MatBot – *chatbot* de auxílio ao ensino da matemática

Trabalho da disciplina de Inteligencia Artificial e Machine Learning

Universidade UniAcademia – Juiz de Fora, MG - 2025

Professor: doutor Tassio Sirqueira
Alunas: Júlia Lopes,
Marília Carvalho

1. Introdução

Os *chatbots* tem um papel crescente na interação humano-computador, isso porque possibilitam uma comunicação adaptada e personalizada entre usuários e sistemas digitais (Kuhn, 2020). Segundo Kuhn (2020), a inteligência artificial (IA) parece estar mais presente nas nossas atividades, possibilitando novas formas de interação com computadores. Essas tecnologias, aliadas ao aprendizado de máquina (ML), oferecem uma melhor compreensão de comandos, o que torna as respostas mais precisas e naturais.

Além disso, Kuhn (2020) afirma que a IA nos *chatbots* permite que sejam oferecidas experiências personalizadas, onde as sugestões e os conteúdos apresentados são contextualizados para aquele usuário. Essa tendência tecnológica pode gerar valor tanto para consumidores quanto para empresas, já que contribui para o desenvolvimento de um mercado mais interativo e adaptável às necessidades dos usuários (Kuhn, 2020). A evolução dos *chatbots* com a implementação de IA e ML continua enriquecendo as possibilidades de comunicação digital e aprimorando a experiência do usuário (Kuhn, 2020).

Desde os primeiros experimentos de Joseph Weizenbaum com o ELIZA, desenvolvido entre 1964 e 1966, a inteligência artificial tem avançado significativamente na capacidade de simular interações humanas (SILVA, 2024). O impacto do ELIZA foi marcante por conta do seu sistema de correspondência de padrões permitia que o programa respondesse de maneira que estimulava a continuidade da conversa, criando a ilusão de compreensão emocional. Esse fenômeno levantou questões éticas e psicológicas sobre a relação entre humanos e máquinas (SILVA, 2024).

Décadas depois, a evolução da IA se concretizou com os modelos de linguagem da série GPT, como o GPT-3 ou o GPT-4 mais recentemente, que, com seus mais de 175 bilhões de parâmetros, trouxe avanços notáveis em síntese de texto, tradução e compreensão contextual (SILVA, 2024). Esses avanços demonstram a crescente sofisticação das interações humano-computador, impulsionando novas formas de

comunicação digital e transformando setores como atendimento ao cliente, publicidade e criação de conteúdo.

Os avanços em inteligência artificial e processamento de linguagem natural têm transformado a interação humano-computador, tornando os *chatbots* uma peça central nesse processo (SILVA, 2024). Segundo Følstad e Brandtzaeg (2017 apud SILVA, 2024), essas tecnologias estão reinventando as interfaces de usuário ao oferecer uma forma de interação mais natural e intuitiva em comparação com os tradicionais menus e comandos.

No âmbito educacional, os pesquisadores destacam o potencial dos *chatbots* como tutores virtuais, permitindo suporte imediato e personalizado para estudantes, auxiliando na resolução de dúvidas e na assimilação de conteúdo (FØLSTAD e BRANDTZAEG, 2017 apud SILVA, 2024). Dessa forma, a evolução dos *chatbots* não apenas melhora a usabilidade das interfaces digitais, mas também amplia as possibilidades de aprendizado, promovendo um ambiente mais acessível e dinâmico para os usuários (FØLSTAD e BRANDTZAEG, 2017 apud SILVA, 2024).

Neste estudo, foi adotada a ferramenta de *Machine Learning* Llama, desenvolvida pela Meta, juntamente com a biblioteca *PyTorch*, uma das principais ferramentas para o desenvolvimento de redes neurais profundas, e *Transformers* para facilitar a utilização de modelos de linguagem pré-treinados (SANTOS, 2024).

Segundo Hybl (2024, p.40), a família do modelo Llama se destaca no campo dos modelos de linguagem, superando diversas opções de código aberto, como MPT, Falcon e Vicuna, tanto na função de complementar o texto quanto na geração de diálogos. Além disso, seu desempenho é comparável ou superior a modelos fechados, como GPT-3.5 e PaLM.

Devido à sua conexão com a Meta, Llama oferece melhor integração com plataformas e documentação mais robusta, tornando-se uma alternativa atrativa para desenvolvedores que precisam ajustar modelos para tarefas específicas (HYBL, 2024). A família Llama inclui modelos de base para complementar o texto em diferentes tamanhos (7, 13, 34 e 70 bilhões de parâmetros) e versões ajustadas para interações conversacionais (HYBL, 2024).

2. Fundamentação Teórica

Os *chatbots* são definidos como softwares capazes de interagir com usuários por meio de linguagem natural, proporcionando uma comunicação fluida e acessível

(MORAES; DE SOUZA, 2015 apud LARISANE KUYVEN et al., 2018). Por meio de mensagens de texto ou voz, esses assistentes virtuais utilizam IA, ML e redes neurais para interpretar e responder às perguntas dos usuários de maneira eficiente (NETO ET AL., 2023, apud SANTOS, 2025).

A comunicação com esses sistemas ocorre majoritariamente por meio de aplicativos de mensagens e plataformas de *chat*. Além de responder às perguntas dos usuários, os *chatbots* podem enviar links e arquivos automaticamente, adaptando-se às necessidades de cada pessoa (LUGLI & LUCCA FILHO, 2020, apud SANTOS, 2025).

O desenvolvimento deles é resultado de avanços científicos ao longo dos anos em diversas áreas, como IA, processamento de linguagem natural, bancos de dados e redes de comunicação. Essas tecnologias possibilitaram a criação de sistemas cada vez mais sofisticados, capazes de interagir de forma natural e eficaz com os usuários (LUGLI & LUCCA FILHO, 2020, apud SANTOS, 2025).

O ML constitui uma área da ciência computacional voltada para o desenvolvimento de modelos teóricos e implementação de sistemas capazes de aprender e adaptar-se. Essa aprendizagem envolve a construção ou a modificação automática de representações de dados, cenários ou objetos em avaliação (MICHALSKI, CARBONELL & MITCHELL, 1986 apud SAIAS et al., 2018).

O propósito central da ML é viabilizar soluções computacionais que executem tarefas ou otimizem seu desempenho com base na experiência prévia, minimizando a intervenção humana (SAIAS et al., 2018, p. 2). A crescente popularização dos *chatbots* e suas ferramentas associadas decorre do objetivo central da IA, que visa à criação de máquinas capazes de interagir com humanos, e da preferência dos usuários por interfaces baseadas em linguagem natural (LEONHARDT et al., 2003; MORAES; DE SOUZA, 2015, apud LARISANE KUYVEN et al., 2018). Para que seja considerado realmente eficiente, o *chatbot* deve compreender as demandas dos usuários, aplicar regras operacionais adequadas, acessar bancos de dados e fornecer respostas claras e precisas, dessa forma, a experiência do usuário torna-se mais satisfatória e produtiva (SANTOS, 2025).

Nesse contexto, identificam-se duas abordagens fundamentais para a aprendizagem de ML: a aprendizagem supervisionada e a não supervisionada. A distinção entre ambas reside na presença ou ausência do valor de saída esperado, acompanhado dos dados de treino de cada instância (SAIAS et al., 2018, p. 2).

Na aprendizagem supervisionada, cada instância de treino dispõe de valores de entrada, bem como do correspondente valor de saída, permitindo que o mecanismo de aprendizagem induza a relação entre ambos (SAIAS et al., 2018). Dentro dessa abordagem, destacam-se duas categorias principais: a classificação, quando o valor de saída é discreto ou categórico, e a regressão, na qual o valor de saída apresenta características contínuas (SAIAS et al., 2018).

