PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS
ESCOLA POLITÉCNICA E DE ARTES
GRADUAÇÃO EM ENGENHARIA DA COMPUTAÇÃO

# USO DA INTELIGÊNCIA ARTIFICIAL EM OTIMIZAÇÃO DE CONSULTA DE BANCO DE DADOS NÃO RELACIONAIS

LEONARDO CASTRO FERREIRA

GOIÂNIA
2024

LEONARDO CASTRO FERREIRA

# USO DA INTELIGÊNCIA ARTIFICIAL EM OTIMIZAÇÃO DE CONSULTA DE BANCO DE DADOS NÃO RELACIONAIS

Trabalho de Conclusão de Curso apresentado à Escola Politécnica e de Artes, da Pontifícia Universidade Católica de Goiás, como parte dos requisitos para a obtenção do título de Bacharel em Engenharia da Computação.

Orientador(a):

Prof. Me. Fernando Gonçalves Abadia

Banca examinadora:

Prof. Dr. Nilson Cardoso Amaral

Prof. Dr. Jose Olimpio Ferreira

GOIÂNIA
2024

LEONARDO CASTRO FERREIRA

**USO DA INTELIGÊNCIA ARTIFICIAL EM OTIMIZAÇÃO DE CONSULTA DE BANCO DE DADOS NÃO RELACIONAIS**

Trabalho de Conclusão de Curso aprovado em sua forma final pela Escola Politécnica e de Artes, da Pontifícia Universidade Católica de Goiás, para obtenção do título de Bacharel em Engenharia da Computação, em _____/_____/_____.

_____

Orientador: Prof. Me. Fernando Gonçalves Abadia

_____

Prof. Dr. Nilson Cardoso Amaral

_____

Prof. Dr. Jose Olimpio Ferreira

GOIÂNIA

2024

À Deus pela minha vida e oportunidades. Ao meu pai, minha mãe e aos meus avós, que já não estão mais aqui entre nós, pelo carinho, amor, afago, diversão e apoio.

## AGRADECIMENTOS

Ao Professor Fernando Gonçalves Abadia, orientador acadêmico, pelo apoio e confiança depositada.

A todos que direta ou indiretamente colaboraram para materialização deste trabalho.

"A teoria dos bancos de dados fornece as bases necessárias para entender como os sistemas funcionam e por que certos métodos são utilizados. No entanto, a prática da implementação e otimização é igualmente essencial para resolver problemas do mundo real."

Hector Garcia, Jeffrey D. Ullman, and Jennifer Widom

# RESUMO

Este estudo propõe a utilização de técnicas de inteligência artificial (IA) para otimizar o desempenho de sistemas de bancos de dados, explorando como algoritmos de aprendizado de máquina podem ser aplicados para aprimorar a eficiência tanto no gerenciamento quanto na consulta de dados. O objetivo principal é desenvolver e implementar modelos de IA, como redes neurais artificiais e algoritmos genéticos, visando automatizar processos essenciais, como a indexação de dados, a otimização das consultas SQL e a previsão da carga de trabalho dos sistemas de banco de dados. Com a aplicação dessas técnicas, espera-se obter uma melhoria significativa no tempo de resposta das consultas, além de aumentar consideravelmente a eficiência geral do sistema. Isso permite uma utilização mais eficaz dos recursos computacionais, possibilitando que sistemas que manipulam grandes volumes de dados operem de maneira mais rápida e eficiente. A automação desses processos pode proporcionar um desempenho mais ágil e confiável, resultando em um gerenciamento de banco de dados mais simplificado. Dessa forma, os administradores de banco de dados poderão focar em atividades mais estratégicas, deixando que tarefas repetitivas e complexas sejam gerenciadas automaticamente pela IA. Além disso, este estudo discute os desafios potenciais e as limitações, como a complexidade de implementação dessas técnicas e a compatibilidade com sistemas de banco de dados tradicionais. Ao final, o trabalho sugere direções futuras para pesquisas adicionais, enfatizando como o uso contínuo de IA tem o potencial de transformar e revolucionar o campo da gestão de bancos de dados, promovendo soluções cada vez mais inovadoras e eficientes.



**Palavras-Chave:** Inteligência artificial (IA). Otimização de Banco de Dados. