



Assistente Educativo para Aprendizado Ativo via Método Socrático

Eric Lopes Oliveira ^{*1}, Felipe de Lima Oliveira Alfaya ^{†1}, Sofia
Andrade Gois ^{‡1}, Mario Jorge Pereira ^{§1*}

¹ Análise e Desenvolvimento de Sistemas

Escola de Tecnologias

Universidade Católica do Salvador (UCSAL)

Av. Prof. Pinto de Aguiar, 2589 Pituaçu, CEP: 41740-090

Salvador/BA, Brasil

¹{*ericlopes.oliveira, felipe.alfaya, sofia.gois*}@ucsal.edu.br

^{1*}{*mario.pereira*}@pro.ucsal.edu.br

Junho 2025

Resumo

Este trabalho tem como objetivo propor o desenvolvimento de um Assistente Pedagógico Digital que é baseado em Inteligência Artificial tendo o foco no método de aprendizado ativo a partir do Método Socrático. Sendo o objetivo principal a concepção de uma ferramenta capaz de estimular a autonomia dos usuários com o raciocínio crítico no universo das atividades de Programação. A pesquisa analisa o uso de Modelos de Linguagem de Grande Escala (LLMs). Espera-se que a

*ericlopes.oliveira@ucsal.edu.br

†felipe.alfaya@ucsal.edu.br

‡sofia.gois@ucsal.edu.br

§mario.pereira@pro.ucsal.edu.br

abordagem da pesquisa contribua para a educação com mais interação, alinhada aos princípios da Educação Ativa.

Palavras-chaves: Inteligência Artificial, Assistente Pedagógico, Modelos de Linguagem de Grande Escala (LLMs), Método Socrático, Educação Ativa, Programação, Engenharia de Prompts, Groq.

1 Introdução

O uso do ChatGPT na educação tem gerado repercussões globais, evidenciando tanto seu potencial quanto os desafios no ambiente de aprendizagem (FÜTTERER et al., 2023). Nesse contexto, torna-se relevante o desenvolvimento de um assistente pedagógico baseado no método socrático, com o objetivo de estimular o raciocínio crítico em atividades de programação. A proposta busca superar práticas passivas de ensino, promovendo autonomia e pensamento reflexivo entre os alunos. Contudo, Rahe e Maalej (2025) observaram que alunos frequentemente solicitam ao ChatGPT soluções completas para tarefas de programação, em vez de utilizá-lo para compreender conceitos ou corrigir erros. Essa prática pode resultar em um ciclo de dependência, onde os alunos se tornam menos propensos a desenvolver habilidades analíticas essenciais.

Diante disso, este trabalho propõe a criação e desenvolvimento de uma ferramenta educacional baseada em IA que funcione como um Assistente Pedagógico. Essa ferramenta oferecerá explicações contextualizadas e orientações estruturadas, sem apresentar respostas completas. Utilizando princípios de engenharia de Prompts (WHITE et al., 2023), o Assistente Pedagógico será programado para atuar como mediador do conhecimento, incentivando a investigação guiada e a construção ativa do aprendizado.

1.1 Justificativa e Relevância

Em vez de simplesmente transmitir conhecimento, a promoção do pensamento crítico vai além de transmitir conteúdo: envolve estimular o raciocínio autônomo dos alunos. Assistentes pedagógicos baseados em modelos de linguagem, mesmo sem treinamento específico, podem conduzir diálogos reflexivos e eficazes (KOJIMA et al., 2022). Integrando o método socrático, essa abordagem favorece a construção ativa do conhecimento e potencializa o aprendizado em programação.

1.2 Objetivo Geral

Desenvolver um Assistente Pedagógico que utilize técnicas socráticas para estimular o raciocínio crítico de alunos em atividades de programação.

1.3 Objetivos Específicos

Criar um Assistente Pedagógico que incentive o aluno a pensar e aprender por meio de dicas que o guiem, usando estratégias inteligentes para tornar as conversas mais eficazes e estimulantes.

2 Referencial Teórico

2.1 Método Socrático na Educação

O Método Socrático é uma técnica pedagógica originada na Grécia Antiga, atribuída ao filósofo Sócrates (469–399 a.C.), que consiste no uso sistemático do questionamento para promover o raciocínio crítico e a construção do conhecimento por meio do diálogo. Diferentemente da simples transmissão de informações, o método busca instigar o aluno a refletir sobre suas próprias ideias, desconstruir suposições e alcançar conclusões fundamentadas por meio de perguntas estratégicas e reflexivas (BRUNSCHWIG; LLOYD; PELLEGRIN, 2003; PAUL; ELDER, 2007).

Historicamente, Sócrates utilizava a maiêutica — um processo dialético que, por meio de perguntas, auxiliava o interlocutor a “dar à luz” suas próprias respostas e compreensões — e a ironia socrática, que consistia em aparentar ignorância para estimular o outro a justificar suas afirmações e, assim, promover uma autoavaliação crítica (PAUL; ELDER, 2007). Essa metodologia, centrada no diálogo construtivo, posiciona o professor mais como um facilitador do conhecimento do que como um transmissor autoritário.

Os pilares do questionamento socrático na educação incluem:

- Maiêutica: estimular o pensamento autônomo do aluno através de perguntas que o levem a descobrir conceitos por si mesmo.
- Ironia: desafiar as crenças e opiniões do aluno para que ele perceba possíveis contradições e falhas em seu raciocínio.
- Diálogo: criar um ambiente de troca aberta, onde o conhecimento é construído coletivamente a partir da interação entre professor e aluno (PAUL; ELDER, 2007).

Diversos estudos contemporâneos têm demonstrado os benefícios pedagógicos do método socrático para o desenvolvimento do pensamento crítico, especialmente em ambientes educacionais tecnológicos. Favero et al. (2024), investigaram a aplicação de Assistentes virtuais educacionais que utilizam técnicas socráticas para estimular o raciocínio reflexivo de alunos em programação, evidenciando que essas interações promovem maior engajamento e autonomia no aprendizado. Além disso, pesquisas como as de Dakka (2015), Awedh et al. (2015) corroboram que o uso de ferramentas digitais baseadas no questiona-

mento socrático potencializa a colaboração e o pensamento crítico dos alunos, contribuindo para um aprendizado mais ativo e significativo .

Dessa forma, o Método Socrático, aliado às tecnologias educacionais, configura-se como uma estratégia eficaz para incentivar o aluno a pensar de forma crítica e independente, estimulando não apenas a memorização, mas a compreensão profunda dos conteúdos e a capacidade de resolver problemas complexos.

2.2 Aprendizado Ativo e Aprendizagem Baseada em Diálogo

A aprendizagem baseada em diálogo tem se firmado como uma ferramenta poderosa para incentivar o amadurecimento de um pensamento crítico, segundo o modelo proposto por Romano et al. (2021, p. 1) "alavancar orientações epistemológicas para aumentar a diversidade e a inclusão, encorajar o pensamento crítico em todos os modos de investigação e promover novas contribuições na ética aplicada".

Ao concordar que "todas as salas de aula são epistemologicamente plurais" os autores do artigo reforçam que o diálogo permite "profundo entendimento e respeito por várias formas de conhecimento" (ROMANO et al., 2021, p. 5) sendo que o papel do instrutor deve ser vasto, em diversas áreas de conhecimento, indo além da simples passagem de conteúdo: "inutores tornam-se partes iguais de professores e alunos" garantindo que esse processo da troca de conhecimento não seja interpretado por mestre e discípulo, mas como dois indivíduos trocando conhecimentos que serão absorvidos por ambas as partes.

O *framework* que foi desenvolvido permite que "oportunidades estruturadas para pensar em diferentes modos de investigação" (ROMANO et al., 2021, p. 3) fortalecendo a promoção de um espaço acadêmico que estimula o diálogo entre diferentes perspectivas e níveis de hierarquia, promovendo uma aprendizagem colaborativa.

Com auxílio do Assistente Pedagógico que é baseado em IA, a aplicação do Método Socrático e do Aprendizado Ativo como uma metodologia para a mudança de abordagem e práticas no ensino, com o objetivo de melhorar a qualidade do ensino e preparar os alunos para os desafios do século XXI garante que com o seu potencial os resultados são exponenciais.

2.3 Modelos de Linguagem e Engenharia de Prompts

Modelos de Linguagem de Grande Escala (LLMs) são sistemas de inteligência artificial projetados para compreender e gerar texto natural com base em grandes volumes de dados textuais. Esses modelos aprendem padrões linguísticos e relações contextuais para produzir respostas coerentes e contextualizadas (BROWN et al., 2020). Segundo Brown et al. (2020), um modelo de linguagem de grande escala, como o GPT-3, é capaz de gerar textos altamente contextuais, utilizando uma arquitetura baseada em transformadores para processar e produzir linguagem natural de forma autônoma. Essa capacidade torna

os LLMs ferramentas poderosas para diversas aplicações, incluindo a educação, onde podem apoiar processos pedagógicos ao oferecer respostas contextualizadas e raciocínios estruturados.

A utilização do Llama, desenvolvido pela Meta, exemplifica como os LLMs podem ser empregados para promover o aprendizado ativo. Sua arquitetura robusta e treinamento em grandes volumes de dados textuais possibilitam a geração de textos coerentes e contextualizados, o que é especialmente útil em áreas que exigem raciocínio lógico e crítico, como a programação. Segundo Touvron et al. (2023), o Llama foi projetado para gerar textos com alta qualidade e coerência, sendo altamente eficaz em tarefas que demandam um raciocínio estruturado e lógico. Contudo, para que esses modelos alcancem seu máximo potencial pedagógico, é essencial o uso de técnicas avançadas de engenharia de Prompts, as quais guiam o modelo a produzir respostas não apenas corretas, mas também explicativas e reflexivas.

Nesse sentido, a engenharia de Prompts assume papel fundamental, pois envolve a criação de instruções claras e detalhadas que orientam o modelo de linguagem a gerar respostas mais precisas e contextualizadas. Como observado por Reynolds e McDonell (2021, p. 6), "é útil abordar a programação de Prompts sob a perspectiva de restringir comportamentos: queremos um Prompt que não seja apenas consistente com a continuação desejada, mas que seja inconsistente com as continuações indesejadas" [tradução nossa]. Isso reforça que a eficácia de um Prompt não depende apenas de guiar o modelo ao resultado esperado, mas também de bloquear caminhos de resposta inadequados ou irrelevantes.