Por outro lado, segundo Saias et al. (2018), na aprendizagem não supervisionada, as instâncias de treino contêm apenas os valores de entrada, sem indicação explícita do valor de saída esperado. O aprendizado ocorre mediante análise de similaridades nos dados, sem informações previamente fornecidas sobre o objetivo final. Uma aplicação clássica dessa abordagem é o agrupamento de dados, conhecido como *clustering* (SAIAS et al., 2018).

Além desses modelos fundamentais, outras estratégias de aprendizagem automática têm sido desenvolvidas. A aprendizagem semi-supervisionada combina elementos das duas abordagens citadas, sendo aplicada em contextos onde há disponibilidade tanto de dados anotados quanto de dados não anotados. Nesse caso, utiliza-se uma abordagem supervisionada para atribuir rótulos às instâncias sem classificação prévia (SAIAS et al., 2018).

Outro conceito relevante é o *reinforcement learning* (aprendizagem por reforço), no qual um agente computacional aprende progressivamente por meio da interação com o ambiente, recebendo feedbacks na forma de penalizações ou recompensas, conforme suas ações ou respostas (SAIAS et al., 2018).

Já o *deep learning* (aprendizagem profunda) baseia-se em redes neurais multicamadas inspiradas no funcionamento do cérebro humano. Embora redes neurais sejam utilizadas há décadas, seu potencial foi significativamente aprimorado graças ao avanço da capacidade computacional contemporânea, permitindo o treinamento eficiente de modelos complexos (SAIAS et al., 2018).

No domínio da mineração de dados, ou *Data Mining*, a aprendizagem automática se desdobra em técnicas como classificação, regressão, agrupamento *clustering* e identificação de padrões frequentes (*frequent pattern mining*) (SAIAS et al., 2018). A análise preditiva, por sua vez, fundamenta-se na descoberta de padrões em dados históricos por meio de métodos estatísticos multivariados e/ou técnicas de ML, auxiliando na previsão de eventos futuros (SAIAS et al., 2018). Estratégias de predição de *churn*

frequentemente integram múltiplas tarefas de mineração de dados, incluindo *clustering* e classificação (SAIAS et al., 2018).

Santos (2025) afirma que no passado, alguns modelos de *chatbots* foram criados com base em regras fixas e, embora esses sistemas fossem mais simples de desenvolver e implementar, eles enfrentavam dificuldades para responder a questões mais complexas. Seu funcionamento dependia da busca por padrões nas consultas dos usuários, o que muitas vezes resultava em respostas imprecisas diante de frases sem correspondência conhecida (SANTOS, 2025).

Dessa forma, é possível inferir que a evolução desses sistemas se deve à incorporação de novas técnicas de IA, permitindo sua categorização em três paradigmas distintos: a primeira geração, baseada na correspondência de padrões e regras gramaticais; a segunda geração, fundamentada em regras de produção e redes neurais artificiais; e a terceira geração, que utiliza linguagens de marcação AIML1 (SGOBBI et al., 2014, apud LARISANE KUYVEN et al., 2018).

Os *chatbots* modernos, equipados com IA e ML, empregam aprendizado de máquina para interpretar dados provenientes de interações humanas anteriores, esse processo envolve o treinamento do sistema com conjuntos específicos de informações, permitindo que ele desenvolva maior capacidade de compreensão e ofereça respostas mais eficazes às consultas dos usuários (BATISTA & SOUZA, 2023, apud SANTOS, 2025).

3. Estado da Arte

A interação contínua entre usuários e *chatbots* permite que a IA aprimore sua precisão e eficácia. Com base no *feedback* recebido, o sistema ajusta seu funcionamento e melhora sua capacidade de resposta, tornando-se cada vez mais adaptado às necessidades individuais dos usuários (BATISTA & SOUZA, 2023, apud SANTOS, 2025). Além de facilitar a comunicação, eles contribuem para a otimização de tarefas diárias, beneficiando tanto clientes quanto funcionários. Processos antes demorados e burocráticos foram simplificados, tornando-se mais ágeis e acessíveis graças a essa tecnologia (SANTOS, 2025).

Diversos estudos contribuem para a compreensão do desenvolvimento e aplicação dos *chatbots*. Soliman e Guetl (2010, apud LARISANE KUYVEN et al., 2018) realizam uma revisão sistemática da literatura sobre agentes pedagógicos inteligentes em ambientes virtuais de aprendizagem. Sewald Junior et al. (2011, apud LARISANE

KUYVEN et al., 2018, p.2) investigam os principais desafios relacionados à fluidez de conversação, com foco nos agentes que operam em língua portuguesa. Bradeško e Mladenić (2012, apud LARISANE KUYVEN et al., 2018, p.2) analisam as tecnologias adotadas pelos *chatbots* vencedores do Prêmio Loebner, observando sua evolução ao longo dos anos. Dale (2017, apud LARISANE KUYVEN et al., 2018) examina o cenário comercial no campo do processamento de linguagem natural, do qual os sistemas conversacionais fazem parte. Por fim, Bernardini, Sônego e Pozzebon (2018, apud LARISANE KUYVEN et al., 2018) realizam uma análise bibliométrica qualitativo-quantitativa, identificando tendências de pesquisa sobre *chatbots* em âmbito global.

Observa-se que, embora a maioria dos estudos sobre o uso de *chatbots* no ensino esteja concentrada em áreas específicas, principalmente na Ciência da Computação, também há pesquisas que buscam desenvolver sistemas flexíveis, estruturados com lógica genérica, capazes de servir como modelos aplicáveis a diversas disciplinas ou com objetivos educacionais distintos, tais como estimular a meta-cognição e a autorregulação (AGUIAR, TAROUCO & REATEGUI, 2011, apud KUYVEN et al., 2018) e incentivar o aprendizado colaborativo (DYKE et al., 2013, apud KUYVEN et al., 2018). Essa abordagem apresenta vantagens no contexto educacional, permitindo que uma mesma ferramenta contribua para diferentes áreas de conhecimento (KUYVEN et al., 2018).

Paralelamente, a educação tem passado por transformações significativas impulsionadas pela IA. Esse processo evidencia uma nova estrutura educacional caracterizada pela personalização da aprendizagem, pelo fornecimento de *feedback* em tempo real e pela adaptação às necessidades individuais dos alunos (CHEN, CHEN & LIN, 2020; TAPALOVA, ZHIYENBAYEVA & GURA, 2022, apud FEITOSA et al., 2025). A IA já está revolucionando a transmissão e assimilação do conhecimento por meio de tutores virtuais inteligentes, plataformas adaptativas de ensino e ferramentas automatizadas de avaliação (HASHIM et al., 2022; HASIBUAN & AZIZAH, 2023, apud FEITOSA et al., 2025).

A matemática enfrenta desafios na sua integração ao ambiente educacional, assim, para Hardman (2019 apud FEITOSA 2025), Hwang e Tu (2021 apud FEITOSA et al. 2025), torna-se essencial garantir rigor conceitual e fomentar o raciocínio crítico para que a IA possa ser empregada eficazmente no ensino da matemática. Dessa forma, a colaboração entre especialistas das áreas de matemática, pedagogia e tecnologia é crucial

para superar barreiras e viabilizar a implementação eficiente da IA nas salas de aula (HARDMAN, 2019; HWANG & TU, 2021, apud FEITOSA et al., 2025).

