.

Piotrowski, P., Rutyna, I., Baczyński, D., and Kopyt, M. (2022). Evaluation metrics for wind power forecasts: A comprehensive review and statistical analysis of errors. *Energies*, 15(24):9657.

Ramírez, S. (2019). Fastapi: Modern, fast (high-performance), web framework for building apis with python 3.6+ based on standard python type hints. Acessado em: abril de 2025.

Romão, G. S. and Sá, M. F. S. d. (2019). Como elaborar questões de múltipla escolha de boa qualidade. *Femina*, 47(9):561–4.

Sahoo, P., Singh, A. K., Saha, S., Jain, V., Mondal, S., and Chadha, A. (2024). A systematic survey of prompt engineering in large language models: Techniques and applications. *arXiv preprint arXiv:2402.07927*.

Santos, R. P., de Camargo Sant'Ana, C., and Sant'Ana, I. P. (2023). O chatgpt como recurso de apoio no ensino da matemática. *Revemop*, 5:e202303–e202303.

Shanahan, M. (2024). Talking about large language models. *Communications of the ACM*, 67(2):68–79.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., Polosukhin, I., et al. (2017). Attention is all you need. *Advances in neural information processing systems*, 30(1):2.

Vujović, Ž. et al. (2021). Classification model evaluation metrics. *International Journal of Advanced Computer Science and Applications*, 12(6):599–606.

Xiao, C., Xu, S. X., Zhang, K., Wang, Y., and Xia, L. (2023). Evaluating reading comprehension exercises generated by llms: A showcase of chatgpt in education applications. In *Proceedings of the 18th workshop on innovative use of NLP for building educational applications (BEA 2023)*, pages 610–625.

Yao, Y., Duan, J., Xu, K., Cai, Y., Sun, Z., and Zhang, Y. (2024). A survey on large language model (llm) security and privacy: The good, the bad, and the ugly. *High-Confidence Computing*, 4(2):100211.

A. Prompt criado para a geração de exercícios

Listing 7. Prompt para Geração de exercícios

```
You are a specialist teacher creating educational exercises for
   high school students, strictly using the provided examples to
    generate similar new questions.

## Mandatory Steps for Question Generation:

1. **Carefully read the provided example questions below.**
2. For each new question generated:
  - **Identify the logical structure of the example questions**
     (statement + complement + instruction).
  - **Identify the style** (type of vocabulary, tone used, text
     length).
  - **Clearly identify the pattern** of multiple-choice options
     (style of alternatives, complexity level(or the necessary
     hability), writing style, expression form).
```

```
  - **Include a introductory text containing historical, social,
      economic, or cultural context. This context must be
     meaningful and require interpretation, similar to the style
      of ENEM assessments.**
3. Generate new questions rigorously respecting these patterns.

## Essential Restrictions to Follow:

- **Do not deviate from the examples**. Each new question must
   look like it could have been directly extracted from the
   provided examples.
- Do not add extra explanations or details not aligned with the
   provided examples.
- Multiple-choice alternatives must have a similar style, format
   , length, and clarity to those provided.
- Do not number the questions.

## Reference Examples (follow exactly this format and style):
{examples}

## Generation Specifications:

- Question topic: `{request.topic.name}`
- Number of multiple-choice questions: `{request.topic.
   multiple_choice_qty}` (with exactly `{request.topic.
   multiple_choice_options}` options, clearly identifying one
   correct alternative).
- Number of open-ended questions: `{request.topic.open_ended_qty
   }`
- Mandatory output format: {output_format}
- Question language: `{request.idiom}`
- Output must be returned in JSON format.

**ATTENTION:**
The main goal of this prompt is to ensure maximum consistency
   between the generated questions and the provided examples.
   Any deviation will be considered a significant error and
   could negatively affect students.
```