Entre as diversas técnicas de engenharia de Prompts, destaca-se o *Chain-of-Thought Prompting (CoT)*, que incentiva o modelo a resolver problemas passo a passo, explicando seu raciocínio ao longo do processo. Wei et al. (2022) evidenciaram que essa técnica aumenta significativamente a acurácia dos modelos em tarefas complexas de raciocínio lógico e matemático, áreas diretamente relacionadas ao ensino de algoritmos e estruturas de controle. Essa capacidade é essencial para que a interação do aluno com o Assistente Pedagógico educacional vá além da simples resposta pronta, estimulando o raciocínio crítico e o aprendizado ativo — princípios fundamentais já discutidos nas seções anteriores.

Além disso, Kojima et al. (2022) demonstraram que modelos instruídos com *Chain of Thought (CoT)* geram explicações mais claras e interpretáveis em tarefas de raciocínio lógico. Embora o estudo não se concentre especificamente no ensino de programação, essa abordagem pode contribuir para a compreensão de conceitos abstratos como recursão e estruturas condicionais, que frequentemente representam desafios para alunos. Ao expor o raciocínio passo a passo, o modelo aproxima-se de uma metodologia semelhante à socrática, incentivando a reflexão guiada e o autoconhecimento.

Por fim, segundo Podolefsky, Moore e Perkins (2013, p. 1), "a técnica de *scaffolding* implícito utiliza affordances, restrições, sinais e *feedback* para estruturar e apoiar a explo-

ração do aluno sem fornecer uma orientação explícita”. *Affordances* são as características ou pistas presentes no ambiente que indicam ao aluno as possíveis ações a serem tomadas, facilitando sua interação e exploração. Com base nessa abordagem, o *feedback* indireto é algo sutil e direcionado, que promove uma aprendizagem mais profunda e uma maior retenção de conhecimento, uma vez que o aluno se torna protagonista no processo de resolução de problemas.

A tabela da Figura 1 expõe diferentes métodos de uso do Prompt, com foco em destacar as suas referências e principais características. Cada método é descrito com base em sua aplicação e nas pesquisas que as validam.

Método de Uso do Prompt	Referência	Características que Diferem
Chain-of-Thought Prompting	Wei et al. (2022)	Introduz raciocínio passo a passo no prompt. Melhora desempenho em tarefas que exigem lógica ou cálculo.
Zero-Shot Prompting	Brown et al. (2020)	O modelo realiza tarefas sem exemplos no prompt. Avalia a capacidade de generalização do modelo com base apenas no treinamento prévio.
Few-Shot Prompting	Reynolds & McDonell (2021)	São fornecidos poucos exemplos no prompt. Isso ajuda a guiar o modelo para a estrutura e o estilo esperados.
Least-to-Most Prompting	Zhou et al. (2022)	Divide tarefas complexas em subtarefas incrementais, resolvendo de forma gradual. Fortalece o raciocínio encadeado de forma estrutural e progressiva.

Figura 1 – Métodos de uso do Prompt

Conforme ilustrado na Figura 1, a tabela reúne quatro dos métodos mais amplamente utilizados na construção de prompts para grandes modelos de linguagem (LLMs), como o GPT. Embora existam outras abordagens emergentes, essas quatro se destacam por sua eficácia demonstrada em múltiplos contextos de aplicação. Cada método representa uma estratégia distinta de interação com o modelo, impactando diretamente a qualidade das respostas conforme a estrutura do prompt.

O método *Chain-of-Thought Prompting*, proposto por Wei et al. (2022, p. 2), define que “uma cadeia de pensamento é uma série de etapas intermediárias de raciocínio em linguagem natural que levam ao resultado final, e nos referimos a essa abordagem como *chain-of-thought*” [tradução nossa]. Essa técnica introduz raciocínio passo a passo nos prompts, sendo especialmente eficaz em tarefas que exigem lógica, interpretação e cálculos intermediários, pois permite que o modelo organize seu processo de resposta de maneira mais estruturada.

Já o *Zero-Shot Prompting*, apresentado por Brown et al. (2020, p. 3), é descrito como “abordagem semelhante ao *few-shot*, porém utilizando apenas uma descrição da tarefa em linguagem natural, sem fornecer exemplos” [tradução nossa]. Isso significa que o modelo executa a tarefa com base apenas na compreensão geral adquirida durante o treinamento, sem exemplos explícitos no prompt, o que evidencia sua capacidade de generalização.

O *Few-Shot Prompting*, conforme discutido por ??, p. 1), parte do princípio de que “o GPT-3 alcançou desempenho de ponta em uma ampla variedade de tarefas sem a necessidade de ajuste fino (*fine-tuning*), utilizando apenas prompts do tipo *few-shot*, nos quais um pequeno número de exemplos de tarefas resolvidas é fornecido como parte da

entrada para o modelo treinado” [tradução nossa]. Esse método fornece exemplos que orientam o modelo quanto à estrutura, estilo e intenção da tarefa, resultando em respostas mais alinhadas com o objetivo esperado.