Para Zawacki-Richter (2019, apud FEITOSA et al., 2025), o avanço da IA na educação proporciona benefícios expressivos, como o desenvolvimento de ferramentas adaptadas às necessidades dos estudantes, sistemas de tutoria inteligentes e plataformas de avaliação automatizadas. Adicionalmente, Zhang e Aslan (2021, apud FEITOSA et al., 2025) afirmam que a IA contribui para a identificação de alunos que enfrentam dificuldades, permitindo a personalização do conteúdo e a criação de ambientes de aprendizagem mais dinâmicos. A longo prazo, a tecnologia possui potencial para enfrentar desafios globais complexos, promovendo melhorias na qualidade de vida e possibilitando novas descobertas no âmbito do conhecimento humano (ZHANG & ASLAN, 2021, apud FEITOSA et al., 2025).

Com isso é possível dizer que a IA tem se consolidado como um agente de transformação no cenário educacional, possibilitando a personalização do aprendizado por meio da otimização de processos e da democratização do acesso ao conhecimento (FEITOSA et al., 2025). Seu potencial para revolucionar o ensino e a aprendizagem impulsiona o desenvolvimento de ferramentas inovadoras que modificam as práticas pedagógicas, redefinindo a relação entre alunos e professores (FEITOSA et al., 2025).

A literatura acadêmica tem evidenciado a eficácia da IA no ensino da matemática, demonstrando impactos positivos em diversos níveis educacionais e conteúdos específicos (ADELABU, MAKGATO & RAMALIGELA, 2019 apud FEITOSA et al., 2025). No contexto do ensino médio, pesquisas indicam que softwares de simulação voltados para a geometria promovem uma melhora significativa na compreensão dos conceitos e na capacidade de resolução de problemas geométricos (ADELABU, MAKGATO & RAMALIGELA, 2019 apud FEITOSA et al., 2025).

Na preparação para exames, a utilização de *chatbots* inteligentes tem se mostrado eficaz no auxílio à resolução de exercícios de cálculo, otimizando o tempo necessário para a realização das questões e fortalecendo a confiança dos estudantes em suas habilidades matemáticas (JANČAŘÍK, MICHAL & NOVOTNÁ, 2023 apud FEITOSA et al., 2025).

Esses estudos ressaltam o potencial da IA na transformação do ensino da matemática, possibilitando abordagens mais eficazes, personalizadas e acessíveis. A incorporação de ferramentas inovadoras e recursos adaptativos contribui diretamente para o desenvolvimento de competências matemáticas essenciais para o êxito acadêmico e

profissional dos estudantes (QIU, PAN & ISHAK, 2022; DABINGAYA, 2022 apud FEITOSA et al., 2025).

ADIGUZEL, KAYA & CANSU (2023 apud FEITOSA et al., 2025) apontam o crescimento do número de publicações acadêmicas sobre IA aplicada à educação ao longo dos últimos anos reforça seu impacto transformador no ensino e na aprendizagem. Em 2020, foram registrados 13 estudos sobre o tema, número que saltou para 37 em 2024, evidenciando o crescente interesse por soluções inovadoras que abordam desafios contemporâneos da educação, como a personalização da aprendizagem, a autonomia do estudante e a democratização do acesso ao conhecimento (ADIGUZEL, KAYA & CANSU, 2023 apud FEITOSA et al., 2025).

Esse aumento reflete o reconhecimento da IA como uma ferramenta promissora para o aprimoramento do ensino e da aprendizagem matemática, especialmente diante de desafios como a desmotivação dos alunos, as dificuldades na abstração de conceitos e a escassez de recursos didáticos apropriados (QIU, PAN & ISHAK, 2022 apud FEITOSA et al., 2025).

Apesar do potencial significativo da IA na educação, Warschauer (2003) e Robinson (2020 apud FEITOSA et al., 2025) levantam alguns desafios que devem ser considerados para sua implementação eficaz: questões éticas relacionadas à coleta e ao uso de dados dos alunos, à privacidade e à autonomia individual demandam atenção e soluções apropriadas. Além disso, o acesso à tecnologia, especialmente em regiões com infraestrutura limitada, configura-se como um obstáculo, requerendo políticas públicas que promovam a inclusão digital e o acesso equitativo às ferramentas de IA (WARSCHAUER, 2003; ROBINSON, 2020 apud FEITOSA et al., 2025).

4. Metodologia

Os modelos de linguagem da coleção Llama 3.2, escolhido para este trabalho, representam uma nova geração de modelos multilingues treinados e ajustados para tarefas de geração de texto. Essa família de modelos inclui versões de 1B e 3B parâmetros, sendo otimizadas para diálogos em múltiplos idiomas e aplicações específicas, como sistemas que buscam e organizam informações de forma autônoma, sem depender de comandos diretos do usuário e sumarização (HUGGING FACE, 2025; META, 2025). De acordo com benchmarks da indústria, esses modelos apresentam desempenho superior em comparação a diversas alternativas de código aberto e fechadas (HUGGING FACE, 2025; META, 2025).

Na documentação no Hugginface (2025) e no META (2025), no que se refere à sua estrutura, o Llama 3.2 opera como um modelo auto-regressivo, baseado em uma arquitetura transformadora otimizada, ou *optimized transformer architecture*. Os modelos ajustados utilizam afinamento supervisionado, ou *supervised fine-tuning* (SFT), e aprendizado por reforço com *feedback* humano (RLHF) para melhorar a utilidade e a segurança das interações (HUGGING FACE, 2025; META, 2025). Além disso, todas as versões do modelo empregam a técnica de *Grouped-Query Attention* (GQA), o que aprimora a escalabilidade na inferência. No caso da versão 3.2, o GQA foi definido em 128k (HUGGING FACE, 2025; META, 2025).

O Llama 3.2 foi desenvolvido para uso comercial e acadêmico em diversos idiomas, incluindo inglês, alemão, francês, italiano, português, hindi, espanhol e tailandês. Os modelos ajustados por instrução são particularmente adequados para aplicações como assistentes conversacionais, ferramentas de escrita assistida por IA e reformulação de consultas. Já os modelos pré-treinados podem ser adaptados para uma ampla variedade de tarefas de geração de linguagem natural, enquanto versões quantizadas são viáveis para dispositivos com recursos computacionais limitados (HUGGING FACE, 2025; META, 2025).

O treinamento do modelo ocorre sobre um conjunto *offline* de dados públicos e sua licença é regida pelo Llama 3.2 *Community License* da Meta, um acordo comercial específico para sua utilização (HUGGING FACE, 2025; META, 2025). Embora seja uma versão estática, futuras atualizações podem ser lançadas para aprimorar sua segurança e capacidades (HUGGING FACE, 2025; META, 2025).

O Llama 3.2 baseia-se em um conjunto de dados públicos disponibilizados online com um corte de conhecimento estabelecido em dezembro de 2023. Os modelos foram treinados sobre um total de 9 trilhões de *tokens*, sua estrutura também suporta incorporações compartilhadas e contagem de *tokens*, garantindo versatilidade na geração de texto em diferentes modalidades linguísticas e contextuais (HUGGING FACE, 2025; META, 2025).

A escolha pela biblioteca *Transformers*, desenvolvida pela Hugging Face, foi devido ao seu objetivo ser facilitar a utilização de modelos de linguagem pré-treinados, tais como BERT, GPT e outros. Essa ferramenta oferece uma interface padronizada, permitindo o carregamento de modelos prontos e a execução de diversas tarefas de processamento de linguagem natural, incluindo classificação, geração de texto, tradução e resposta a perguntas (SANTOS, 2024).