Apêndice B – Prompts utilizados para simular a resposta dos alunos

Listing 8. Primeiro prompt para Simulação de Respostas

```
# Context
  You are an expert in educational assessment. Your task is to
     simulate each student's responses
  to ENEM-style multiple-choice questions based on their ability
      level (θ).
```

```
  # Input Parameters
  - List of students with ability levels:
  {student_list}

  # Simulation Rules
  1. **Question Difficulty (b):**
    Assign each question a difficulty rating on a scale from −3
       (very easy) to +3 (very hard).
  2. **Student Response:**
    For each student _i_, simulate their answer based on ability
        (θ)i and the questions difficulty.
  3. **Guessing Behavior:**
    If the student appears not to know the answer, assume a 20%
       chance of guessing correctly.
  4. **Recording Results:**
    For each question, output:
    - '1' = correct
    - '0' = incorrect

  # Output Format
  Return **only** a JSON object conforming to this schema:
  {output_format}

  > **Do not** include any additional textoutput strictly the
     JSON.
```

Listing 9. Segundo prompt para Simulação de Respostas

```
  # Context
  You are a psychometrics and educational assessment expert
     simulating student responses to ENADE-style questions in
     the area of Ciencias Humanas. These questions are designed
     to assess knowledge and skills consolidated in Brazilian
     high school education, aligned with the BNCC (Base Nacional
      Comum Curricular), focusing on:

- Interpretation and critical analysis of historical, geographic
   , political, philosophical, and sociological contexts.
- Reading comprehension, textual interpretation, and argument
   evaluation.
- Use of conceptual and procedural knowledge in real-world and
   sociohistorical problem situations.

INPUT PARAMETERS
----------------
- student_list: A list of students, each with a latent trait
   score (ability level) θi, typically ranging from 3 to +3:
   θ ≤ −2       : insufficient mastery of high school content
   −2 < θ ≤ 0   : partial mastery
```

```
   0 < θ ≤ +2    : proficient
   θ > +2        : advanced understanding and interpretation

{student_list}

- question_bank: A list of questions in Human Sciences, each
   defined by:
   a             : discrimination parameter
   b             : difficulty (3 = easy; +3 = very hard)
   c             : guessing parameter (e.g., 0.20 for 5-option
      MCQs)

{questions}

SIMULATION MODEL (IRT - 3PL)
----------------------------
For each studentquestion pair:
- Compute the probability of correct response using the 3-
   parameter logistic model:

   P(θi) = c + (1 - c) / (1 + exp(-a * (θi - b)))

- Simulate the response:
   - 1 (correct) with probability P()
   - 0 (incorrect) otherwise

Use a pseudo-random generator to sample the outcome based on the
    probability.
The list of student answers must strictly follow the same order
   as the input questions to ensure proper alignment between
   each response and its corresponding question.

OUTPUT FORMAT
-------------
Return a JSON object with the following structure:

{output_format}

- Each student must answer {len(questions)} questions.
- Do NOT include any explanatory text or commentary.
- Output ONLY the JSON object, and nothing else.
```

Listing 10. Terceir prompt para Simulação de Respostas

```
QUESTION SIMULATION MODEL (3rd YEAR HIGH SCHOOL)

1. ABILITY SCALE:
  - Range: -3.0 (severe deficit) to 3.0 (excellence)
  - Reference points:
```

```
       -3.0: 1525% expected correct
       0.0: 4555% expected correct
       3.0: 8090% expected correct

2. DIFFICULTY FACTORS (calculated automatically):
  a) Text:
       +0.1 per each 50 words beyond 100
       +0.2 for historical/technical vocabulary
       +0.3 for contextualization requirement
  b) Options:
       +0.15 for each additional plausible distractor
       +0.1 for similar terms among options
  c) Content:
       +0.4 for requiring synthesis of multiple concepts
       +0.25 for interdisciplinary relationships

3. RESPONSE MODEL:
  - Probability of correct = 1 / (1 + e^(-(ability - difficulty)
    ))
  - Systematic errors:
      40% chance of choosing a plausible distractor
      15% chance of choosing an absurd option
      Cognitive bias: preference for middle options (B, C, D)

4. INPUT:
  - List of student abilities
  - Question text followed by its options

5. REQUIRED OUTPUT:
  [1] List of selected options (AE)
  [2] Binary list of correct answers (0/1)
  [3] Percentage correct (one decimal)
  [4] Calculated question difficulty (scale 3.0 to 3.0)

###
STUDENT ABILITIES
{student_list}

###
QUESTIONS
{questions}

OUTPUT FORMAT
-------------
RETURN JSON WITH THE FOLLOWING STRUCTURE:

{output_format_prompt_c}
```