Por fim, o *Least-to-Most Prompting*, proposto por Zhou et al. (2022, p. 3), afirma que “o least-to-most prompting ensina modelos de linguagem a resolver um problema complexo por meio de sua decomposição em uma série de subproblemas mais simples” [tradução nossa]. Essa abordagem facilita o raciocínio encadeado ao guiar o modelo por etapas progressivas, promovendo uma resolução gradual e mais precisa de tarefas desafiadoras.

2.3.1 Arquitetura e Aceleração de IA (Groq)

A revolução dos modelos de linguagem de grande escala (LLMs) tem transformado a interação com a inteligência artificial, abrindo novas possibilidades para a educação por meio do aprendizado ativo e do método socrático, como discutido por Binns (2023). No entanto, o pleno aproveitamento desse potencial depende não apenas da qualidade dos modelos, mas também da infraestrutura capaz de processá-los eficientemente.

Quando comparadas a outras soluções, como *GPUs* da NVIDIA e *TPUs* do Google, as arquiteturas da Groq se destacam pela redução de latência e eficiência energética, além de possibilitar maior escalabilidade, o que torna a infraestrutura mais acessível para instituições que buscam implementar IA em larga escala sem incorrer em altos custos operacionais (JOUPPI et al., 2017). Isso é essencial para que o Assistente Pedagógico mantenha uma interação fluida, com respostas rápidas que possibilitem a aplicação dinâmica do método socrático e o estímulo contínuo do raciocínio crítico do usuário — elementos centrais já abordados nas seções anteriores. Diferente das *GPUs* convencionais, que são mais genéricas e comumente usadas para uma variedade de aplicações, as arquiteturas da Groq são personalizadas para aplicações específicas de IA, oferecendo uma aceleração significativamente maior (GROQ, 2023).

Além do ganho em desempenho, essa sinergia entre modelos abertos de linguagem e hardware dedicado democratiza o acesso à inteligência artificial avançada. Instituições educacionais com recursos limitados podem, assim, oferecer experiências interativas sofisticadas sem depender exclusivamente de soluções proprietárias e onerosas, ampliando o alcance e o impacto do aprendizado ativo baseado em diálogo (BOMMASANI et al., 2021).

Dessa forma, a integração entre modelos de linguagem e hardware especializado transcende o aspecto técnico e configura uma transformação pedagógica: ao viabilizar interações rápidas e econômicas, permite que a IA atue como parceira ativa no desenvolvimento cognitivo dos alunos, fomentando a construção crítica do conhecimento e a autonomia, pilares do método socrático explorado anteriormente.

Assim, essa arquitetura especializada é parte essencial do ecossistema educacional digital contemporâneo, fechando o ciclo que vai do questionamento socrático, passando

pelo aprendizado ativo, até a implementação eficaz com tecnologia de ponta.

3 Metodologia

Para avaliar a percepção de público-alvo sobre o Assistente Pedagógico com abordagem socrática desenvolvido, foi realizada uma pesquisa de natureza qualitativa. Esta metodologia foi escolhida para coletar *feedbacks* diretos sobre a experiência de uso, a utilidade da ferramenta e a eficácia do método socrático implementado.

3.1 Participantes

Os participantes da pesquisa foram professores da área de tecnologia, convidados a interagir com o protótipo do Assistente Pedagógico. A seleção se deu por conveniência, buscando profissionais com experiência no ensino de programação que pudessem oferecer uma avaliação crítica da ferramenta.

3.2 Instrumento de Coleta de Dados

Foi utilizado um questionário digital, elaborado na plataforma Google Forms. O formulário foi estruturado com perguntas fechadas (de múltipla escolha) e abertas, permitindo a coleta de dados quantitativos sobre a percepção geral e dados qualitativos sobre sugestões de melhoria.

O questionário abordou os seguintes eixos principais, baseados nas perguntas:

- **Utilidade:** Avaliar se o "Assistente Pedagógico foi útil para ajudá-lo a resolver problemas de programação?".
- **Usabilidade:** Medir a percepção sobre a facilidade de uso com a pergunta: "A interface do Assistente Pedagógico foi fácil de usar e entender?".
- **Eficácia do Método Socrático:** Verificar se "A abordagem do Método Socrático fez você pensar mais sobre suas próprias ideias e raciocínios?".
- **Sugestões de Melhoria:** Coletar *feedbacks* abertos para futuras iterações da ferramenta através das perguntas "Quais melhorias você sugeriria para o Assistente Pedagógico?" e "Outras Sugestões".

3.3 Procedimentos

Os procedimentos para a coleta de dados seguiram as seguintes etapas:

1. Os professores participantes foram convidados a interagir livremente com o Assistente Pedagógico, testando suas funcionalidades em um cenário simulado de Assistente Pedagógico.
2. Após a interação, foi disponibilizado o *link* para o questionário no Google Forms.
3. Os participantes responderam ao formulário, e os dados foram registrados automaticamente pela plataforma, incluindo o carimbo de data/hora de cada resposta.
4. Os dados coletados foram então exportados e consolidados para análise.