A partir da versão 4.43.0 da biblioteca *Transformers*, é possível realizar inferência conversacional por meio da abstração do *Transformers* pipeline ou utilizando as classes Auto em conjunto com a função *generate*() (HUGGING FACE, 2025).

Já o PyTorch destaca-se como uma das principais ferramentas para o desenvolvimento de redes neurais profundas, proporcionando estruturas eficientes para a manipulação de tensores e a criação de modelos de aprendizado de máquina (SANTOS, 2024).

Para começar a usar o modelo Llama, é necessário fazer o cadastro no portal da Meta (2025; HUGGINGFACE, 2025) que disponibilizará gratuitamente os pesos e o tokenizador. Para isso, siga os seguintes passos descritos na documentação da Meta (2025) ou do Hugging Face (2025):

- Acesso ao site oficial: Visite o portal da Meta Llama.

- Leia e aceite os termos e condições de uso da licença.

- Após a aprovação da solicitação, uma URL assinada será enviada por e-mail.

- Inicie a instalação da ferramenta de linha de comando Llama executando no terminal:

$$pip\ install\ llama-stack$$

- Para listar os liste os modelos disponíveis para a sua instalação execute no terminal:

$$llama\ model\ list$$

- Caso necessite acessar versões anteriores, utilize no terminal:

$$llama\ model\ list\ --show-all$$

- Após identificar o modelo desejado, execute o seguinte comando para download, para nosso modelo foi usado a versão 3.2-1B-Instruct, caso deseje outra versão troque o modelo desejado pelo de sua escolha.

$$llama\ download\ --source\ meta\ --model-id\ 3.2-1B-Instruct$$

- Será solicitada a URL assinada que recebeu por e-mail, cole ela no terminal para iniciar o processo de transferência de dados.

Para operacionalizar o modelo Llama, é necessário instalar as dependências adequadas dentro de um ambiente Python apropriado, o venv. Para isso, execute:

$$pip\ install\ torch$$

Para utilizar o modelo com a biblioteca *transformers*, os pesos podem ser baixados e armazenados em cache por meio da instalação do pacote correspondente:

$$pip\ install\ llama-models$$

O modelo Llama 3.2 foi pré-treinado pela Meta (2025) utilizando um conjunto de dados de até 9 trilhões de tokens provenientes de fontes públicas. Durante o treinamento dos modelos Llama 3.2 de 1B e 3B, foram incorporados logits dos modelos Llama 3.1 de 8B e 70B na etapa inicial, utilizando os resultados desses modelos maiores como alvos em nível de token (HUGGING FACE, 2025). Após um processo de poda, a técnica de destilação de conhecimento foi empregada para recuperação do desempenho (HUGGING FACE, 2025).

Segundo a Meta (2025), a fase de pós-treinamento seguiu um processo semelhante ao do Llama 3.1, envolvendo múltiplas rodadas de alinhamento sobre o modelo pré-treinado. Cada etapa incluiu Fine-Tuning Supervisionado (SFT), Amostragem por Rejeição (RS) e Otimização Direta de Preferência (DPO), o conjunto de dados utilizado teve um corte temporal em dezembro de 2023 (HUGGING FACE, 2025).

O treinamento foi conduzido utilizando bibliotecas personalizadas e infraestrutura específica, incluindo um cluster de GPUs da Meta (2025) e um ambiente de produção dedicado, as etapas de ajuste fino, quantização, anotação e avaliação também foram realizadas dentro desse ambiente. Em termos de consumo energético, a Meta (2025) informa que o treinamento exigiu um total acumulado de 916 mil horas de GPU, utilizando hardware do tipo H100-80GB (com um consumo máximo de 700W por unidade). O tempo total de treinamento é determinado com base no uso cumulativo de GPUs, e o consumo energético foi ajustado considerando a eficiência do hardware (HUGGING FACE, 2025).

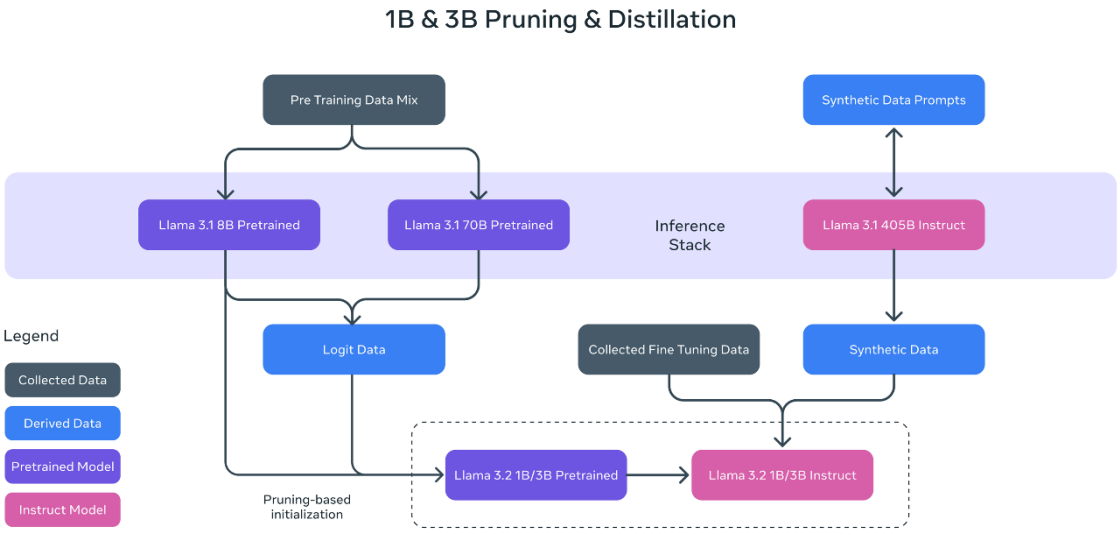

Figura 1 - Poda e destilação do modelo LLaMA 3.2

Fonte: Meta, 2025

O modelo foi otimizado para inferência utilizando a estrutura ExecuTorch do PyTorch e um *backend* da Arm CPU, com foco na qualidade do modelo, velocidade de preenchimento/decodificação e consumo de memória (HUGGING FACE, 2025). A estratégia de quantização desenvolvida é composta por três etapas principais (HUGGING FACE, 2025):

- Quantização dos pesos das camadas lineares dentro dos blocos do transformador para um esquema de 4 bits por grupo (tamanho do grupo = 32) e quantização dinâmica de ativação em 8 bits por *token* (HUGGING FACE, 2025).

- Aplicação de quantização em 8 bits por canal na camada de classificação para pesos, além de quantização dinâmica por *token* para ativação (HUGGING FACE, 2025).

- A camada de *embedding* segue um esquema semelhante ao da classificação, utilizando quantização de 8 bits por canal (HUGGING FACE, 2025).

A metodologia de coleta de dados combina informações geradas por humanos, por meio de fornecedores especializados, com dados sintéticos, visando reduzir potenciais riscos de segurança (META, 2025). Além disso, foram desenvolvidos classificadores baseados em grandes modelos de linguagem (LLMs) para aprimorar o controle de qualidade dos dados, permitindo a seleção criteriosa de *prompts* e respostas (HUGGING FACE, 2025).

5. Implementação

Para iniciar o *chatbot* corretamente é necessário iniciar o ambiente de desenvolvimento conforme os requisitos estabelecidos. Primeiramente, deve-se ativar o servidor ao vivo (*live server*) para o HTML, permitindo a visualização dinâmica e imediata das alterações realizadas no código. Esse procedimento viabiliza uma abordagem mais eficiente na construção da interface gráfica, ao proporcionar uma atualização automática da página sem necessidade de recarga manual. Dessa forma, desenvolvedores podem observar em tempo real os efeitos das modificações aplicadas, otimizando o fluxo de trabalho e garantindo maior precisão na implementação dos elementos visuais.

