Universidade Federal de Sergipe
Campus Prof. Alberto Carvalho
Departamento de Sistemas de Informação

Título do Projeto

Modelos de IA Generativa e a Prática da Engenharia de *Prompt*

Orientador: *Prof. Dr. Alcides Xavier Benicasa*
Aluno(s): Carlos Henrique Lima de Jesus e Charles Dayan da Conceição Costa

1 Introdução

Nos últimos anos, a Inteligência Artificial (IA) tem avançado rapidamente, transformando setores como saúde, educação, indústria e entretenimento. Entre seus ramos mais inovadores, a Inteligência Artificial Generativa (IAG) se destaca por sua capacidade de criar textos, imagens, códigos e outros conteúdos de forma autônoma.

Modelos como o GPT (do inglês, *Generative Pre-trained Transformer*), desenvolvido pela OpenAI, revolucionaram o Processamento de Linguagem Natural (PLN). Baseados na arquitetura *Transformer* proposta por Vaswani et al. (2017a), esses modelos geram respostas cada vez mais coerentes e contextualizadas. Estudos como o de Brown et al. (2020) demonstram que eles aprendem tarefas complexas com poucos exemplos, ampliando significativamente seu potencial de aplicação.

No entanto, a eficácia desses modelos não depende apenas de sua arquitetura. A forma como os *prompts* são estruturados influencia diretamente a qualidade das respostas geradas. A Engenharia de *Prompt* surge como um campo dedicado a otimizar essa interação (LIU et al., 2021). Apesar dos avanços, a ausência de padrões bem definidos e a necessidade de ajustes iterativos tornam a formulação de *prompts* um desafio, especialmente para usuários inexperientes (REYNOLDS; MCDONNELL, 2021).

Diante desse cenário, a sistematização de boas práticas na construção de *prompts* é essencial para garantir previsibilidade e controle sobre as respostas dos modelos de IA. Estudos recentes demonstram que a Engenharia de *Prompt* tem se consolidado como uma área estratégica para maximizar a eficiência dessas ferramentas.

O acesso a modelos de linguagem ocorre, predominantemente, por meio de *APIs* (do inglês, *Application Programming Interfaces*), interfaces conversacionais ou integrações em sistemas, que permitem a interação entre usuários e a inteligência artificial sem a necessidade de conhecimento técnico aprofundado sobre sua arquitetura interna. Contudo, a eficácia das respostas geradas por LLMs (do inglês, *Large Language Models*) não está restrita apenas à sua capacidade de processamento, mas também à qualidade e precisão das solicitações formuladas. Inclusive, de acordo com Liu et al. (2021), diferentes *prompts* aplicados a um mesmo modelo po-

dem resultar em respostas significativamente distintas, evidenciando a importância de estratégias que otimizem essa interação.

Embora existam diretrizes e exemplos de *prompts* previamente estabelecidos, sua aplicação nem sempre é suficiente para atender a demandas específicas. Isso ocorre porque as LLMs, apesar de sua capacidade de generalização, possuem limitações inerentes, como a dificuldade em interpretar contextos implícitos ou seguir instruções complexas sem orientação clara. A formulação de comandos precisa ser ajustada para garantir maior precisão, coerência e alinhamento com os objetivos desejados, o que demanda um processo iterativo de experimentação e refinamento (REYNOLDS; MCDONNELL, 2021).

A elaboração de novos *prompts* possibilita a personalização da comunicação com as LLMs, adaptando-as a tarefas e domínios específicos. Essa customização é essencial para superar as limitações dos modelos, conferindo maior controle sobre as respostas geradas, reduzindo ambiguidades e aumentando a previsibilidade dos resultados. Além disso, amplia a aplicabilidade das LLMs em diversas áreas do conhecimento, desde a geração de conteúdo até a análise de dados complexos. Estudos recentes destacam que a formulação de *prompts* influencia diretamente a eficácia dos modelos, reforçando a necessidade de estratégias bem definidas para sua construção (BROWN et al., 2020).

Diversos estudos têm explorado essas abordagens sob diferentes perspectivas. Andrade (2024), em um trabalho recente, investigou o uso de *prompts* para resolução de correferências em português, enquanto Gouveia (2024) propôs estratégias para extração estruturada de informações. Já Silva (2024) analisou como certas formulações de *prompt* podem induzir alucinações em modelos como ChatGPT e Gemini. Aplicações práticas também têm ganhado destaque, como no caso do Tribunal de Contas do Município do Rio de Janeiro (NASCIMENTO, 2024). Por fim, Bitelli (2024) comparou a engenharia de *prompt* com técnicas de *fine-tuning* na extração de entidades jurídicas, evidenciando os desafios da adaptação à língua portuguesa.

Portanto, a sistematização de boas práticas na criação de *prompts* não apenas potencializa a eficiência desses sistemas, mas também viabiliza sua utilização de maneira mais estratégica e confiável. Essa abordagem é fundamental para garantir que as LLMs atendam às demandas específicas de cada contexto, promovendo uma interação mais eficiente e confiável entre usuários e IA.

2 Objetivos do Projeto

Este projeto tem como foco principal a construção de *prompts* eficientes para modelos de linguagem de grande escala (LLMs), com ênfase em modelos amplamente acessíveis e populares na atualidade.

2.1 Objetivo geral

O objetivo geral deste projeto é desenvolver um *framework* prático e acessível de engenharia de *prompt* voltado para a extração, interpretação e simplificação de conteúdos especializados presentes em documentos PDF, utilizando modelos de linguagem de grande escala (LLMs), com foco em domínios técnicos e normativos. O objetivo é tornar o acesso a informações complexas mais ágil, acessível e confiável para usuários com diferentes níveis de familiaridade com o tema.

2.2 Objetivos específicos

Para alcançar o objetivo geral, foram estabelecidos os seguintes objetivos específicos:

- Investigar os fundamentos teóricos sobre modelos de IA generativa.
- Analisar boas práticas e estratégias de construção de *prompts* eficazes.
- Projetar e implementar um *framework* prático com *prompts* personalizáveis, voltados para a extração, interpretação e simplificação de informações em documentos PDF.
- Testar e validar o *framework* com usuários reais ou simulações práticas.
- Documentar os resultados obtidos e propor recomendações para o uso eficiente e ético de LLMs na análise de documentos complexos.

2.3 Relevância do Projeto

A capacidade de criar *prompts* eficientes é fundamental para maximizar o potencial dos LLMs, especialmente em cenários onde a precisão e a relevância das respostas são críticas. Ao fornecer diretrizes claras e ferramentas práticas, este projeto busca democratizar o acesso a essas tecnologias, permitindo que mais pessoas possam utilizá-las de forma estratégica e confiável. Além disso, a personalização de *prompts* permite que os usuários obtenham respostas mais direcionadas, adaptadas a suas necessidades específicas, em vez de depender de resultados genéricos disponíveis na internet.

3 Revisão Bibliográfica Inicial

Nesta seção será apresentada uma breve revisão teórica sobre os principais tópicos que têm relação com nosso trabalho, a fim de fornecer o embasamento necessário para compreender o que será proposto. A revisão bibliográfica foi dividida com os seguintes tópicos: Inteligência Artificial (IA) (3.1), Processamento de Linguagem Natural (PLN) (3.2), Modelos de Linguagem de Grande Escala (LLMs) (3.2.1) e Engenharia de *Prompt* (3.2.5).

3.1 Inteligência Artificial

A Inteligência Artificial (IA) é um campo multidisciplinar que busca desenvolver sistemas capazes de executar tarefas que normalmente exigiriam inteligência humana. Segundo Norvig e Russell (2013), a IA pode ser compreendida por diferentes abordagens, as quais podem ser classificadas em quatro categorias principais: sistemas que pensam como humanos, sistemas que agem como humanos, sistemas que pensam racionalmente e sistemas que agem racionalmente. Essas abordagens apresentam as diferentes perspectivas adotadas ao longo da história da IA, desde modelos que são baseados no pensamento humano até aqueles que priorizam a racionalidade matemática e computacional.

evitar abaixo, acima, etc.

Na imagem abaixo (Figura 1) são apresentadas as definições clássicas do campo, evidenciando como diferentes autores interpretam a IA. Enquanto algumas definições enfatizam a simulação do pensamento humano, outras focam mais no comportamento inteligente das máquinas.

Pensando como um humano	**Pensando racionalmente**
"O novo e interessante esforço para fazer os computadores pensarem (...) *máquinas com mentes*, no sentido total e literal." (Haugeland, 1985) "[Automatização de] atividades que associamos ao pensamento humano, atividades como a tomada de decisões, a resolução de problemas, o aprendizado..." (Bellman, 1978)	"O estudo das faculdades mentais pelo uso de modelos computacionais." (Charniak e McDermott, 1985) "O estudo das computações que tornam possível perceber, raciocinar e agir." (Winston, 1992)
Agindo como seres humanos	**Agindo racionalmente**
"A arte de criar máquinas que executam funções que exigem inteligência quando executadas por pessoas." (Kurzweil, 1990) "O estudo de como os computadores podem fazer tarefas que hoje são melhor desempenhadas pelas pessoas." (Rich and Knight, 1991)	"Inteligência Computacional é o estudo do projeto de agentes inteligentes." (Poole *et al.*, 1998) "AI... está relacionada a um desempenho inteligente de artefatos." (Nilsson, 1998)

Figura 1 – Algumas definições de inteligência artificial, organizadas em quatro categorias.

Fonte: Norvig e Russell (2013)

Com os avanços recentes na área, especialmente a partir do desenvolvimento de modelos de aprendizado profundo e de linguagem de grande escala, novas definições de inteligência artificial têm sido propostas para refletir a complexidade e o impacto social desses sistemas. Segundo Bommasani e al. (2022), a IA moderna, particularmente na forma de modelos fundacionais, deve ser compreendida como um conjunto de tecnologias amplamente aplicáveis, capazes de generalizar para múltiplas tarefas com base em grandes volumes de dados. Essa perspectiva amplia a compreensão tradicional da IA, destacando seu papel como infraestrutura central na ciência, na indústria e na sociedade contemporânea.

3.1.1 Aprendizado de Máquina

O aprendizado de máquina é o estudo de algoritmos computacionais que melhoram automaticamente através da experiência (MITCHELL, 1997).

O aprendizado de máquina é um dos principais pilares da IA, ele permite que os sistemas de computador melhorem continuamente seu desempenho à medida que são expostos a mais dados. Com o tempo, esses sistemas se ajustam e se aperfeiçoam, tornando-se mais eficientes para processar conjuntos de dados maiores e mais variados (MITCHELL, 1997).

Os principais tipos de Aprendizado de Máquina são: Aprendizado Supervisionado, Não Supervisionado e por Reforço (GOODFELLOW; BENGIO; COURVILLE, 2016):

- **Aprendizado Supervisionado:** o algoritmo aprende com exemplos rotulados, ou seja, ele recebe entradas com suas respostas corretas (rótulos) para aprender a classificar ou prever novos dados. Exemplos de aplicações incluem classificação de imagens (como distinguir gatos e cachorros), previsão de vendas, detecção de fraudes e diagnóstico médico.
- **Aprendizado Não Supervisionado:** o algoritmo trabalha com dados sem rótulos, buscando identificar padrões ou agrupamentos nos dados, como no caso de agrupamento (*clustering*), onde os dados são organizados em grupos semelhantes sem saber a que classe pertencem, o algoritmo é aplicável em segmentação de mercado, análise de sentimentos, redução de dimensionalidade e detecção de anomalias.
- **Aprendizado por Reforço:** o algoritmo aprende por tentativa e erro, recebendo *feedback* em forma de recompensa ou punição, sem saber a resposta correta previamente. É amplamente utilizado em jogos, robótica ou até mesmo em sistemas de recomendação.

Como um exemplo para estes tipos de aprendizado de máquina, podemos citar o trabalho de Leão et al. (2021), onde foram utilizadas técnicas de aprendizado supervisionado e não supervisionado para aprimorar a aprendizagem adaptativa. Algoritmos como Árvore de Decisão e Regressão Linear, do aprendizado supervisionado, ajudaram a prever e classificar o desempenho dos alunos, enquanto o *K-Means*, do aprendizado não supervisionado, foi usado para segmentar os alunos em grupos com características semelhantes. Possibilitando um ensino mais personalizado e adaptado às necessidades de cada aluno.

3.1.2 Redes Neurais

Uma rede neural é um modelo de aprendizado de máquina inspirado no cérebro humano, composto por camadas de neurônios artificiais. Ela identifica padrões em dados, aprende com exemplos e faz previsões ou classificações, ajustando seus parâmetros durante o treinamento para melhorar seu desempenho em várias tarefas (GOODFELLOW; BENGIO; COURVILLE, 2016).

Em sua forma mais geral, de acordo com Haykin (2001), uma rede neural é um sistema projetado para modelar a maneira como o cérebro realiza uma tarefa particular, sendo normalmente implementada utilizando-se componentes eletrônicos ou é simulada por propagação em

um computador digital. Para alcançarem bom desempenho, as redes neurais empregam uma interligação maciça de células computacionais simples, denominadas de "neurônios" ou unidades de processamento.

As redes neurais artificiais são comumente utilizadas na resolução de problemas complexos, onde o comportamento das variáveis não é rigorosamente conhecido. Uma de suas principais características é a capacidade de aprender por meio de exemplos e de generalizar a informação aprendida, gerando um modelo não-linear, tornando sua aplicação na análise espacial bastante eficiente (SPöRL; CASTRO; LUCHIARI, 2011).

O desenvolvimento e as aplicações das redes neurais artificiais se estendem por uma grande variedade de campos, não limitados a uma área específica. A razão para esta adoção cada vez mais frequente, inclusive como alternativa a modelos tradicionais, é o fato de elas modelaram a capacidade do cérebro humano (PALIWAL; KUMAR, 2009).

Na área da saúde, segundo o estudo de Barreto et al. (2018), as redes neurais foram utilizadas para analisar imagens de exames citológicos, identificando padrões que indicam a presença de câncer cervical (câncer de colo de útero). As redes são treinadas com dados de exames anteriores, permitindo a diferenciação entre células normais e anormais. Essa abordagem oferece maior precisão e rapidez no diagnóstico, superando métodos tradicionais e possibilitando uma maior automação na análise de exames, o que pode resultar em diagnósticos mais rápidos e acessíveis para a detecção precoce da doença.

Diante dessas características, as redes neurais artificiais apresentam grande relevância para o desenvolvimento deste trabalho, uma vez que oferecem mecanismos eficientes de aprendizagem e generalização, fundamentais para a construção de soluções baseadas em inteligência artificial. Sua capacidade de reconhecer padrões complexos a partir de dados torna-se especialmente útil no contexto proposto, contribuindo para a automação, precisão e adaptabilidade do sistema em desenvolvimento.

3.1.3 Deep Learning

De acordo com LeCun, Bengio e Hinton (2015) o *Deep Learning* utiliza redes neurais artificiais com múltiplas camadas que aprendem representações hierárquicas dos dados, permitindo avanços significativos em tarefas como reconhecimento de imagem, fala e linguagem. Esse método se aproxima do desempenho humano em diversas aplicações complexas.

Uma das principais características do *Deep Learning* é sua capacidade de aprender automaticamente representações úteis diretamente a partir dos dados brutos, sem a necessidade de definir previamente quais características devem ser extraídas. Isso é possível graças ao empilhamento de camadas não lineares em redes profundas, que capturam diferentes níveis de abstração (GOODFELLOW; BENGIO; COURVILLE, 2016). Essa característica tem tornado o *Deep Learning* muito eficaz em cenários onde os dados são abundantes e complexos, como no Processamento de Linguagem Natural (PLN).

Os progressos em *Deep Learning* só foram possíveis graças ao uso de unidades de

processamento gráfico (GPUs, do inglês, *Graphics Processing Unit*) de alto desempenho, que permitem executar muitos cálculos ao mesmo tempo. Essa capacidade de processamento paralelo é essencial para treinar redes neurais profundas com grandes volumes de dados (PANDEY et al., 2022).

Nas seções seguintes, serão apresentados os avanços nas arquiteturas modernas de *Deep Learning*, que são fundamentais para o desenvolvimento deste trabalho. Destacam-se, entre eles, os Modelos de Linguagem de Grande Escala (LLMs), capazes de processar sequências de texto de forma mais eficiente e precisa. Essas capacidades permitem que os modelos compreendam o contexto, gerem linguagem natural e realizem tarefas complexas, constituindo, assim, a base para a aplicação da Engenharia de *Prompt*.

3.2 Processamento de Linguagem Natural (PLN)

Segundo Caseli e Nunes (2024), o Processamento de Linguagem Natural (PLN) é um campo de pesquisa que tem como objetivo investigar e propor métodos e sistemas de processamento computacional da linguagem humana. O PLN pode ser dividido em duas grandes subáreas: a Interpretação de Linguagem Natural (NLU, do inglês, *Natural Language Understanding*) e a Geração de Linguagem Natural (NLG, do inglês, *Natural Language Generation*).

A NLU envolve a análise e compreensão da língua, focando em entender a intenção do usuário. Já a NLG se refere à criação de respostas em linguagem natural, como no caso dos *chatbots* (CASELI; NUNES, 2024). Essas duas subáreas são complementares e essenciais para o desenvolvimento de sistemas que interagem com a linguagem humana de forma eficiente.

O campo do PLN evoluiu significativamente desde os anos 1950. Na década de 1950, os primeiros sistemas baseados em regras, como o Georgetown-IBM *Experiment*, marcaram o início das pesquisas em PLN. Nos anos 1960, surgiram os primeiros *chatbots*, como o ELIZA, que simulava uma conversa com um terapeuta (JURAFSKY; MARTIN, 2009).

Além dos famosos *chatbots*, o PLN também está presente diversas aplicações como em motores de busca (Google, Bing), assistentes virtuais (Siri, Alexa), tradução automática, dentre outros. Tais aplicações demonstram a versatilidade e a importância do PLN em diversas áreas (JURAFSKY; MARTIN, 2025).

3.2.1 Modelos de IA Generativa

A IA Generativa (GenAI, do inglês, *Generative Artificial Intelligence*) refere-se a técnicas computacionais que são capazes de gerar novos conteúdos, como imagens, textos, vídeos e áudio, a partir de padrões aprendidos nos dados de treinamento (FEUERRIEGEL et al., 2023).

Entre as diversas abordagens da IA Generativa, os Modelos de Linguagem de Grande Escala (LLMs) se destacam por sua capacidade de processar e gerar textos coerentes e contextualmente relevantes. Conforme descrito por Abdullahi e Timonera (2024), os LLMs são um tipo

de modelo de IA que foi treinado por algoritmos de aprendizagem profunda para reconhecer, gerar, traduzir e/ou resumir grandes quantidades de linguagem humana escrita e dados textuais.

3.2.2 Arquiteturas Fundamentais de Modelos de IA Generativa

Os modelos de inteligência artificial generativa baseiam-se em arquiteturas específicas de redes neurais profundas, projetadas para lidar com tarefas complexas de geração de texto, imagem, áudio e outras modalidades. Nesta seção, são apresentadas as principais arquiteturas utilizadas em modelos de linguagem, com ênfase em seus componentes fundamentais, funcionamento e papel no desempenho de tarefas condicionadas por prompts (GOODFELLOW; BENGIO; COURVILLE, 2016).

Arquitetura *Transformer*

As inovações recentes no campo da inteligência artificial têm impulsionado o desenvolvimento de sistemas cada vez mais sofisticados, capazes de compreender e gerar linguagem natural com elevado nível de precisão (BOMMASANI; AL., 2022). No contexto das LLMs, destaca-se a utilização de redes neurais profundas estruturadas sobre a arquitetura *Transformer* (VASWANI et al., 2017a), que revolucionou a área ao permitir o processamento eficiente de sequências textuais longas e complexas.

Essas redes são projetadas para processar grandes volumes de dados textuais, identificando e aprendendo padrões linguísticos complexos. Dessa forma, ao receberem um *prompt* (entrada textual), são capazes de gerar respostas de maneira fluente e coerente, muitas vezes com qualidade comparável à de textos produzidos por seres humanos (VASWANI et al., 2017a).

A arquitetura *Transformer* foi introduzida por Vaswani et al. (2017b), vindo a revolucionar o PLN ao substituir as redes neurais recorrentes e convolucionais por um mecanismo de atenção. Esse modelo processa todas as palavras de uma sequência simultaneamente, em vez de analisá-las de forma sequencial, permitindo que o treinamento seja mais eficiente e tenha um melhor aproveitamento de paralelismo (VASWANI et al., 2017b). De acordo com os autores, o *Transformer* é composto por duas partes principais:

- ***Encoder***: que recebe a entrada e a transforma em representações numéricas, aplicando camadas de auto-atenção (*self-attention*) para entender as relações entre as palavras;
- ***Decoder***: que usa essas representações para gerar a saída, também utilizando mecanismos de atenção para focar nas partes mais relevantes da entrada durante a geração de texto.

A estrutura geral da arquitetura *Transformer* pode ser observada na imagem (Figura 2) abaixo:

evitar abaixo, acima, etc.

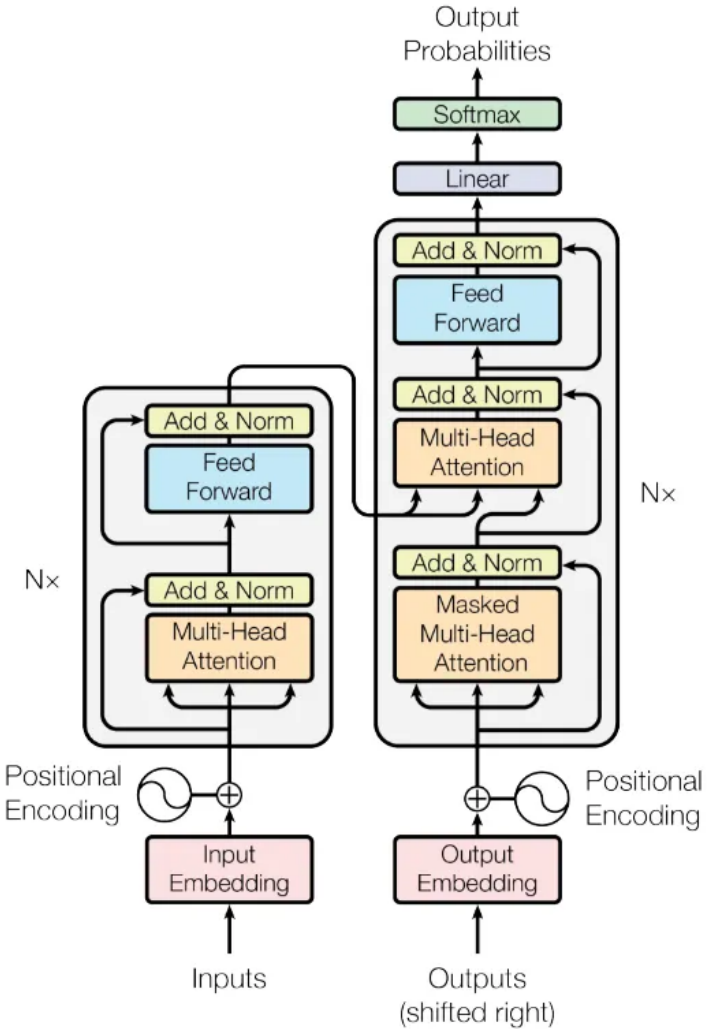

Figura 2 – Arquitetura Geral *Transformer*.

Fonte: Vaswani et al. (2017b)

A Figura 2 ilustra a estrutura completa da arquitetura *Transformer*, destacando os componentes internos do *encoder* e do *decoder*. Cada camada do codificador (à esquerda) é composta por dois blocos principais: o primeiro é uma atenção multi-cabeças (*Multi-Head Attention*), que permite ao modelo capturar diferentes relações entre as palavras da entrada; o segundo é uma rede *feed-forward*, que aplica transformações não lineares aos dados. Ambas as subcamadas utilizam conexões residuais seguidas por normalização (*Add & Norm*), o que contribui para a estabilidade e eficiência do treinamento (VASWANI et al., 2017b).

Já o decodificador (à direita) possui três subcamadas: uma atenção mascarada multi-cabeças, que impede que cada posição acesse informações futuras da sequência de saída; uma atenção multi-cabeças que considera as saídas do *encoder* para alinhar a geração de texto com a entrada; e uma rede *feed-forward*. Assim como no *encoder*, todas as subcamadas são acompanhadas de conexões residuais e normalização. A saída do decodificador é passada por uma camada linear seguida de uma função *softmax*, que gera as probabilidades de cada palavra no vocabulário (VASWANI et al., 2017b).

Essa organização da arquitetura mostra como o *Transformer* elimina a necessidade de recorrência, ao mesmo tempo em que garante aprendizado contextual e um paralelismo eficiente.

Essas inovações tornaram o *Transformer* a base para modelos modernos de linguagem, como GPT, LLaMA, DeepSeek, apresentados à seguir, dentre outros, estabelecendo assim um novo padrão no campo da IA (VASWANI et al., 2017a).

Arquitetura Autoregressiva *(Decoder-Only)*

A arquitetura autoregressiva baseada apenas em decodificador (*Decoder-Only*) é uma variação mais simplificada do modelo *Transformer* original proposto por Vaswani et al. (2017b), na qual apenas a parte do *decoder* é utilizada. Esse tipo de arquitetura é particularmente voltado para tarefas de geração de texto, como modelagem de linguagem, completamento de frases e diálogo, e tem sido amplamente adotado em modelos de LLMs, como o GPT (RADFORD et al., 2018; BROWN et al., 2020).

Assim como a versão completa do *Transformer*, a arquitetura *Decoder-Only* é composta por múltiplas camadas empilhadas, sendo que cada uma contém três componentes principais (VASWANI et al., 2017b):

- **Máscara de Atenção Multi-Cabeças (*Masked Multi-Head Attention*)**: impede que cada posição da sequência ”veja”as palavras futuras, preservando a ordem autoregressiva da linguagem.
- **Rede *Feed-Forward***: uma rede neural totalmente conectada aplicada de forma independente a cada posição da sequência, com função de ativação não linear.
- ***Add & Norm***: conexões residuais seguidas de normalização por camadas, o que facilita o treinamento profundo.

O funcionamento da arquitetura *Decoder-Only* é autoregressivo, o que significa que o modelo gera uma palavra por vez, sempre com base nas anteriores. Durante o treinamento, a sequência de entrada é deslocada (*shifted right*), e o modelo é treinado para prever o próximo *token* (VASWANI et al., 2017b).

Esse mecanismo torna a arquitetura ideal para tarefas de geração de texto contínua, onde cada palavra é condicionada à sequência anterior. O uso exclusivo do decodificador, combinado com a atenção mascarada, permite que o modelo seja amplamente escalável, como demonstrado no trabalho de Brown et al. (2020), que utiliza essa arquitetura para processar bilhões de parâmetros e gerar texto com coerência e fluidez.

Arquitetura *(Encoder-Only)*

A arquitetura *Encoder-Only* é uma adaptação da arquitetura *Transformer* proposta por Vaswani et al. (2017b), na qual apenas a parte do codificador (*encoder*) é utilizada. Esse modelo

é especialmente eficaz em tarefas de classificação de texto, análise de sentimentos, detecção de entidades e representação semântica, nas quais o principal objetivo é entender e codificar a entrada, e não gerar saídas sequenciais.

Essa abordagem ganhou certo destaque após o surgimento do modelo BERT (do inglês, *Bidirectional Encoder Representations from Transformer*), proposto por Devlin et al. (2019), que explora de forma bidirecional o contexto das palavras, permitindo que o modelo compreenda a frase como um todo, e não apenas da esquerda para a direita, como nos modelos autoregressivos.

A arquitetura *Encoder-Only* é composta por múltiplas camadas empilhadas de codificadores, onde cada camada possui dois blocos principais (DEVLIN et al., 2019):

- **Mecanismo de Atenção Multi-Cabeças (*Self-Attention*)**: permite que o modelo analise simultaneamente todas as palavras da entrada, capturando suas relações contextuais em diferentes níveis.
- **Rede *Feed-Forward***: realiza transformações não lineares em cada posição da sequência, enriquecendo a representação semântica dos *tokens*.

  Ambos os blocos são acompanhados por conexões residuais e camadas de normalização (*Add* & *Norm*), que estabilizam o treinamento.

Diferente da arquitetura *Decoder-Only*, o *Encoder-Only* não gera texto, mas sim representações vetoriais ricas da entrada textual. Cada *token* da sequência é convertido em um vetor contextualizado, levando em conta tanto os elementos anteriores quanto os posteriores da frase. Esse processo bidirecional é possível porque, ao contrário dos modelos autoregressivos, o *encoder* pode acessar toda a sequência ao mesmo tempo (DEVLIN et al., 2019).

Ainda segundo Devlin et al. (2019), essas representações são utilizadas como entrada para uma camada de classificação, tokenização de tarefas específicas, ou como vetores de *embeddings* em sistemas maiores.

Arquiteturas Multimodais

As Arquiteturas Multimodais são uma evolução dos modelos de linguagem, pois permitem que eles entendam diferentes tipos de informações, como texto, imagens, áudio e vídeo, ao mesmo tempo. Com isso, esses modelos se tornam mais completos e capazes de realizar tarefas variadas, indo além da geração e compreensão apenas de texto (LI et al., 2023).

De acordo com Li et al. (2023), os modelos multimodais geralmente têm os seguintes componentes principais:

- ***Encoders* especializados para cada modalidade**: cada tipo de dado (texto, imagem, áudio, etc.) é inicialmente processado por um *encoder* específico que transforma esse dado em uma representação vetorial.

- **Espaço Latente Compartilhado**: após a codificação, os diferentes vetores são projetados em um espaço semântico comum, permitindo que as informações de diferentes modalidades sejam alinhadas e correlacionadas.
- **Mecanismos de Atenção Cruzada (*Cross-Attention*)**: permitem que uma modalidade consulte informações relevantes em outra, melhorando a integração dos dados.
- ***Decoders* ou Cabeças de Saída**: dependendo da tarefa (geração de texto, legenda para imagens, descrição de vídeos, resposta auditiva, etc.), o modelo utiliza os *decoders* adequados para gerar saídas na modalidade desejada.

O funcionamento das arquiteturas multimodais consiste em, inicialmente transformar cada tipo de entrada como texto, imagem, áudio ou vídeo em representações numéricas através de *encoders*. Em seguida essas informações são combinadas em um espaço compartilhado, onde o modelo entende as relações entre elas usando atenção cruzada (*Cross-Attention*). No final o modelo gera uma saída na modalidade desejada, como texto, descrição de imagens ou respostas visuais (LI et al., 2023).

Inovações Recentes nas Arquiteturas

Nas últimas gerações de modelos de IA, especialmente de modelos de linguagem, diversas inovações arquiteturais têm sido propostas para melhorar desempenho, eficiência e capacidade multimodal. Uma das principais tendências é a adoção de arquiteturas unificadas e mais flexíveis, capazes de lidar com múltiplas modalidades, como texto, imagens, áudio e vídeo, dentro de um mesmo modelo (BOMMASANI; AL., 2022).

Entre as inovações, se destaca a introdução de mecanismos mais eficientes de atenção, como a Atenção Escassa (*Sparse Attention*), que reduz o custo computacional, focando apenas nas partes mais relevantes da sequência, e a Atenção Linear (*Linear Attention*), que busca tornar o processamento mais escalável em sequências longas (FEDUS; ZOPH; SHAZEER, 2022).

Além disso, surgiram arquiteturas híbridas, que combinam as características de *Transformers* com outras abordagens, como redes convolucionais ou recorrentes, a fim de melhorar a eficiência ou o desempenho em tarefas específicas (DOSOVITSKIY et al., 2021).

Há também avanços no uso de técnicas de aprendizado mais eficientes, como MoE (do inglês, *Mixture of Experts*), que permite ativar apenas partes do modelo conforme a necessidade, reduzindo custos computacionais sem perder desempenho (FEDUS; ZOPH; SHAZEER, 2022).

É notável que existe uma grande tendência na área, que é tornar os modelos não somente maiores, mas deixá-los mais eficientes, versáteis e capazes de trabalhar de maneira mais alinhada com as várias formas de comunicação humana.

3.2.3 Principais Modelos de IA Generativa e Suas Arquiteturas

Nos últimos anos, os avanços em IA resultaram na criação de diversos modelos generativos de grande impacto. Estes modelos são capazes de produzir textos, imagens, códigos, entre outros conteúdos, com alta qualidade e coerência. A seguir, são apresentados os principais modelos de IA Generativa, destacando-se suas arquiteturas e características específicas.

GPT

O GPT, criado pela OpenAI, é um modelo de linguagem baseado na arquitetura *Transformer*. O modelo é treinado de forma não supervisionada e consegue gerar respostas coerentes, realizar traduções e até mesmo criar código (BROWN et al., 2020).

O GPT-3, lançado em 2020, possui 175 bilhões de parâmetros e se destacou por sua capacidade de gerar textos muito semelhantes à escrita humana, superando seu antecessor, o GPT-2, que tinha apenas 1,5 bilhão de parâmetros (BROWN et al., 2020).

Em 2023, foi introduzido o GPT-4, representando um avanço significativo em relação ao seu antecessor, o GPT-3. Embora a OpenAI não tenha divulgado o número exato de parâmetros, estima-se que o GPT-4 tenha mais de 1 trilhão de parâmetros, além de apresentar melhorias na compreensão contextual, tradução, raciocínio lógico e capacidades multimodais, sendo capaz de processar entradas de texto e imagem (BAKTASH; DAWODI, 2023).

O GPT-4 apresenta avanços significativos em sua capacidade de realizar tarefas complexas a partir de poucos exemplos ou apenas de instruções, o que demonstra uma capacidade avançada de adaptação a diferentes contextos.

As inovações do GPT-4, como a capacidade de compreender contexto, gerar linguagem natural de alta qualidade e realizar tarefas com poucos exemplos, ampliaram sua aplicação em diversas áreas (OpenAI et al., 2024). Além do próprio ChatGPT, que é o assistente virtual da OpenAI, o GPT-4 tem sido utilizado em outros assistentes virtuais, como o Microsoft Copilot (SPATARO, 2023).

Na educação, o GPT-4 é usado no Duolingo Max, uma versão *premium* do *app* para aprender idiomas, onde o modelo auxilia oferecendo explicações e permitindo praticar conversas com personagens virtuais (DUOLINGO, 2023).

O modelo também está presente no Khanmigo, um tutor virtual da Khan Academy, ajudando estudantes com explicações detalhadas, oferecendo *feedback* em exercícios e auxiliando na prática de habilidades como redação e programação, tornando o aprendizado mais interativo e adaptado às necessidade dos alunos (OPENAI, 2023).

Adicionalmente, o GPT-4 tem sido aplicado na área da saúde para apoio ao diagnóstico, educação médica e interpretação de casos complexos. Em neuro-oftalmologia, por exemplo, o modelo alcançou 82% de precisão no diagnóstico de doenças, mostrando potencial como ferramenta auxiliar para profissionais da área (MADADI et al., 2023). Já em cardiologia, o GPT-4 atingiu 92% de acerto em questões complexas, superando estudantes de medicina e

indicando utilidade tanto para educação quanto para suporte clínico (HARIRI, 2023). Apesar dos avanços, a supervisão humana continua essencial para garantir a segurança e confiabilidade dos resultados.

LLaMA

O LLaMA (do inglês, *Large Language Model Meta AI*), desenvolvido pela Meta, é um modelo de linguagem também baseado na arquitetura *Transformer*, projetado para gerar texto e realizar tarefas avançadas de PLN. Treinado em grandes volumes de dados, o LLaMA aprende padrões linguísticos e gera respostas coerentes em diversas situações (TOUVRON et al., 2023).

O LLaMA foi treinado apenas com dados de domínio público, permitindo assim uma maior transparência e reprodutibilidade na pesquisa com modelos de linguagem. A proposta inicial do LLaMA era democratizar o acesso a grandes modelos de linguagem, disponibilizando-os para a comunidade científica para fins de pesquisa (TOUVRON et al., 2023).

Com modelos variando de 7 bilhões a 65 bilhões de parâmetros, o LLaMA oferece um equilíbrio entre desempenho e eficiência computacional, sendo mais acessível que outros modelos de grande escala. Ele é otimizado para ser eficaz com menos recursos, mantendo assim uma boa performance em tarefas como tradução, sumarização e resposta a perguntas (TOUVRON et al., 2023).

De acordo com Touvron et al. (2023), por ser um modelo de código aberto, o LLaMA se destaca como uma alternativa mais acessível para a comunidade acadêmica e para desenvolvedores, permitindo estudo, adaptação e personalização sem os custos de soluções proprietárias.

As inovações trazidas pelo LLaMA, como a sua arquitetura aberta e facilidade de adaptação, têm ampliado seu uso em várias áreas. Na medicina, o Me-LLaMA, uma versão do LLaMA-2 adaptada especialmente para entender textos médicos. O modelo foi treinado com vários dados clínicos e conseguiu resultados superiores ao ChatGPT em 7 de 8 testes diferentes, mostrando que pode ser uma ferramenta útil para ajudar profissionais da área da saúde (XIE et al., 2024).

O modelo PMC-LLaMA é uma versão personalizada do LLaMA-2, que foi treinada com dados do PubMed Central, um repositório de artigos científicos biomédicos. Ele foi desenvolvido para melhorar o desempenho em tarefas médicas, como responder perguntas sobre os conteúdos médicos. Em *benchmarks* médicos especializados como MedMCQA e PubMedQA, o PMC-LLaMA superou o ChatGPT e o LLaMA-2, se mostrando eficaz em compreender e gerar respostas precisas para questões médicas (WU et al., 2023).

Na área de programação, o Code LLaMA é uma versão do LLaMA treinada especificamente para entender e gerar código em várias linguagens de programação, como Python, Java e C++. Ele foi avaliado em diversos testes que medem a capacidade de completar, corrigir e explicar códigos, alcançando resultados muito próximos ou até mesmo superiores aos dos melhores modelos de código abertos disponíveis (ROZIèRE et al., 2024).

Outro exemplo do uso do LLaMA é o Meta AI, assistente virtual da Meta que foi integrado ao WhatsApp, Instagram e Facebook. Baseado no LLaMA 4, ele permite interações em linguagem natural, como responder perguntas, gerar textos e até mesmo imagens, além de realizar buscas em tempo real (META, 2025).

DeepSeek

O DeepSeek é um modelo de linguagem de código aberto desenvolvido pela empresa chinesa *Hangzhou DeepSeek Artificial Intelligence*. O modelo é projetado para lidar com tarefas complexas de processamento de linguagem natural (PLN), combinando a arquitetura *Transformer* com a técnica de *Mixture-of-Experts (MoE)* para melhorar a eficiência computacional e a performance em inferência (DEEPSEEK-AI, 2024).

Desde seu lançamento, o DeepSeek tem passado por uma evolução contínua. A versão inicial, DeepSeek-V1, foi lançada em 2023 e marcou o início do projeto com um modelo voltado a tarefas gerais de PLN. Em 2024, foi lançada a versão DeepSeek-V2, trazendo melhorias significativas, como a adoção da arquitetura MoE e a capacidade de lidar com múltiplas modalidades de entrada e tarefas de raciocínio (DeepSeek-AI et al., 2025).

A versão mais recente, DeepSeek-R1, lançada em 2025, introduziu o uso de aprendizado por reforço com feedback humano ou RLHF (do inglês, *Reinforcement Learning with Human Feedback*), incentivando capacidades de raciocínio simbólico e lógico. O modelo R1 possui um total de 236 bilhões de parâmetros, dos quais apenas uma fração é ativada para cada *token* processado, o que permite manter a eficiência computacional (DeepSeek-AI et al., 2025).

A arquitetura do DeepSeek é baseada no modelo *Transformer*, conforme proposto por Vaswani et al. (2017a), que se tornou a base de quase todos os modelos modernos de linguagem natural. A partir da versão V2, o DeepSeek passou a utilizar o mecanismo MoE, no qual partes específicas do modelo são ativadas dinamicamente de acordo com a entrada. Essa abordagem torna o modelo mais escalável, mantendo desempenho elevado com menor custo computacional. Na versão R1, esse sistema é complementado com RLHF, que permite adaptar o modelo com base em avaliações humanas e objetivos personalizados (DeepSeek-AI et al., 2025).

Entre suas principais características, o DeepSeek se destaca por ser um modelo de código aberto, o que facilita seu uso e personalização por pesquisadores e desenvolvedores. Ele oferece alto desempenho em tarefas que exigem raciocínio lógico, solução de problemas matemáticos e programação. A integração do RLHF proporciona ainda uma melhora significativa na qualidade das respostas geradas, especialmente em contextos complexos.

As aplicações do DeepSeek são amplas. O modelo tem sido utilizado para assistência à programação, resolução de problemas matemáticos, sistemas de diálogo e atendimento automatizado, além de servir como base para pesquisas em engenharia de *prompts* e desenvolvimento de novas arquiteturas em PLN. Sua acessibilidade e eficiência o tornam uma opção promissora tanto para aplicações acadêmicas quanto comerciais (DeepSeek-AI et al., 2025).

Gemini

O Gemini é uma família de modelos de linguagem desenvolvida pelo Google DeepMind, projetada para competir com os modelos mais avançados da atualidade, como o GPT-4 da OpenAI. Assim como os modelos da família GPT, o Gemini é baseado na arquitetura *Transformer*, sendo otimizado para tarefas multimodais, ou seja, capazes de compreender e gerar texto, código, imagem, áudio e vídeo em alguns casos (Google DeepMind, 2023a).