3.4 História de Usuário

Como aluno, quero aprender a solucionar um problema de programação encontrado durante uma atividade do curso de TI com a ajuda do Assistente Pedagógico, para melhorar minhas habilidades em lógica de programação, entender conceitos e aplicá-los em problemas reais, ao invés de simplesmente seguir instruções prontas.

3.5 Casos de Uso

- **Título:** Solucionar um Problema de Lógica de Programação com o Método Socrático.
- **Ator:** Aluno
- **Objetivo Principal:** Resolver um problema de lógica para calcular a média inteira de dois números fornecidos, utilizando a orientação do Assistente Pedagógico através do método socrático.

3.6 Critérios de Aceitação

1. O Aluno deverá ser capaz de solucionar autonomamente o problema da média inteira utilizando conhecimentos transmitidos pelo Assistente Pedagógico durante a interação.
2. O Assistente Pedagógico utilizará o Método Socrático para estimular o pensamento crítico, fornecendo orientações sem entregar respostas diretas, guiando o Aluno com conceitos e aplicações práticas ao longo do processo.
3. O Aluno deverá aplicar as orientações recebidas do Assistente Pedagógico para implementar corretamente o sistema que calcula automaticamente a média inteira de dois números inteiros fornecidos.

3.7 Pré-Condições

1. O Aluno terá um problema de programação que precisa ser resolvido.
2. O Assistente Pedagógico estará disponível para guiar o Aluno com o Método Socrático.

3.8 Fluxo Principal

- A) O Aluno acessa o sistema e descreve o problema em um campo de texto.
- B) O Assistente Pedagógico inicia a conversa, guiando o Aluno com perguntas reflexivas usando o Método Socrático.
- C) O Assistente Pedagógico apresenta perguntas que incentivam o Aluno a refletir sobre o contexto do problema.
- D) O Aluno responde às perguntas e faz novas perguntas para aprofundar seu entendimento.
- E) O Assistente Pedagógico continua com orientações até que o Aluno compreenda completamente o processo.
- F) O Aluno implementa o código conforme as orientações e verifica o resultado.

3.9 Pós-Condições

1. O aluno consegue implementar o que foi aprendido.
2. O problema é solucionado pelo Método Socrático, promovendo o fortalecimento das habilidades críticas e conhecimento do aluno.

3.10 Modelos C4

Para representar de forma clara e objetiva o sistema e as suas interações, foram construídos os Modelos C4, divididos nas camadas:

Camada de Contexto (Figura 2)

A Figura 2 apresenta uma visão ampla das interações entre o Aluno, o Assistente Pedagógico, o Navegador e os serviços externos do GroqCloud API, com foco em abordar as finalidades e suas dependências.

Camada de Container (Figura 3)

A Figura 3 detalha os componentes técnicos, ilustrando a comunicação do Back-End com as rotas da API e a Aplicação *Web* que foi desenvolvida em Next.js com React e o serviço externo GroqCloud API com foco no processamento das requisições e das respostas com o uso do modelo Llama 3.1.80.