Além disso, é necessário iniciar o servidor da aplicação, que envolve a ativação do ambiente virtual e a execução da API via Flask. Para isso, deve-se primeiramente executar o comando:

```
.venv\Scripts\Activate.ps1
```

Esse comando configura o ambiente virtual e assegura um espaço isolado para a instalação de dependências específicas do projeto. Após essa etapa, o servidor pode ser iniciado através do comando:

```
flask --app servidor run
```

Este permitindo que a API esteja operacional e pronta para manipular requisições. Este processo garante a organização eficiente dos recursos computacionais, além de viabilizar uma estrutura modular e escalável para o desenvolvimento da aplicação.

Com o ambiente rodando, podemos explicar como o desenvolvimento do projeto foi conduzido. Inicialmente o *front-end*, escolhemos a implementação de uma interface web baseada em HTML (HyperText Markup Language) e CSS (Cascading Style Sheets), tecnologias fundamentais para a construção de aplicações front-end. O HTML foi utilizado como a estrutura principal do documento, permitindo a organização semântica dos elementos. Paralelamente, o CSS foi empregado para estilização, possibilitando o aprimoramento visual. Juntas, essas tecnologias asseguram a separação entre estrutura e apresentação, promovendo uma experiência de usuário otimizada.

O design do projeto foi concebido com uma estética simples e funcional, priorizando a clareza e a facilidade de navegação. A escolha de uma abordagem minimalista permitiu a organização intuitiva dos elementos visuais, reduzindo a presença de detalhes excessivos e focando na legibilidade e na interação objetiva do usuário com a interface. Dessa forma, os componentes gráficos foram posicionados de maneira direta, sem distrações ou sobrecarga de informações.

Além da simplicidade, a definição das cores seguiu um critério de contraste adequado, garantindo uma experiência visual confortável e eficiente. As tonalidades foram selecionadas para facilitar a leitura dos textos e a diferenciação entre seções distintas da interface, permitindo que os usuários encontrem informações com rapidez. O contraste entre fundo e elementos gráficos foi pensado para oferecer boa visibilidade, evitando combinações que possam comprometer a legibilidade.

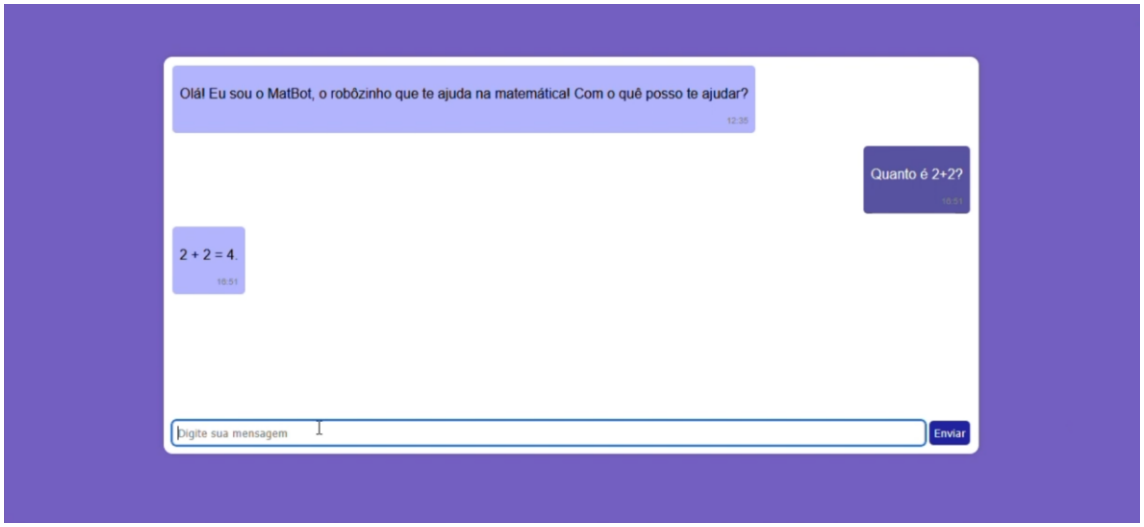

Figura 2 - Front-end do chatbot MatBot
Fonte: elaborado pelas autoras, 2025

A implementação de JavaScript no projeto foi fundamental para garantir a interatividade na funcionalidade de envio de mensagens. Por meio desse recurso, foi possível capturar o texto digitado pelo usuário em uma caixa de entrada e enviá-lo ao sistema ao clicar em um botão de envio. O código JavaScript foi responsável por ouvir eventos de interação, como o clique no botão ou a tecla "Enter", garantindo que a mensagem fosse processada corretamente.

Para estruturar essa funcionalidade, utilizou-se a manipulação do DOM (*Document Object Model*), permitindo acessar e modificar elementos da página em tempo real. O *script* recupera o valor da caixa de texto e o exibe na interface, possibilitando a comunicação entre usuário e sistema sem a necessidade de recarregamento da página. Além disso, foram implementadas verificações para evitar o envio de mensagens vazias, proporcionando uma experiência mais fluida e intuitiva.

```javascript
async function getResposta(chatBox, messageText) {

    console.log("1");

    await fetch("http://127.0.0.1:5000/enviaMensagem",{
        method: "POST",
        body: JSON.stringify({Mensagem:messageText}),
        headers: {"Content-Type":"application/json"}
    }).then((response) => {
        console.log("2");
        if (response.ok) {
            console.log("4");
            return response.json();
        } else {
            console.log("3");
            throw new Error("Erro na resposta do servidor");
        }
    }).then((data) => {
            const resposta = data.resultado;
        let newResponse = document.createElement("div");
        newResponse.classList.add("message", "received");
        newResponse.innerHTML = `<p>${resposta}</p><span class="timestamp">${new Date().toLocaleTimeString([], {hour: '2-digit', minute: '2-digit'})}</span>`;
        chatBox.appendChild(newResponse);
        chatBox.scrollTop = chatBox.scrollHeight;

    }).catch((error) => {
            console.error("Erro:", error);
    });
}

function sendMessage() {
    const inputField = document.getElementById("user-input");
    const messageText = inputField.value;

    if (messageText.trim() === "") return;

    const chatBox = document.querySelector(".chat-box");
    const newMessage = document.createElement("div");
    newMessage.classList.add("message", "sent");
    newMessage.innerHTML = `<p>${messageText}</p><span class="timestamp">${new Date().toLocaleTimeString([], {hour: '2-digit', minute: '2-digit'})}</span>`;
    chatBox.appendChild(newMessage);

    getResposta(chatBox, messageText);

    inputField.value = "";
    chatBox.scrollTop = chatBox.scrollHeight;

}
```

Figura 3 - Código JavaScript
Fonte: elaborado pelas autoras, 2025

Outro aspecto relevante foi o uso de requisições assíncronas, via Fetch API, permitindo que as mensagens fossem enviadas para um servidor de forma eficiente. Esse processo assegura que os dados sejam transmitidos sem interrupções na interface, garantindo um fluxo contínuo de interação entre o usuário e a aplicação. Assim, o JavaScript desempenhou um papel central na automação e otimização da funcionalidade de envio de mensagens, contribuindo para a dinâmica da plataforma.