A primeira geração, Gemini 1, foi lançada em dezembro de 2023, e incluiu três versões principais: Gemini 1 Nano, Gemini 1 Pro e Gemini 1 Ultra. O modelo Gemini 1 Nano foi projetado para dispositivos móveis e embarcados; o Gemini 1 Pro, voltado para uso geral e disponível na versão gratuita do Bard (posteriormente renomeado para Gemini); e o Gemini 1 Ultra, o mais poderoso da geração, destinado a aplicações de alta complexidade e disponível apenas na versão paga, Gemini Advanced (Google DeepMind, 2023b).

Em fevereiro de 2024, foi lançada a linha Gemini 1.5, com destaque para o modelo Gemini 1.5 Pro. Essa versão trouxe avanços significativos em eficiência e capacidade de contexto, suportando janelas de contexto de até 1 milhão de *tokens*, com capacidade estável de desempenho em janelas de até 128 mil *tokens*, superando o limite da maioria das LLMs concorrentes (Google DeepMind, 2024). Isso permite ao modelo analisar grandes volumes de informação em uma única chamada, sendo ideal para tarefas como análise de documentos extensos, revisão de código ou integração com múltiplas fontes de dados.

Os modelos Gemini foram treinados com dados multimodais desde o início, incluindo texto, imagens, vídeos e código. Essa abordagem permitiu ao modelo desenvolver habilidades avançadas de raciocínio visual, entendimento de gráficos e tabelas, além de habilidades matemáticas e científicas mais robustas (Google DeepMind, 2023a). Em *benchmarks* acadêmicos como o MMLU, o Gemini 1.5 Pro demonstrou desempenho comparável ao GPT-4, evidenciando seu alto nível de competência geral.

O Gemini também se destaca por seu foco em segurança, alinhamento ético e mitigação de vieses. Segundo o relatório técnico, os modelos passaram por avaliações rigorosas quanto à toxicidade, alucinação e vulnerabilidades, com melhorias contínuas nos processos de *fine-tuning* e RLHF (Google DeepMind, 2023a).

Na prática, o Gemini tem sido utilizado em diversas aplicações: desde integração com os serviços do Google Workspace (como Gmail, Docs e Sheets) através do Gemini no Workspace, até suporte ao desenvolvimento de código com integração no Android Studio e no Colab, além de *chatbots* e assistentes integrados ao sistema Android.

Tais características tornam o Gemini uma das mais promissoras LLMs do mercado atual, com ampla capacidade de adaptação e aplicação em diferentes áreas, como educação, pesquisa científica, atendimento ao cliente, desenvolvimento de software e muito mais.

Claude

A Claude é uma família de modelos de linguagem desenvolvida pela empresa norte-americana Anthropic, fundada em 2021 por ex-pesquisadores da OpenAI. Esses modelos são projetados com ênfase na segurança, alinhamento ético e interpretabilidade, utilizando a arquitetura *Transformer* como base (ANTHROPIC, 2023). A nomenclatura "Claude"é uma homenagem ao matemático e engenheiro Claude Shannon, considerado o pai da teoria da informação.

O primeiro modelo, Claude 1, foi lançado em março de 2023 como resultado das pesquisas iniciais da Anthropic sobre aprendizado supervisionado constitucional, um método voltado ao alinhamento ético dos modelos. Em julho do mesmo ano, a Anthropic lançou o Claude 2, com avanços significativos em codificação, raciocínio e compreensão de linguagem natural, além de um aumento na janela de contexto para até 100.000 *tokens*. Essa evolução permitiu que o modelo mantivesse coerência em diálogos extensos e executasse tarefas mais sofisticadas (ANTHROPIC, 2023).

A terceira geração, Claude 3, lançada em março de 2024, consolidou a Claude como uma família de modelos, composta por três variantes: Haiku, Sonnet e Opus. Essa divisão permitiu atender diferentes demandas de mercado, equilibrando velocidade, custo computacional e desempenho. Todos os modelos da série Claude 3 são multimodais, ou seja, capazes de interpretar não apenas texto, mas também imagens e outros tipos de dados, com desempenho competitivo frente a modelos como GPT-4 e Gemini 1.5 (ANTHROPIC, 2024).

Em maio de 2025, foi anunciada a versão Claude 4, incorporando novas capacidades interativas, como raciocínio mais avançado, integração com ferramentas externas e melhorias na gestão da memória de longo prazo. Essas atualizações colocaram a Claude 4 entre os modelos mais avançados da atualidade, ampliando seu uso em aplicações como sistemas de diálogo, assistência à programação, tutores educacionais, análise de dados e pesquisa científica (ANTHROPIC, 2025).

A família Claude se destaca por seu compromisso com a segurança e a transparência, elementos centrais na filosofia da Anthropic, que propõe o uso de técnicas como *Constitutional AI* e auditoria contínua de comportamento como formas de mitigar riscos de uso indevido ou alucinação de conteúdo (ANTHROPIC, 2022). Essa abordagem posiciona a Claude como uma das alternativas mais robustas e responsáveis no cenário de modelos de linguagem de grande porte

Os modelos da família Claude são amplamente utilizados em ambientes profissionais que exigem alto grau de confiabilidade e controle, como empresas de tecnologia, setores jurídicos e instituições educacionais. Eles têm sido aplicados em tarefas como análise contratual, geração de relatórios técnicos, atendimento automatizado e suporte a desenvolvedores na escrita e correção de código. A versão Claude 3.5, por exemplo, vem sendo integrada a plataformas corporativas para oferecer assistência contextual com memória persistente, permitindo que equipes mantenham continuidade em projetos complexos (ANTHROPIC, 2025).

3.2.4 Comparativo das Arquiteturas e Capacidades

A presente seção apresenta um comparativo detalhado entre os principais modelos de linguagem de última geração, com base em atributos fundamentais para sua avaliação e aplicação prática. Para isso, foram considerados aspectos como desempenho em *benchmarks* padronizados de conhecimento geral, raciocínio matemático e codificação, além de suas capacidades multimodais, suporte a contextos longos, estratégias de alinhamento ético e grau de acessibilidade e licenciamento. A análise busca evidenciar os pontos fortes e limitações de cada arquitetura, oferecendo uma visão crítica que auxilie na escolha do modelo mais adequado para diferentes tipos de tarefas e contextos de uso. As informações foram sistematizadas nas Tabelas 1 e 2, a partir de dados atualizados extraídos das documentações oficiais e repositórios de avaliação.

Tabela 1 – Comparação geral entre os modelos

Modelo	**Multimodalidade e Contexto Longo**	**Alinhamento Ético**	**Acessibilidade e Licenciamento**
GPT-4o	Texto, código, imagem, áudio / 128.000 tokens	Reinforcement Learning from Human Feedback (RLHF)	API paga, não open-source
LLaMA 3	Texto, código / 8.192 tokens	Supervised Fine-Tuning (SFT) e Reinforcement Learning from Human Feedback (RLHF)	Gratuito com restrições, semi-aberto
Gemini 1.5 Pro	Texto, código, imagem, áudio e vídeo / 128.000 tokens	Reinforcement Learning from Human Feedback (RLHF) e Constitutional AI-like guidelines	API paga, fechado
Claude 3.5 Sonnet	Texto, código / 200.000 tokens	Constitutional AI	API paga, fechado
DeepSeek V2	Texto, código, imagem, áudio / 128.000 tokens	Supervised Fine-Tuning (SFT) e Reinforcement Learning (RL)	Código aberto, uso gratuito

Fontes: Elaborado pelos autores com base em informações das documentações oficiais dos modelos.

A Tabela 1 mostra os principais modelos de linguagem, destacando suas capacidades multimodais, janela de contexto (número de *tokens*) e alinhamento ético. O GPT-4o suporta texto, código, imagem e áudio, enquanto o Gemini 1.5 Pro se destaca por oferecer uma multimodalidade mais completa, incluindo texto, código, imagem, áudio e também vídeo, ambos têm janelas de contexto amplas de 128.000 *tokens*.

O Claude 3.5 Sonnet se destaca pela sua ampla janela de contexto de 200.000 *tokens*, ideal para lidar com textos muito longos. Já o LLaMA 3 foca em texto e código tendo um limite menor de 8.192 tokens, usando o RLHF (do inglês, Reinforcement Learning from Human Feedback) para garantir respostas de qualidade, porém com menos capacidade para contextos extensos.

No quesito alinhamento ético, todos os modelos utilizam técnicas atuais, como RLHF ou Constitutional AI, para reduzir vieses e garantir respostas mais seguras. Sobre o acesso, o DeepSeek V2 é *open-source* e gratuito, enquanto a maioria como GPT-4o, Gemini e Claude

funcionam via API paga e fechada. O LLaMA 3 tem um modelo semiaberto, dando mais liberdade para desenvolvedores, porém, tendo algumas limitações. Isso mostra diferentes formas de equilibrar controle, segurança e acesso aos LLMs.

Já na Tabela 2 foi feita uma comparação do desempenho de cinco modelos em conhecimento geral, matemática e codificação. O GPT-4o lidera em conhecimento geral e codificação, enquanto o Gemini 1.5 Pro tem a melhor pontuação em matemática. O DeepSeek V2 também apresenta bom desempenho em codificação, e o Claude 3.5 Sonnet mostra resultados equilibrados em todas as áreas. O LLaMA 3 tem desempenho um pouco mais modesto, mas ainda assim consistente, principalmente em conhecimento geral.

Tabela 2 – Comparação entre os modelos em alguns *benchmarks*

Modelo	**Conhecimento Geral (MMLU-Pro)**	**Matemática (MATH-500)**	**Codificação (HumanEval)**
GPT-4o	72,6%	75.2%	87.2%
LLaMA 3	52.8%	65%	72%
Gemini 1.5 Pro	69%	82.8%	79.3%
Claude 3.5 Sonnet	56.8%	72.5%	81.7%
DeepSeek V2	54.8%	82.8%	83.5%

Fontes: Wang et al. (2024), ValsAI (2025), Liu et al. (2023)

Esses resultados mostram que cada um dos modelos têm seus pontos fortes, o que permite escolher o mais adequado para diferentes tipos de tarefas, seja para resolver problemas matemáticos, gerar código ou responder perguntas sobre assuntos gerais.

3.2.5 O Papel dos Prompts na Interação com Modelos de IA Generativa

Com a crescente popularização da utilização de IA Generativa, como ChatGPT, DeepSeek, LLaMA, Gemini e Claude, surge a necessidade de não apenas fazer perguntas simples, mas de elaborar comandos precisos e bem estruturados para obter resultados mais satisfatórios e aprimorados. Esses comandos são chamados de *prompts*, e são essenciais para guiar a IA a gerar respostas mais alinhadas com as expectativas do usuário (WHITE et al., 2023).