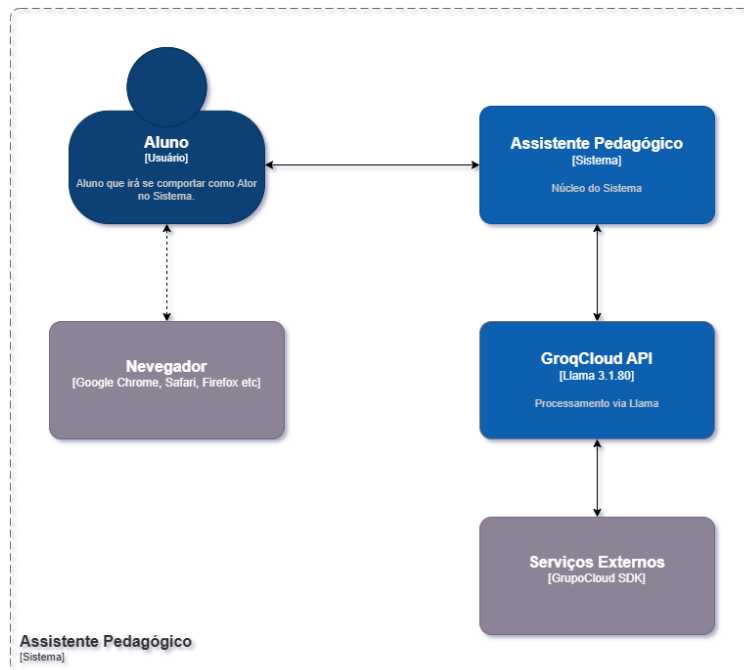


Figura 2 – Camada Contexto

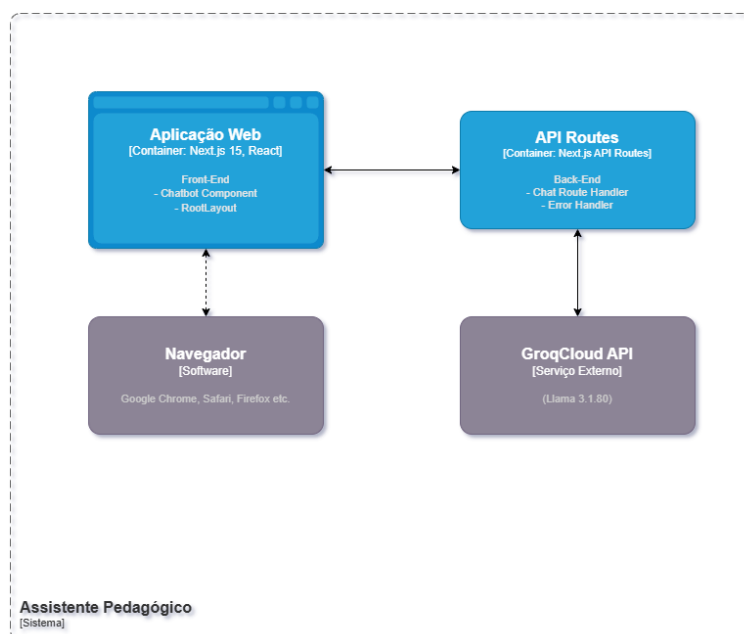


Figura 3 – Camada Container

4 Projeto

O nascimento desse projeto possui princípios tecnológicos no viés pedagógico, com o destaque no uso do Método Socrático e Aprendizado Ativo. A solução desenvolvida abrange um Assistente Pedagógico que se baseia no modelo de linguagem de grande escala (LLM), o Llama, que é integrado com a infraestrutura para aumentar a eficiência com o Groq, com objetivo de trazer uma experiência ao usuário final a interação em tempo real.

4.1 Construção do Método Socrático da Solução

A finalidade da proposta para elaboração do sistema está em sua construção de diálogo entre os atores Assistente e Aluno, que se desenvolve semelhante a um Assistente Virtual Pedagógico. Em disparidade dos *Softwares* tradicionais que o assistente ao invés de dar respostas subjetivas, entrega as respostas de forma direta adotando o método de questionamentos guiados, nessa abordagem, que parte do princípio do método socrático clássico, pretende romper com as concepções prévias e fornecer recursos para o fortalecimento do pensamento crítico utilizando artifícios da ironia construtiva e miaêutica digital.

Essa interação com o aluno a partir de uma interface amigável que remete a um *Chat* onde o o assistente pedagógico guia o diálogo com perguntas encadeadas promovendo a formação de prognósticos, fundamentos e um aprendizado constante. Sendo o seu objetivo principal despertar o protagonismo do aluno, em harmonia com os pressupostos pedagógicos defendidos por Dewey (1916) e Piaget (1970).



Figura 4 – Interface da Aplicação Assistente Pedagógico.

Interface da Aplicação

A Figura 4 apresenta a interface amigável da aplicação Assistente Pedagógico. O usuário é recebido com uma saudação e uma breve explicação sobre o método socrático utilizado para auxiliar no aprendizado através de perguntas e reflexões. A interface de chat permite que o aluno digite suas perguntas e explore diversos assuntos, promovendo uma interação guiada pelo assistente.

4.2 Infraestrutura Tecnológica

Com foco na facilitação da interação entre o Assistente Pedagógico e o Aluno de forma instantânea, foi adotada na concepção do projeto uma arquitetura baseada em computação de grande desempenho, utilizando o Llama, alinhado com o Groq que aumenta significativamente a aceleração através dos *Hardware*s de alta performance. Onde é projetado para processamento de IA com aplicabilidade em executar bilhões de instruções por segundo a resposta é enviada para o aluno de forma consistente e rápida.

Seguindo o modelo Cliente-Servidor, o Aluno sendo o lado do cliente interagindo com uma interface leve e amigável, acessada através do navegador e o Assistente sendo o lado do servidor, onde toda a lógica é processada sem interferir no uso da plataforma, gerando uma linguagem natural e humanizada, interpretando as mensagens e formulando as respostas através do modelo Llama.

A divisão entre o cliente e o servidor tem como a principal vantagem em relação a outros modelos a drástica redução da carga de processamento no dispositivo em que o usuário acessa, permitindo que as máquinas com baixo desempenho computacional possam usufruir da experiência da mesma forma em que uma máquina de alto desempenho computacional. O uso da Groq para melhorar a execução com baixa latência é importante para manter a interação, mantendo o ritmo natural e fluido da conversa.

4.3 Engenharia de Prompt e Procedimento de Raciocínio

A estrutura do prompt foi cuidadosamente elaborada para promover um diálogo em etapas, com o objetivo de estimular a reflexão e o questionamento por parte do estudante. Para isso, adotou-se a técnica conhecida como *Chain-of-Thought Prompting*, não apenas por sua eficiência técnica, mas sobretudo por seu valor pedagógico. Essa abordagem orienta o modelo de linguagem a formular respostas explicativas e estruturadas, criando um ambiente propício para que o aluno acompanhe o raciocínio, compreenda o encadeamento lógico das ideias e participe ativamente da construção do conhecimento.

Ao invés de apenas instruir o Assistente Pedagógico a responder corretamente, o Prompt que foi concebido com objetivo de estimular o desenvolvimento do conhecimento é a proposta deste projeto, onde a Engenharia de Prompt que é habitualmente tratada como um componente técnico para sistemas baseados em inteligência artificial, a LLM é uma ferramenta para mediação do aprendizado.