O projeto utilizou Flask, um microframework em Python, para a criação de uma API destinada ao processamento e gerenciamento das mensagens enviadas pelos usuários. A API foi desenvolvida para receber os dados da interface *front-end*, processá-los e fornecer respostas adequadas conforme as requisições realizadas. A simplicidade e flexibilidade do Flask permitiram uma implementação eficiente, garantindo uma comunicação ágil entre o sistema e os usuários.

Dentro da arquitetura do projeto, a API Flask foi estruturada utilizando rotas HTTP, possibilitando o envio e recebimento de informações por meio de métodos como POST e GET. O método POST foi empregado com a função sendMessage() para receber mensagens do usuário e processá-las, enquanto o método GET, com a função getResposta(), chama a API e cria o diálogo com o *back-end* do *chatbot.* Na conexão com o *back-end,* o método GET presente na função getBotResponse() interage com o código LLM e retorna a resposta do *bot* para ser entregue ao usuário através da função getResponse(). Para manter um fluxo seguro e organizado, foram implementadas validações de entrada e resposta, reduzindo a possibilidade de erros ou inconsistências na comunicação.

O uso de Flask no projeto viabilizou um ambiente ágil e modular, facilitando a manutenção e futuras expansões do sistema, promovendo escalabilidade sem comprometer o desempenho.

```
1     from flask import Flask, request
2     from chatbot import getBotResponse
3     from flask_cors import CORS
4
5     app=Flask(__name__)
6     CORS(app)
7     @app.route("/")
8     def home ():
9         return "hello world"
10
11    @app.route("/enviaMensagem",methods=["POST"])
12    def enviaMensagem ():
13        if request.method=="POST":
14           resposta = getBotResponse(request.get_json(force=True).get("Mensagem"))
15           return {"resultado":resposta}
```

Figura 4 - API Flask
Fonte: elaborado pelas autoras, 2025

O *back-end* do *chatbot* desenvolvido neste projeto utiliza Python, aliado a bibliotecas Llama, Torch e Transformers explicadas anteriormente. Elas foram escolhidas por proporcionarem interações inteligentes e naturalizadas com os usuários. A implementação dessas tecnologias permitiu a criação de um sistema capaz de processar consultas e gerar respostas coerentes, garantindo uma experiência fluida e envolvente. Com uma base sólida de modelos de linguagem, o *chatbot* consegue interpretar comandos e estruturar respostas precisas, ajustadas ao contexto da conversa.

A integração do Llama ao sistema se mostrou essencial para garantir a qualidade da geração de texto. Como modelo de linguagem pré-treinado, ele oferece uma base robusta para processamento de texto, permitindo que o *chatbot* compreenda as entradas do usuário e produza respostas contextualizadas. Com o objetivo de otimizar o desempenho do *chatbot*, o projeto incorporou o uso do Torch, um *framework* para manipulação de tensores e execução eficiente de operações matemáticas em GPU. Essa abordagem permitiu acelerar o processamento das inferências realizadas pelo modelo de linguagem, reduzindo a latência das respostas e tornando a interação mais dinâmica. O Torch otimizou a implementação de cálculos complexos que garantindo uma melhor fluidez das conversas, viabilizando um sistema ágil e responsivo.

Além disso, a biblioteca Transformers, da Hugging Face, foi empregada para facilitar a implementação e gerenciamento do modelo de linguagem. Essa ferramenta possibilitou o carregamento e ajuste dos modelos pré-treinados, além de fornecer uma interface prática para a utilização de modelos como o Llama. A interação entre Transformers e Torch permitiu a execução otimizada das inferências, garantindo respostas rápidas e bem estruturadas.

O fluxo de funcionamento do *chatbot* segue uma lógica bem definida: o usuário envia uma mensagem, que é processada pelo sistema e repassada ao modelo de linguagem via API. O modelo, por sua vez, interpreta a solicitação, gera uma resposta adequada e a retorna para a interface, garantindo a comunicação entre usuário e *chatbot*. Esse ciclo acontece em tempo real, tornando as interações dinâmicas e permitindo que o sistema forneça informações conforme a necessidade do usuário.

Por fim, a integração entre Python, Llama, Torch e Transformers resultou em um *chatbot* robusto e eficiente, capaz de realizar interações de maneira fluida e coerente. A combinação dessas tecnologias assegurou que o sistema operasse com alto desempenho, mantendo a qualidade das respostas e a rapidez no processamento. Com uma abordagem estruturada e otimizada, o projeto alcançou um modelo funcional, pronto para atender às demandas dos usuários em um ambiente digital interativo.

```python
import torch
from transformers import pipeline

model_id = "meta-llama/Llama-3.2-1B-Instruct"

def getBotResponse (userInput):

    print ("1")

    messages = [
    {"role": "system", "content": "You are a chat meant to help users with Math."},
    {"role": "user", "content": userInput},
    ]

    print ("2")

    pipe = pipeline (
        "text-generation",
        model=model_id,
        torch_dtype=torch.bfloat16,
        device_map="auto",)

    print ("3")

    outputs = pipe(
        messages,
        max_new_tokens=64,
    )

    return (outputs[0]["generated_text"][-1]["content"])
```

Figura 5 - Função de buscar a resposta do bot
Fonte: elaborado pelas autoras, 2025

```python
from transformers import AutoTokenizer, Llama4ForConditionalGeneration
import torch

model_id = "meta-llama/Llama-4-Scout-17B-16E-Instruct"

tokenizer = AutoTokenizer.from_pretrained(model_id)

messages = [
    {"role": "user", "content": "Who are you?"},
]
inputs = tokenizer.apply_chat_template(
    messages, add_generation_prompt=True, return_tensors="pt", return_dict=True
)

model = Llama4ForConditionalGeneration.from_pretrained(
    model_id, device_map="auto", torch_dtype=torch.bfloat16
)

outputs = model.generate(**inputs.to(model.device), max_new_tokens=100)
outputs = tokenizer.batch_decode(outputs[:, inputs["input_ids"].shape[-1] :])
print(outputs[0])

#torchrun --nnodes=1 --nproc_per_node=8 inference.py
```

Figura 6 - Conexão com Llama
Fonte: elaborado pelas autoras, 2025

6. Conclusões e Trabalhos Futuros

Os *chatbots* tornaram-se essenciais na interação humano-computador, permitindo comunicação personalizada e adaptada aos usuários por meio da IA e do ML (Kuhn, 2020). Desde a criação do ELIZA por Joseph Weizenbaum, a evolução da IA levantou questões éticas e psicológicas sobre a capacidade das máquinas de simular interações humanas (SILVA, 2024). Com os avanços de modelos de linguagem como GPT-3 e GPT-4, a comunicação digital tornou-se mais sofisticada, impactando setores como atendimento ao cliente, publicidade e criação de conteúdo (SILVA, 2024). Além disso, o processamento de linguagem natural tem tornado interfaces digitais mais intuitivas e acessíveis, beneficiando não apenas interações comerciais, mas também aplicações educacionais por meio de tutores virtuais personalizados (FØLSTAD E BRANDTZAEG, 2017 apud SILVA, 2024).

O desenvolvimento dos *chatbots* está diretamente ligado ao avanço do aprendizado de máquina, que inclui abordagens supervisionadas e não supervisionadas, além de estratégias mais complexas como *deep learning* e aprendizagem por reforço

(SAIAS et al., 2018). Inicialmente, esses sistemas funcionavam com regras fixas, mas com aprimoramentos na mineração de dados e no processamento de linguagem natural, tornaram-se mais eficientes ao adaptar respostas com base na análise de dados históricos (LUGLI & LUCCA FILHO, 2020 apud SANTOS, 2025). Essa evolução possibilitou que *chatbots* fossem integrados a aplicativos de mensagens, proporcionando interações mais fluídas e personalizadas para os usuários (NETO et al., 2023 apud SANTOS, 2025).