A Engenharia de *Prompt* surge como um conjunto de técnicas e métodos para formular instruções (*prompts*) precisas e bem estruturadas para obter respostas ou informações desejadas de LLMs, otimizando a eficácia de interação com o modelo (KNOTH et al., 2024).

Funcionando como uma forma de programação, os *prompts* permitem personalizar as interações e adaptar instruções em linguagem natural, o que aumenta a eficácia dos modelos de IA em diversas aplicações. Nos últimos anos, observa-se uma tendência crescente na aplicação da engenharia de *prompts* em LLMs pré-treinados, visando solucionar tarefas de PLN com recursos limitados (YANG et al., 2022).

Segundo Saravia (2022), o *prompt* pode conter alguns elementos como por exemplo:

- **Instrução:** uma tarefa ou instrução específica que você deseja que o modelo execute;

- **Contexto:** informações externas ou contexto adicional que podem direcionar o modelo para melhores respostas;
- **Dados de entrada:** a entrada ou pergunta para a qual estamos interessados em encontrar uma resposta;
- **Indicador de saída:** O tipo ou formato esperado da resposta gerada pelo modelo.

Os *prompts* podem ser classificados em diferentes níveis, de 1 a 4, variando desde perguntas simples até comandos mais complexos que incluem exemplos, permitindo que o modelo de linguagem decomponha a solicitação em componentes específicos. Além disso, os *prompts* podem ser categorizados dependendo da abordagem utilizada como (WHITE et al., 2023):

- ***Prompts* instrutivos**: dão comandos explícitos como "Resuma este texto...", orientando diretamente a tarefa;
- ***Prompts* de sistema**: usados geralmente no início do *prompt* para definir o "papel" do modelo, seu estilo e limites (ex.: "Você é um assistente educacional..."");
- ***Prompts* de pergunta-resposta**: se baseiam em perguntas diretas do tipo "O que é tal coisa?", focando em obter uma resposta precisa;
- ***Prompts* mistos**: combinam elementos instrutivos, de sistema e perguntas para guiar respostas mais sofisticadas, fornecendo exemplos e contexto.

O desenvolvimento de *prompts* é um processo iterativo, que exige clareza, concisão e a não tenha complexidades desnecessárias no texto. Para alcançar melhores resultados, é fundamental aplicar as principais técnicas de engenharia de *prompt*, como (NASCIMENTO, 2024; SARAVIA, 2022; XU et al., 2023):

- ***Zero-shot Prompting:*** realiza perguntas ou instruções sem apresentar exemplos anteriores, confiando na habilidade do modelo de entender e gerar respostas adequadas.
- ***Few-shot Prompting:*** apresenta alguns exemplos prévios para orientar o modelo sobre o tipo de resposta desejada.
- ***Chain-of-Thought Prompting:*** incentiva o modelo a explicar o raciocínio passo a passo, dividindo problemas complexos em etapas lógicas para que a solução fique mais clara e estruturada.
- ***Three-of-Thought Prompting:*** é parecida com a anterior, mas ela incentiva que o modelo divida o raciocínio em três etapas principais para resolver problemas de forma clara e concisa, sem exagerar na complexidade.

- ***Iterative Prompting*:** refina progressivamente o *prompt* com base em *feedback* contínuo, ajustando-o para melhorar a precisão e a relevância das respostas fornecidas pelo modelo.
- ***Maieutic Prompting*:** estimula o modelo a chegar a respostas por meio de perguntas que promovem o pensamento crítico e a reflexão.
- ***Example-based Prompting*:** guia o modelo a seguir um formato ou estilo específico, sendo útil para tarefas que exigem consistência ou formatação.
- ***Negative/Positive Prompting*:** inclui diretrizes claras sobre conteúdos que devem ser incluídos ou excluídos nas respostas.
- ***Hybrid Prompting*:** combina diferentes técnicas de *prompt* para maximizar a eficácia.
- ***Contextualization Prompting*:** passa informações relevantes para o *prompt* para que o modelo entenda melhor o contexto.
- ***Role-Playing Prompting*:** cria cenários onde o modelo assume um papel ou persona específica para explorar diferentes abordagens em suas respostas.
- ***Adversarial Prompting*:** consiste em passar informações especificamente projetadas para confundir, induzir ao erro ou manipular as saídas do modelo.
- ***Expert Prompting*:** é a técnica de instruir o modelo a responder como um especialista em determinada área, utilizando linguagem e raciocínio técnico.

A engenharia de *prompt* desempenha um papel central na eficácia das interações com modelos de LLMs, sendo considerada uma habilidade essencial na era da IA. Segundo Liu et al. (2021), a formulação adequada de *prompts* pode melhorar significativamente o desempenho dos modelos em tarefas complexas.

Isso acontece porque LLMs são altamente sensíveis à forma e estrutura das instruções recebidas, o que torna a clareza, o contexto e a especificidade dos *prompts* determinantes para a qualidade das respostas (WEI et al., 2023). Além disso, a engenharia de *prompt* permite que pessoas sem conhecimento técnico utilizem modelos de IA de forma eficaz, tornando o uso dessa tecnologia mais acessível a todos (BROWN et al., 2020).

Apesar de sua eficácia e simplicidade, a engenharia de *prompt* possui limitações, especialmente em tarefas que exigem conhecimento técnico aprofundado ou adaptação a contextos muito específicos. Nesses casos, uma alternativa mais robusta é o *fine-tuning*, que consiste no processo de adaptar um modelo de linguagem já treinado para uma tarefa específica, utilizando um conjunto adicional de dados rotulados. Antes da popularização dos grandes modelos generativos atuais, essa técnica era amplamente utilizada para especializar os modelos em domínios específicos, como jurídico, médico ou científico (HOWARD; RUDER, 2018).

Enquanto as técnicas utilizadas na engenharia de *prompt* (*zero-shot, few-shot, chain-of-thought*) aproveitam modelos pré-treinados sem alterar seus parâmetros, o *fine-tuning* ajusta internamente o modelo com dados específicos.

3.3 Trabalhos Relacionados

Nesta seção, discutimos alguns trabalhos relevantes que abordam diferentes aplicações e propostas envolvendo técnicas de Engenharia de *Prompt*.

Nascimento (2024) explora o uso da Engenharia de *Prompt* como estratégia para otimizar o uso de modelos de linguagem, especificamente o GPT, no Tribunal de Contas do Município do Rio de Janeiro, com o objetivo de aprimorar a análise documental.

O estudo de Kepel e Valogianni (2024) apresenta a ferramenta APET (do inglês, *Autonomous Prompt Engineering Toolkit*), que permite ao modelo GPT-4 otimizar seus próprios *prompts* de forma autônoma. Utilizando técnicas como *Expert Prompting* e *Chain-of-Thought*, a abordagem foi aplicada em tarefas como ordenação de palavras, geração de formas geométricas e raciocínio lógico. Os resultados indicaram um ganho de até 6,8% na acurácia, sem necessidade de *fine-tuning*, destacando o potencial da engenharia de *prompt* automatizada no aprimoramento de LLMs.

utilizar a aspas corretas

Já no trabalho de Silva (2024), foram analisadas as famosas "alucinações" em modelos de linguagem como ChatGPT 4, Gemini 1.5, Copilot e Perplexity, utilizando *prompts* que desencadeiam respostas irrelevantes, inventadas ou inconsistentes. Essas alucinações representam um desafio no avanço das LLMs.

No trabalho de Gouveia (2024), é investigado como a engenharia de *prompt* pode ser aplicada para extrair conhecimento de LLMs de forma estruturada e eficiente. O autor propõe uma metodologia experimental com variações de *prompts* para responder perguntas sobre um conjunto de documentos.

realizaram

Santos, Martins e Evangelista (2024) ~~realizou~~ uma revisão sistemática da literatura sobre o uso do ChatGPT em contextos acadêmicos de Interação Humano-Computador (IHC). O objetivo é identificar métodos eficazes de engenharia de *prompt* que possam aprimorar a precisão e a eficácia das respostas geradas pelo modelo, contribuindo para práticas mais confiáveis e consistentes em ambientes acadêmicos.

O estudo de Andrade (2024) investiga a aplicação de técnicas de engenharia de *prompt* em LLMs com o objetivo de resolver automaticamente correferências na língua portuguesa, ou seja, identificar quando diferentes expressões em um texto se referem à mesma entidade.

Um exemplo recente do uso combinado de engenharia de *prompt* e *fine-tuning* em diferentes LLMs é apresentado por Bitelli (2024), que trata da extração de entidades numéricas em textos jurídicos com modelos *open-source* (como LLaMA 2) e fechados (como ChatGPT e Gemini). O autor compara abordagens baseadas em *prompts* e ajuste fino, incluindo a técnica LoRA, e discute diferenças entre arquiteturas e os desafios de adaptação à língua portuguesa.

No trabalho de Nunes (2025), é explorado o uso da engenharia de *prompt* para automatizar a geração de código Python a partir de requisitos textuais. Utilizando LLMs com a biblioteca LangChain, o autor aplica técnicas que refinam os *prompts* e estruturam as instruções para gerar pipelines ETL de forma mais precisa.

Viana (2025) desenvolveu um estudo comparativo entre métodos tradicionais de previsão de séries temporais como *Random Forest* e redes neurais LSTM e uma abordagem inovadora baseada no uso de LLMs. Foram usadas técnicas de *prompt* para instruir o modelo Gemini 1.5 Pro a prever com base nos dados, sem precisar de treinamento adicional. O objetivo foi avaliar o desempenho preditivo dessas abordagens em diferentes cenários reais, como vendas no varejo e embarques no transporte público.

No trabalho de Silveira e Pereira (2025), é investigado o uso da engenharia de *prompt* como ferramenta discursiva na geração de respostas pelo ChatGPT. Por meio de *prompts* estruturados com personas ideológicas, as autoras analisam como o modelo responde a discursos associados à misoginia em comunidades *incel*, revelando implicações enunciativas e ideológicas nas interações com LLMs.