4.4 Inovação Tecnológica e Pedagógica

A disparidade com sistemas de Assistente Pedagógico que são fortemente baseados em instruções e regras bem estabelecidas, o Assistente Pedagógico aqui proposto utiliza métodos de personalização e adequação com base no perfil que o Prompt consegue distinguir

dos alunos. Essa habilidade é capaz de ajustar dinamicamente customizando o aprendizado em consonância com as propensões da sociedade contemporânea de educação digital focada no desenvolvimento do pensamento crítico para o aluno.

Em suma, este projeto estabelece fundamentos educacionais imutáveis com tecnologias de ponta em IA, firmando-se como um produto inovador no contexto do ensino de programação. Ao mesclar o poder das LLMs com o Método Socrático e para aumento de performance computacional da Groq, a solução se dá como uma alternativa eficaz e customizada, estimulando o protagonismo do aluno e fortalecimento de suas habilidades cognitivas com foco no pensamento crítico.

5 Validação e Resultados

Para validar a aplicabilidade do Assistente Pedagógico que foi desenvolvido neste projeto, foi realizada uma pesquisa com professores da área de tecnologia da informação. Esses resultados foram coletados a partir de um formulário do Google Forms, onde em cada pergunta foi empregado a Escala de Likert, com níveis de utilidade, eficácia do Método Socrático e usabilidade. A análise dos resultados pode ser observada nas figuras a seguir:

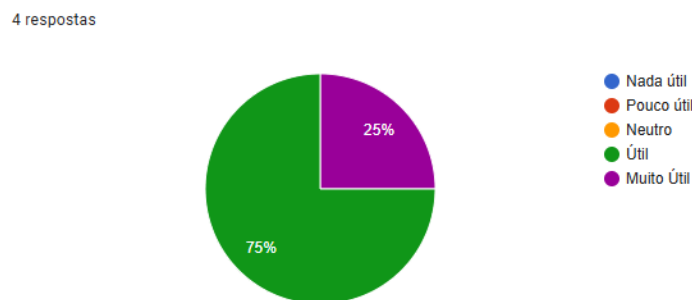


Figura 5 – Utilidade do Assistente Pedagógico na Resolução de Problemas de Programação.

A figura 5 demonstra que 75% dos professores que participaram da pesquisa classificaram a Utilidade do Assistente Pedagógico como "Útil" e 25% como "Muito Útil", indicando que o produto desenvolvido foi bem recebido e considerada eficiente em sua proposta.

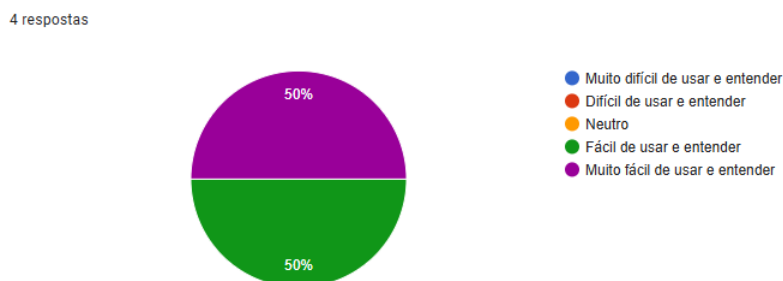


Figura 6 – Avaliação da Usabilidade da Interface do Assistente Pedagógico.

A figura 6 demonstra que 50% dos professores que participaram da pesquisa classificaram a Interface do Assistente Pedagógico "Muito fácil de usar e entender" e a outra metade "Fácil de usar e entender", indicando que a usabilidade é simples de se utilizar.

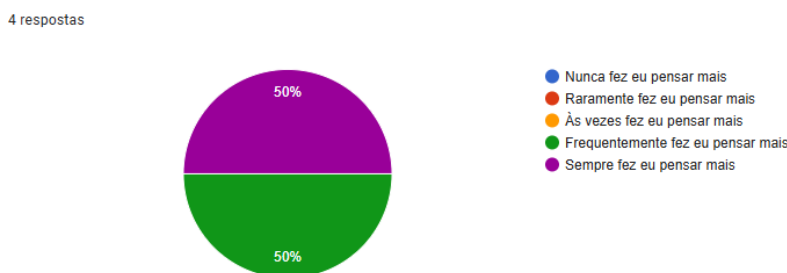


Figura 7 – Percepção sobre a Eficácia do Método Socrático utilizado pelo Assistente Pedagógico.

A figura 7 demonstra que 50% dos professores que participaram da pesquisa classificaram que o uso do Método Socrático "Sempre fez pensar mais" e a outra metade "Frequentemente me fez pensar mais", provando que esse tipo de abordagem estimula o pensamento crítico e o amadurecimento das habilidades cognitivas.

5.1 Sugestões de Melhorias ao Produto

Além das perguntas de escolha única, os professores indicaram melhorias, destacando a grande parte em mudanças na Interface Gráfica, além da formatação de texto. Esses palpites fornecem uma tendência de que a interface do produto é um pilar importante para o usuário final. Com essas apurações é possível afirmar que o Assistente Pedagógico cumpre com o seu objetivo, com uma forte aprovação dos entrevistados, ademais, existem melhorias a serem implementadas em possíveis atualizações.