Além dos avanços técnicos, a pesquisa destaca a relevância do modelo Llama, da Meta, que se diferencia de outras soluções de código aberto ao oferecer melhor integração com plataformas digitais e maior eficiência no desenvolvimento de redes neurais profundas (HYBL, 2024; SANTOS, 2024). Essa inovação, aliada ao uso de ferramentas como PyTorch e Transformers, contribui para aprimorar a interação conversacional, elevando a sofisticação dos *chatbots* modernos. Com esse progresso, as soluções baseadas em IA têm se expandido para diversos setores, otimizando serviços e automatizando processos com níveis mais elevados de precisão e contextualização (SANTOS, 2024).

O modelo de linguagem Llama 3.2 traz otimizações para diálogos em múltiplos idiomas e tarefas de geração de texto (HUGGING FACE, 2025; META, 2025). Com versões de 1B e 3B parâmetros, ele opera com uma arquitetura transformadora otimizada e utiliza técnicas como aprendizado por reforço com *feedback* humano (RLHF) e *Grouped-Query Attention* (GQA) para melhorar a inferência (HUGGING FACE, 2025; META, 2025). Desenvolvido para aplicações comerciais e acadêmicas, o modelo se destaca na sumarização e busca de informações autônomas, apresentando um desempenho superior em *benchmarks* da indústria quando comparado a alternativas abertas e fechadas (HUGGING FACE, 2025; META, 2025).

A metodologia de treinamento envolveu um conjunto de 9 trilhões de *tokens* com um corte temporal em dezembro de 2023, garantindo que o Llama 3.2 fosse adequado para contextos diversos (HUGGING FACE, 2025; META, 2025). Para a implementação, utiliza-se a biblioteca Transformers, que simplifica o uso de modelos pré-treinados, e o PyTorch, uma ferramenta essencial para a construção de redes neurais profundas (SANTOS, 2024). O processo de *fine-tuning* incluiu otimizações supervisionadas, técnicas de quantização e estratégias avançadas de alinhamento,

permitindo uma melhor adaptação do modelo às necessidades dos usuários (HUGGING FACE, 2025; META, 2025).

A operacionalização do Llama 3.2 requer o registro na Meta para o acesso aos pesos e ao *tokenizador*, além da instalação de pacotes específicos via Python (HUGGING FACE, 2025; META, 2025). O modelo foi otimizado para inferência utilizando a estrutura ExecuTorch do PyTorch e um *backend* baseado na CPU Arm, melhorando sua eficiência em diferentes dispositivos (HUGGING FACE, 2025).

O projeto foi desenvolvido com a implementação de uma interface *web* utilizando HTML e CSS, tecnologias essenciais para a estruturação e estilização de aplicações *front-end*. O HTML proporcionou a organização semântica dos elementos, enquanto o CSS aprimorou a apresentação visual, garantindo uma experiência de usuário otimizada. O design seguiu uma abordagem minimalista, focando na clareza e na facilidade de navegação, evitando excessos gráficos e garantindo legibilidade. A definição de cores foi realizada com base no contraste adequado para melhorar a acessibilidade e facilitar a identificação de informações na interface.

A funcionalidade de envio de mensagens foi integrada por meio da implementação de JavaScript, permitindo interatividade na comunicação entre usuário e sistema. Esse recurso possibilitou a captura e processamento do texto digitado sem a necessidade de recarregamento da página, utilizando manipulação do DOM e requisições assíncronas via Fetch API para garantir um fluxo contínuo de interação. Além disso, foram aplicadas verificações para evitar o envio de mensagens vazias, tornando a experiência mais intuitiva e eficiente.

O projeto também incorporou uma API desenvolvida com Flask, um micro*framework* em Python, para gerenciar as mensagens enviadas pelos usuários. A API foi estruturada com rotas HTTP, permitindo o envio e processamento de dados de forma ágil e segura (Meta, 2025). O *back-end* do *chatbot* foi construído com Python, utilizando bibliotecas como Llama, Torch e Transformers, que proporcionam interações naturais e eficientes na geração de respostas. A integração do modelo Llama foi fundamental para aprimorar a qualidade do texto gerado, enquanto o Torch contribuiu para acelerar a inferência e reduzir a latência nas respostas (HUGGING FACE, 2025; META, 2025).

Por fim, o fluxo de funcionamento do *chatbot* foi estruturado para garantir comunicação dinâmica e em tempo real entre usuário e sistema. O modelo de linguagem interpretou as solicitações enviadas e gerou respostas contextualizadas, proporcionando um alto desempenho na interação digital. Com a combinação de Python, Llama, Torch e

Transformers, o projeto resultou em um *chatbot* eficiente e responsivo, pronto para atender às demandas dos usuários e aprimorar a experiência interativa em plataformas digitais (HUGGING FACE, 2025; META, 2025).

Nosso projeto enfrentou desafios significativos relacionados ao processamento computacional exigido para sua execução eficiente. Para um modelo de larga escala, o Llama demanda uma infraestrutura robusta, especialmente no que se refere à capacidade de CPU e GPU. A inferência em tempo real e a necessidade de lidar com um alto volume de interações simultâneas podem resultar em latências elevadas caso os recursos computacionais sejam insuficientes (situação vivenciada durante o desenvolvimento).

Além disso, o treinamento do Llama exige uma grande quantidade de dados e processamento intensivo, o que pode limitar sua aplicabilidade em ambientes com restrições de *hardware* (META, 2025). A técnica de quantização utilizada para reduzir o consumo de memória e melhorar a eficiência da inferência é uma solução para mitigar esses desafios, mas ainda assim a demanda por processamento é um fator determinante para a implementação do modelo (META, 2025).

Para lidar com essas limitações, estratégias de otimização são fundamentais na utilização do Llama. No entanto, mesmo com o uso de técnicas de otimização, a necessidade de *hardware* especializado permanece uma barreira para a implementação escalável do modelo, exigindo um planejamento cuidadoso para maximizar sua eficiência dentro das limitações computacionais disponíveis (META, 2025).

No cenário atual, *chatbots* modernos desempenham um papel crucial na comunicação digital, atuando em áreas como atendimento ao cliente e suporte educacional (BATISTA & SOUZA, 2023 apud SANTOS, 2025). Sua capacidade de compreender padrões complexos e adaptar-se às necessidades dos usuários fortalece sua aplicação em diferentes contextos, ampliando a eficiência das interações automatizadas. Dessa forma, a evolução contínua dessas tecnologias reflete a busca por soluções mais inteligentes e acessíveis, promovendo experiências mais produtivas e dinâmicas na interação entre humanos e máquinas (LEONHARDT et al., 2003; MORAES; DE SOUZA, 2015 apud LARISANE KUYVEN et al., 2018).

A interação contínua entre usuários e *chatbots* possibilita melhorias constantes na precisão e eficácia da IA, já que o sistema ajusta seu funcionamento com base no *feedback* recebido, tornando-se cada vez mais adaptado às necessidades individuais dos usuários (BATISTA & SOUZA, 2023 apud SANTOS, 2025). Essa

tecnologia não apenas facilita a comunicação, mas também otimiza processos diários, tornando tarefas burocráticas mais ágeis e acessíveis (SANTOS, 2025). Diversos estudos investigam o desenvolvimento e aplicação dos *chatbots*, abrangendo desde agentes pedagógicos inteligentes até desafios na fluidez da conversação e análise de tecnologias premiadas (SOLIMAN & GUETL, 2010 apud LARISANE KUYVEN et al., 2018; SEWALD JUNIOR et al., 2011 apud LARISANE KUYVEN et al., 2018; BRADEŠKO & MLADENIĆ, 2012 apud LARISANE KUYVEN et al., 2018; DALE, 2017 apud LARISANE KUYVEN et al., 2018; BERNARDINI, SÔNEGO & POZZEBON, 2018, apud LARISANE KUYVEN et al., 2018). Além disso, pesquisas exploram sua utilização no ensino, destacando abordagens que incentivam a metacognição, a autorregulação e o aprendizado colaborativo (AGUIAR, TAROUCO & REATEGUI, 2011 apud KUYVEN et al., 2018; DYKE et al., 2013 apud KUYVEN et al., 2018).