Tabela 3 – Comparação entre trabalhos relacionados quanto ao modelo LLM utilizado e características do *prompt*

Referência	**Modelo LLM**	**Tipo de Prompt**	**Objetivo do Prompt**	**Fine-tuning**	**Métricas**
Andrade (2024)	GPT-4	Few-shot + Zero-shot	Resolução de correferência	Não	CoNLL Score
Gouveia (2024)	LLaMA 3 + BERT + RoBERTa	Few-shot + One-shot + Zero-shot	Extração de informações em documentos	Não	Qualitativa-descritiva
Silva (2024)	GPT-4o + Gemini 1.5 + Copilot + Perplexity	Adversarial	Análise de alucinações em LLMs	Não	Qualitativa-descritiva
Nascimento (2024)	GPT-4	Role-Playing + Contextualization + Neg./Positive	Otimização de análise documental no TCMRio	Não	Qualitativa
Bitelli (2024)	GPT-3.5 + Gemini + LLaMA 2	Few-shot + Zero-shot	Extração de informações em textos jurídicos	Sim	F1-score, Precisão
Nunes (2025)	LLaMA 3.1 405B	Chain-of-Thought	Geração de código a partir de texto	Não	Qualitativa-descritiva
Kepel e Valogianni (2024)	GPT-4	Expert + Chain-of-Thought	Resolução de tarefas lógicas e visuais	Não	Acurácia
Viana (2025)	Gemini 1.5 Pro	Zero-shot	Comparar previsões de séries temporais	Não	SMAPE, SEM
Silveira e Pereira (2025)	GPT-4	Zero-shot	Analisar implicações ideológicas e discursivas em respostas do ChatGPT.	Não	Qualitativa-discursiva
Santos, Martins e Evangelista (2024)	GPT-4	Iterative	Engenharia de *prompt* para melhorar respostas do ChatGPT em IHC.	Não	Qualitativa-descritiva
Trabalho Proposto	GPT-4 + DeepSeek + LLaMA	Few-shot + Chain-of-Thought + Role-Playing + Negative + Contextualization **Não foi falado no texto abaixo**	Extração, interpretação e simplificação de informações técnicas em documentos PDFs	Não	Qualitativa

Diferentemente dos trabalhos anteriores, que focam em tarefas específicas dentro de áreas determinadas como o jurídico, educacional ou técnico, este trabalho propõe uma abordagem ampla, utilizando diversos LLMs como o GPT-4, LLaMA 3 e outros modelos disponíveis via API para a extração e interpretação de conteúdo especializado em documentos PDF enviados pelos usuários. Para isso, serão usadas técnicas de *prompt* como a *Chain-of-Thought*, *Few-shot*, *Negative* e *Role-Playing*, escolhidas por serem amplamente utilizadas, conforme apresentado

nos trabalhos apresentados nesta seção, além de favorecerem respostas claras e alinhadas ao conteúdo original. Por fim, pretendemos realizar uma avaliação qualitativa dos resultados, com base na clareza, coerência com o texto original passado como entrada, efetividade interpretativa e utilidade prática, por meio da comparação entre respostas geradas por diferentes LLMs e organizadas em quadros síntese.

4 Metodologia

Esta seção apresenta os procedimentos metodológicos adotados para a condução da pesquisa sobre a prática da engenharia de *prompt* aplicada a modelos de linguagem natural baseados em IA generativa. A metodologia foi delineada com o intuito de garantir o rigor científico e permitir a reprodutibilidade dos experimentos realizados, bem como a validação dos resultados obtidos.

4.1 Classificação da Pesquisa

A pesquisa proposta pode ser classificada de acordo com as seguintes dimensões:

- **Quanto à natureza:** trata-se de uma pesquisa *aplicada*, pois busca resolver um problema prático relacionado à construção eficiente de *prompts* para modelos generativos.
- **Quanto à abordagem:** a abordagem é *qualitativa*, com elementos *experimentais*, uma vez que envolve a análise do comportamento dos modelos frente a diferentes estruturas de *prompts*.
- **Quanto aos objetivos:** trata-se de uma pesquisa *exploratória e descritiva*, pois visa mapear práticas existentes e descrever os efeitos causados por diferentes tipos de instruções fornecidas aos modelos.
- **Quanto aos procedimentos técnicos:** a pesquisa utiliza como base uma *revisão bibliográfica* e a realização de *experimentação prática*.

4.2 Etapas da Pesquisa

A pesquisa será conduzida em duas etapas principais:

a) **Etapa 1 – Fundamentação Teórica e Definição de Métricas**

- Levantamento bibliográfico sobre o conceito de engenharia de *prompt* e suas variações (*zero-shot*, *few-shot*, *chain-of-thought*).
- Estudo da evolução dos modelos generativos de linguagem, como GPT, Gemini e LLaMA.

- Definição de métricas qualitativas para avaliação dos textos gerados, tais como coerência, relevância e adequação. **Como serão medidas?**

b) **Etapa 2 – Experimentação Prática**

- Seleção de modelos generativos disponíveis (OpenAI, Hugging Face, Google AI).
- Construção de diferentes conjuntos de *prompts* com estratégias variadas.
- Aplicação dos *prompts* aos modelos e coleta das respostas geradas.
- Análise dos resultados com base nas métricas definidas.
- Formulação de diretrizes para construção eficaz de *prompts*.

4.3 Coleta e Análise dos Dados

Os dados da pesquisa consistirão nas respostas textuais geradas pelos modelos de linguagem. A análise será de natureza qualitativa, incluindo:

- Avaliação comparativa entre as respostas dos modelos frente a diferentes *prompts*.
- Verificação da influência das variações nos *prompts* sobre a qualidade dos textos gerados.
- Registro e categorização das observações em quadros síntese.

4.4 Limitações da Pesquisa

As principais limitações do estudo incluem:

- Acesso parcial ou restrito a funcionalidades avançadas de alguns modelos.
- Subjetividade na análise qualitativa, mesmo com critérios pré-estabelecidos.
- Rápida evolução tecnológica, que pode impactar a validade de parte dos resultados ao longo do tempo.

4.5 Cronograma de Atividades

Atividade/Mês	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set
Definição do tema									
Levantamento bibliográfico									
Escrita Proposta TCC 1									
Apresentação TCC I									
Desenvolvimento do Framework									
Análise dos Resultados									
Escrita TCC II									
Apresentação TCC II									

4.6 Recursos

Para a execução deste trabalho, serão utilizados recursos que permitam tanto a experimentação prática quanto a análise qualitativa dos resultados. A implementação será realizada utilizando a linguagem de programação Python, escolhida por sua grande compatibilidade com bibliotecas de manipulação de documentos, integração com APIs e suporte a *frameworks* de IA. O desenvolvimento será realizado no ambiente Visual Studio Code (VSCode), devido à sua versatilidade, facilidade de organização de projetos e suporte a extensões úteis para testes e depuração.

O projeto utilizará dois tipos de abordagem para o uso de modelos de linguagem: execução local, por meio do ambiente Ollama, que permite rodar modelos como LLaMA e uma execução em nuvem, por meio de chamadas às APIs de modelos de linguagem, como o GPT e o DeepSeek Chat. Essa combinação visa comparar o desempenho de diferentes arquiteturas em tarefas de extração, interpretação e simplificação de conteúdos normativos em arquivos PDF.

Também serão empregadas bibliotecas específicas para leitura e extração de texto de PDFs, além de módulos para análise e pós-processamento das respostas geradas pelos modelos. Todo o ambiente será organizado em ambientes virtuais para garantir a reprodutibilidade dos experimentos.

5 Fontes de Pesquisa

ABDULLAHI, A.; TIMONERA, K. Large language model: A guide to the question ‘what is an llm”. *eWeek: Technology News for IT Professionals Tech Buyers*, 2024. Disponível em: ⟨https://www.eweek.com/artificial-intelligence/large-language-model/⟩.

ANDRADE, M. P. Uso da engenharia de prompt em llms para a resolução de correferência na língua portuguesa. *Universidade Federal de Santa Catarina, Centro de Comunicação e Expressão, Língua Portuguesa e Literaturas em Língua Portuguesa.*, 11 2024. Disponível em: ⟨https://repositorio.ufsc.br/handle/123456789/261149⟩.

ANTHROPIC. *Constitutional AI: Harmlessness from AI Feedback*. 2022. Disponível em: ⟨https://www.anthropic.com/index/constitutional-ai⟩.

ANTHROPIC. *Introducing Claude 2*. 2023. Disponível em: ⟨https://www.anthropic.com/index/claude-2⟩.

ANTHROPIC. *Introducing the Claude 3 model family*. 2024. Disponível em: ⟨https://www.anthropic.com/news/claude-3⟩.

ANTHROPIC. *Introducing Claude 3.5 Sonnet: our most advanced model yet*. 2025. Disponível em: ⟨https://www.anthropic.com/news/claude-3-5-sonnet⟩.

BAKTASH, J. A.; DAWODI, M. *Gpt-4: A Review on Advancements and Opportunities in Natural Language Processing*. 2023. Disponível em: ⟨https://arxiv.org/abs/2305.03195⟩.

BARRETO, R. G. et al. Utilizando redes neurais artificiais para o diagnóstico de câncer cervical. *REVISTA SAÚDE CIÊNCIA ONLINE*, v. 7, n. 2, 2018.

BITELLI, B. V. *Extração de informações numéricas em textos jurídicos usando Grandes Modelos de Língua (LLMs)*. Dissertação (Dissertação de Mestrado) — Universidade de São Paulo, Instituto de Matemática e Estatística, 2024. Disponível em: ⟨https://www.teses.usp.br/teses/disponiveis/45/45134/tde-05022025-190506/publico/Bruno_Bitelli_dissertacao_corrigida.pdf⟩.

BOMMASANI, R.; AL. et. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2022.

BROWN, T. B. et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, v. 33, 2020.

CASELI, H. M.; NUNES, M. G. V. (Ed.). *Processamento de Linguagem Natural: Conceitos, Técnicas e Aplicações em Português*. 2. ed. BPLN, 2024. ISBN 978-65-00-95750-1. Disponível em: ⟨https://brasileiraspln.com/livro-pln/2a-edicao/⟩.

DEEPSEEK-AI. *DeepSeek-V2: Unlocking the Scaling Limits of Language Models*. 2024. ArXiv preprint arXiv:2401.04778. Disponível em: ⟨https://arxiv.org/abs/2401.04778⟩.

DeepSeek-AI et al. *DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning*. 2025. Disponível em: ⟨https://arxiv.org/abs/2501.12948⟩.

DEVLIN, J. et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. Disponível em: ⟨https://arxiv.org/abs/1810.04805⟩.

DOSOVITSKIY, A. et al. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. 2021. Disponível em: ⟨https://arxiv.org/abs/2010.11929⟩.

DUOLINGO. Introducing duolingo max, a learning experience powered by gpt-4. 2023. Disponível em: ⟨https://blog.duolingo.com/duolingo-max/⟩.

FEDUS, W.; ZOPH, B.; SHAZEER, N. *Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity*. 2022. Disponível em: ⟨https://arxiv.org/abs/2101.03961⟩.