6 Conclusão

Para validar a abordagem e a usabilidade do protótipo, foi conduzida uma pesquisa qualitativa com professores da área de tecnologia, cujos resultados foram altamente promissores e corroboram a hipótese central do projeto. Os participantes avaliaram a ferramenta como "Útil" a "Muito Útil" na resolução de problemas e consideraram sua interface "Fácil de usar e entender". O ponto de maior relevância na avaliação foi a confirmação de que a abordagem socrática "Sempre" ou "Frequentemente" incentivou os usuários a refletirem mais profundamente sobre suas próprias linhas de raciocínio, atingindo o principal objetivo pedagógico da ferramenta.

Apesar do sucesso na validação dos objetivos primários, a pesquisa também foi fundamental para identificar oportunidades claras de aprimoramento. É importante ponderar que a avaliação foi realizada com uma amostra reduzida, servindo como um estudo piloto. As sugestões dos avaliadores direcionam os próximos passos do desenvolvimento, que incluem melhorias na experiência do usuário (UX), como a implementação de formatação de texto (negrito e cores), a sintaxe colorida para trechos de código (*syntax highlighting*) e a personalização da interface, como a escolha entre temas claro e escuro.

Conclui-se, portanto, que este trabalho atingiu com êxito seu objetivo principal. Foi concebido e validado um protótipo de Assistente Pedagógico que não apenas funciona como uma ferramenta de auxílio, mas que, através do Método Socrático e de técnicas de IA robustas, se posiciona como um agente catalisador do pensamento crítico. A pesquisa demonstra o potencial significativo dos LLMs, quando alinhados a princípios pedagógicos sólidos e técnicas de prompt avançadas, para transformar a educação em programação, tornando-a mais interativa, reflexiva e alinhada aos preceitos da Educação Ativa.

Referências

- AWEDH, M. et al. Using socratic and smartphones for the support of collaborative learning. *arXiv preprint arXiv:1501.01276*, 2015.
- BINNS, D. Ai and the socratic method: Engaging with technology through critical thinking. *Journal of Educational Technology*, Springer, v. 45, p. 121–134, 2023.
- BOMMASANI, R. et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.
- BROWN, T. B. et al. Language models are few-shot learners. *Advances in neural information processing systems*, v. 33, p. 1877–1901, 2020.
- BRUNSCHWIG, J.; LLOYD, G. E. R.; PELLEGRIN, P. *A Guide to Greek Thought: Major Figures and Trends*. [S.l.]: Harvard University Press, 2003.

- DAKKA, S. M. Using socratic to enhance in-class student engagement and collaboration. *International Journal on Integrating Technology in Education, Academy and Industry Research Collaboration Center (AIRCC)*, v. 4, n. 3, p. 13–19, set. 2015. ISSN 2320-1886. Disponível em: <<http://dx.doi.org/10.5121/ijite.2015.4302>>.
- DEWEY, J. *Democracy and Education: An Introduction to the Philosophy of Education*. [S.l.]: Macmillan, 1916.
- FAVERO, L. et al. Enhancing critical thinking in education by means of a socratic chatbot. *arXiv preprint arXiv:2409.05511*, 2024.
- FÜTTERER, T. et al. Chatgpt in education: global reactions to ai innovations. *Scientific reports*, Nature Publishing Group UK London, v. 13, n. 1, p. 15310, 2023.
- GROQ. *Why AI Requires a New Chip Architecture*. 2023. <https://groq.com/why-ai-requires-a-new-chip-architecture/?utm_source=chatgpt.com>. Accessed: 2023-06-05.
- JOUPPI, N. P. et al. In-datacenter performance analysis of a tensor processing unit. In: *Proceedings of the 44th annual international symposium on computer architecture*. [S.l.: s.n.], 2017. p. 1–12.
- KOJIMA, T. et al. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, v. 35, p. 22199–22213, 2022.
- PAUL, R.; ELDER, L. Critical thinking: The art of socratic questioning. *Journal of developmental education*, Appalachian State University d/b/a, v. 31, n. 1, p. 36, 2007.
- PIAGET, J. Science of education and the psychology of the child. trans. d. coltman. Orion, 1970.
- PODOLEFSKY, N. S.; MOORE, E. B.; PERKINS, K. K. Implicit scaffolding in interactive simulations: Design strategies to support multiple educational goals. *arXiv preprint arXiv:1306.6544*, 2013.
- RAHE, C.; MAALEJ, W. How do programming students use generative ai? *arXiv e-prints*, p. arXiv-2501, 2025.
- REYNOLDS, L.; MCDONELL, N. Prompt programming for large language models: Beyond the few-shot paradigm. *arXiv preprint arXiv:2102.07350*, 2021.
- ROMANO, M. R. et al. Dialogue-based learning: A framework for inclusive science education and applied ethics. *Frontiers in Communication*, Frontiers Media SA, v. 6, p. 731839, 2021.
- TOUVRON, H. et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.
- WEI, J. et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, v. 35, p. 24824–24837, 2022.
- WHITE, J. et al. A prompt pattern catalog to enhance prompt engineering with chatgpt. *arXiv preprint arXiv:2302.11382*, 2023.

ZHOU, D. et al. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*, 2022.