Paralelamente, a IA tem impactado significativamente a educação, permitindo a personalização da aprendizagem, *feedback* em tempo real e adaptação às necessidades dos alunos (CHEN, CHEN & LIN, 2020 apud FEITOSA et al., 2025; TAPALOVA, ZHIYENBAYEVA & GURA, 2022 apud FEITOSA et al., 2025). No ensino da matemática, a colaboração entre especialistas das áreas de pedagogia e tecnologia é essencial para superar desafios na integração da IA ao ambiente educacional (HARDMAN, 2019 apud FEITOSA et al., 2025; HWANG & TU, 2021 apud FEITOSA et al., 2025). A literatura acadêmica evidencia benefícios da IA, como tutores virtuais inteligentes e ferramentas adaptativas que auxiliam na identificação de dificuldades dos alunos e personalizam conteúdos (ZAWACKI-RICHTER, 2019 apud FEITOSA et al., 2025; ZHANG & ASLAN, 2021 apud FEITOSA et al., 2025). Entretanto, desafios como privacidade, inclusão digital e acesso à tecnologia ainda precisam ser superados para que a IA possa promover mudanças efetivas e acessíveis no ensino e na aprendizagem matemática (WARSCHAUER, 2003 apud FEITOSA et al., 2025; ROBINSON, 2020 apud FEITOSA et al., 2025).

A utilização da IA no contexto educacional favorece o desenvolvimento do raciocínio lógico dos alunos, por meio da resolução de problemas interativos e desafiadores, adaptados ao ritmo individual de aprendizagem (SU, 2022 apud FEITOSA et al., 2025). Ferramentas de visualização e simulação também se mostram fundamentais para a compreensão de conceitos abstratos, tornando o aprendizado mais concreto, essa abordagem inovadora permite que os estudantes explorem diferentes soluções, desenvolvam estratégias autônomas de resolução e consolidem uma base sólida para o

sucesso na matemática (ASMUSS & BUDKINA, 2019 apud FEITOSA et al., 2025; VAERENBERGH & P'EREZ-SUAY, 2021 apud FEITOSA et al., 2025).

Assim, a implementação de um *chatbot* tutor voltado para o ensino de trigonometria, integrando tecnologias como *Deep Learning*, processamento de linguagem natural, sistemas tutores inteligentes e agentes conversacionais baseados no Watson (GLIOZZO et al., 2017), poderá beneficiar estudantes de cursos superiores de engenharia (LARISANE KUYVEN et al., 2018).

Com isso, há possibilidades de oportunidades de pesquisa e inovação, considerando que o número de estudos encontrados foi inferior às expectativas (LARISANE KUYVEN et al., 2018). A pesquisa voltada para aplicações educacionais revela-se mais complexa e recebe menor incentivo financeiro, ao passo que grande parte dos *chatbots* disponíveis na internet possuem propósitos comerciais, justificando o volume de investimentos e o retorno financeiro, no entanto, os trabalhos existentes apresentam qualidade e contribuem significativamente para a expansão do conhecimento na área (LARISANE KUYVEN et al., 2018). A análise dos artigos evidenciou que os principais fatores que dificultam a disseminação do uso de *chatbots* na educação incluem a necessidade de uma base de conhecimento robusta para interações satisfatórias e a maior complexidade e imprevisibilidade dos diálogos entre o agente e o aluno, se comparados às conversações casuais (LARISANE KUYVEN et al., 2018).

7. Referencia

FEITOSA, M. M., LEMOS, J. de J. S., NASCIMENTO, L., & MACHADO, A. M. B. (2025). *Inteligência Artificial e a Matemática: uma Revisão Sistemática de Literatura sobre Aplicações em Educação e Ensino*. **EaD Em Foco**, 15(1), e2410. Disponível em: https://doi.org/10.18264/eadf.v15i1.2410. Acesso em: 3 jun. 2025.

HUGGING FACE. *Llama 3.2-1B Instruct. Hugging Face, 2025.* Disponível em: https://huggingface.co/meta-llama/Llama-3.2-1B-Instruct. Acesso em: 19 de maio de 2025.

HYBL, Matous. *Comprehensive Question and Answer Generation with LLaMA 2*. **Tese (Master of Science in Computer Science).** Southern Adventist University. 2024. Acessado em https://knowledge.e.southern.edu/mscs_theses/2, em 24 de maio de 2025.

KUHN, Paloma. *Interação humano-computador mediada pela inteligência artificial: a experiência de usuário com assistentes pessoais virtuais.* **Monografia (Bacharel em**

Publicidade e Propaganda) - Universidade do Vale do Rio dos Sinos – UNISINOS. São Leopoldo. 2020.

LARISANE KUYVEN, N.; ANDRÉ ANTUNES, C.; JOÃO DE BARROS VANZIN, V.; LUIS TAVARES DA SILVA, J.; LOUREIRO KRASSMANN, A.; MARGARIDA ROCKENBACH TAROUCO, L. *Chatbots na educação: uma Revisão Sistemática da Literatura*. **RENOTE**, Porto Alegre, v. 16, n. 1, 2018. DOI: 10.22456/1679-1916.86019. Disponível em: https://seer.ufrgs.br/index.php/renote/article/view/86019. Acesso em: 3 jun. 2025.

META. *Llama 3*. Disponível em: https://www.llama.com/models/llama-3/Acesso em: 07 jun. 2025.

META. *Llama 3.2 Connect 2024: Vision, Edge e Mobile Devices*. Disponível em: https://ai.meta.com/blog/llama-3-2-connect-2024-vision-edge-mobile-devices/. Acesso em: 07 jun. 2025.

SAIAS, José; MAIA, Miguel; RATO, Luis; GONÇALVES, Teresa. *Machine Learning: um estudo sobre conceitos, tarefas e algoritmos relacionados com predição e recomendação*. Évora: Universidade de Évora, Dep. de Informática, ECT, 2018.

SANTOS, Alessandro Carvalho. *O uso de Large Language Models na melhoria do atendimento em sistemas socioassistenciais*. Cachoeiro de Itapemirim: Instituto Federal do Espírito Santo, 2024.

SANTOS, Isabela. *Automação com Inteligência Artificial: Análise de uma empresa provedora de plataforma de chatbot com IA*. **Trabalho de Conclusão de Curso (Graduação em Ciências Contábeis)** – Universidade Federal de Uberlândia, Uberlândia, 2025.

SILVA, Matheus Afonso Batista da. *Do Eliza ao ChatGPT: História e Evolução da Inteligência Artificial*. **Monografia (Bacharel em Ciências da Computação)** – Pontifícia Universidade Católica de Goiás. GOIÂNIA. 2024.

8. Apêndices

Github do projeto: https://github.com/MaryRCarvalho/MatBotLLM.git

Vídeo do chatbot funcionando:

https://github.com/MaryRCarvalho/MatBotLLM/blob/master/MatBot.mp4