FEUERRIEGEL, S. et al. Generative ai. *Business amp; Information Systems Engineering*, Springer Science and Business Media LLC, v. 66, n. 1, p. 111–126, set. 2023. ISSN 1867-0202. Disponível em: ⟨http://dx.doi.org/10.1007/s12599-023-00834-7⟩.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. MIT Press, 2016. Disponível em: ⟨http://www.deeplearningbook.org⟩.

Google DeepMind. *Gemini 1 Technical Report*. 2023. Disponível em: ⟨https://storage.googleapis.com/deepmind-media/gemini/gemini_1_report.pdf⟩.

Google DeepMind. *Introducing Gemini: our most capable and general AI model yet*. 2023. Disponível em: ⟨https://www.deepmind.com/blog/introducing-gemini-our-most-capable-and-general-ai-model-yet⟩.

Google DeepMind. *Gemini 1.5 Technical Report*. 2024. Disponível em: ⟨https://storage.googleapis.com/deepmind-media/gemini/gemini_1.5_report.pdf⟩.

GOUVEIA, R. S. Prompt engineering for knowledge extraction from large language models. *Faculdade de Ciências e Tecnologia da Universidade de Coimbra*, 9 2024. Disponível em: ⟨https://estudogeral.uc.pt/handle/10316/118113⟩.

HARIRI, W. *Analyzing the Performance of ChatGPT in Cardiology and Vascular Pathologies*. 2023. Disponível em: ⟨https://arxiv.org/abs/2307.02518⟩.

HAYKIN, S. *Redes Neurais: Princípios e Prática*. Bookman Editora, 2001. ISBN 9788577800865. Disponível em: ⟨https://books.google.com.br/books?id=bhMwDwAAQBAJ⟩.

HOWARD, J.; RUDER, S. Universal language model fine-tuning for text classification. In: GUREVYCH, I.; MIYAO, Y. (Ed.). *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. [s.n.], 2018. p. 328–339. Disponível em: ⟨https://aclanthology.org/P18-1031/⟩.

JURAFSKY, D.; MARTIN, J. H. *Speech and Language Processing*. 2nd. ed. Upper Saddle River, NJ: Pearson Education, 2009.

JURAFSKY, D.; MARTIN, J. H. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*. 3rd. ed. [s.n.], 2025. Online manuscript released January 12, 2025. Disponível em: ⟨https://web.stanford.edu/~jurafsky/slp3/⟩.

KEPEL, D.; VALOGIANNI, K. Autonomous prompt engineering in large language models. *arXiv preprint arXiv:2407.11000*, 2024. Disponível em: ⟨https://arxiv.org/abs/2407.11000⟩.

KNOTH, N. et al. Ai literacy and its implications for prompt engineering strategies. *Computers and Education: Artificial Intelligence*, v. 6, p. 100225, 2024. ISSN 2666-920X. Disponível em: ⟨https://www.sciencedirect.com/science/article/pii/S2666920X24000262⟩.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *Nature*, Nature Publishing Group, v. 521, n. 7553, p. 436–444, 2015. Disponível em: ⟨https://www.nature.com/articles/nature14539⟩.

LEãO, J. C. et al. InteligÊncia artificial na educaÇÃo: aplicações do aprendizado de máquina para apoiar a aprendizagem adaptativa. *Revista Revivale: Dialogos Interdisciplinares*, v. 1, n. 1, 2021. Disponível em: ⟨https://revivale.ifnmg.edu.br/index.php/revivale/article/view/13⟩.

LI, B. et al. *MIMIC-IT: Multi-Modal In-Context Instruction Tuning*. 2023. Disponível em: ⟨https://arxiv.org/abs/2306.05425⟩.

LIU, J. et al. Is your code generated by chatGPT really correct? rigorous evaluation of large language models for code generation. In: *Thirty-seventh Conference on Neural Information Processing Systems*. [s.n.], 2023. Disponível em: ⟨https://openreview.net/forum?id=1qvx610Cu7⟩.

LIU, P. et al. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*, 2021. Disponível em: ⟨https://arxiv.org/abs/2107.13586⟩.

MADADI, Y. et al. *ChatGPT Assisting Diagnosis of Neuro-ophthalmology Diseases Based on Case Reports*. 2023. Disponível em: ⟨https://arxiv.org/abs/2309.12361⟩.

META. april 2025. Disponível em: ⟨https://about.fb.com/news/2025/04/introducing-meta-ai-app-new-way-access-ai-assistant/⟩.

MITCHELL, T. *Machine Learning*. McGraw-Hill, 1997. (McGraw-Hill International Editions). ISBN 9780071154673. Disponível em: ⟨https://books.google.com.br/books?id=EoYBngEACAAJ⟩.

NASCIMENTO, J. R. d. Exploração de técnicas de engenharia de prompt para aprimorar os resultados do uso de llm no tcmrio. 2024. Disponível em: ⟨https://repositorio.ufrn.br/handle/123456789/58251⟩.

NORVIG, P.; RUSSELL, S. *Inteligência Artificial: Tradução da 3a Edição*. Elsevier Brasil, 2013. ISBN 9788535251418. Disponível em: ⟨https://books.google.com.br/books?id=BsNeAwAAQBAJ⟩.

NUNES, R. L. P. Uma proposta de sistema para geraÇÃo de cÓdigo python a partir de uma descriÇÃo de requisitos de um sistema de informaÇÃo: Uma abordagem com large language model. *INSTITUTO FEDERAL DO ESPÍRITO SANTO CURSO BACHARELADO EM SISTEMAS DE INFORMAÇÃO*, 2025. Disponível em: ⟨https://repositorio.ifes.edu.br/handle/123456789/5927⟩.

OPENAI. Khan academy explores the potential for gpt-4 in a limited pilot program. 2023. Disponível em: ⟨https://openai.com/index/khan-academy/⟩.

OpenAI et al. *GPT-4 Technical Report*. 2024. Disponível em: ⟨https://arxiv.org/abs/2303.08774⟩.

PALIWAL, M.; KUMAR, U. A. Neural networks and statistical techniques: A review of applications. *Expert Systems with Applications*, v. 36, n. 1, p. 2–17, 2009. ISSN 0957-4174. Disponível em: ⟨https://www.sciencedirect.com/science/article/pii/S0957417407004952⟩.

PANDEY, M. et al. The transformational role of gpu computing and deep learning in drug discovery. *Nature Machine Intelligence*, v. 4, p. 211–221, 03 2022.

RADFORD, A. et al. Improving language understanding by generative pre-training. *Open AI Research*, 2018. Disponível em: ⟨https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf⟩.

REYNOLDS, L.; MCDONNELL, K. Prompt programming for large language models: Beyond the few-shot paradigm. *arXiv preprint arXiv:2102.07350*, 2021.

ROZIèRE, B. et al. *Code Llama: Open Foundation Models for Code*. 2024. Disponível em: ⟨https://arxiv.org/abs/2308.12950⟩.

SANTOS, G.; MARTINS, J.; EVANGELISTA, G. Prompt engineering com chatgpt no contexto acadêmico de ihc: uma revisão rápida da literatura. SBC, Porto Alegre, RS, Brasil, p. 144–148, 2024. ISSN 0000-0000. Disponível em: ⟨https://sol.sbc.org.br/index.php/ihc_estendido/article/view/30653⟩.

SARAVIA, E. Prompt Engineering Guide. *https://github.com/dair-ai/Prompt-Engineering-Guide*, 12 2022.

SILVA, W. J. L. Engenharia de prompt: Uma análise das ”alucinações”em inteligências artificiais generativas. *Fundação de Ensino e Pesquisa do Sul de Minas*, 6 2024. Disponível em: ⟨http://repositorio.unis.edu.br/handle/prefix/2727⟩.

SILVEIRA, K.; PEREIRA, M. H. d. M. ImplicaÇÕes enunciativas e ideolÓgicas da engenharia de prompt no chatgpt: um estudo sobre a construção discursiva da misoginia entre jovens incels. *Open Minds International Journal*, v. 6, n. 1, p. 191–208, jun. 2025. Disponível em: ⟨https://openmindsjournal.com/index.php/openminds/article/view/350⟩.

SPATARO, J. Introducing microsoft 365 copilot – your copilot for work. 2023. Disponível em: ⟨https://blogs.microsoft.com/blog/2023/03/16/introducing-microsoft-365-copilot-your-copilot-for-work/⟩.

SPöRL, C.; CASTRO, E.; LUCHIARI, A. AplicaÇÃo de redes neurais artificiais na construÇÃo de modelos de fragilidade ambiental. *Revista do Departamento de Geografia*, v. 21, p. 113–135, jul. 2011. Disponível em: ⟨https://www.revistas.usp.br/rdg/article/view/47233⟩.

TOUVRON, H. et al. *LLaMA: Open and Efficient Foundation Language Models*. 2023. Disponível em: ⟨https://arxiv.org/abs/2302.13971⟩.

VALSAI. *MATH 500 Benchmark*. 2025. Disponível em: ⟨https://www.vals.ai/benchmarks/math500-05-30-2025⟩.

VASWANI, A. et al. Attention is all you need. *Advances in Neural Information Processing Systems*, v. 30, 2017.

VASWANI, A. et al. Attention is all you need. 2017. Disponível em: ⟨https://arxiv.org/abs/1706.03762⟩.

VIANA, L. Z. B. Previsão de séries temporais orientada por prompt com modelos de linguagem de grande escalas. *Universidade Federal do Ceará, Campus Cratéus*, 2025. Disponível em: ⟨http://repositorio.ufc.br/handle/riufc/80020⟩.

WANG, Y. et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *arXiv preprint arXiv:2406.01574*, 2024.

WEI, J. et al. *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models*. 2023. Disponível em: ⟨https://arxiv.org/abs/2201.11903⟩.

WHITE, J. et al. *A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT*. 2023. Disponível em: ⟨https://arxiv.org/abs/2302.11382⟩.

WU, C. et al. *PMC-LLaMA: Towards Building Open-source Language Models for Medicine*. 2023. Disponível em: ⟨https://arxiv.org/abs/2304.14454⟩.

XIE, Q. et al. *Me LLaMA: Foundation Large Language Models for Medical Applications*. 2024. Disponível em: ⟨https://arxiv.org/abs/2402.12749⟩.

XU, B. et al. *ExpertPrompting: Instructing Large Language Models to be Distinguished Experts*. 2023. ArXiv preprint arXiv:2305.14688. Disponível em arXiv.

YANG, X. et al. *What GPT Knows About Who is Who*. 2022. Disponível em: ⟨https://arxiv.org/abs/2205.07407⟩.

Itabaiana/SE, _____ de _____ de 20 _____

_____

Orientador

_____

Orientado(a)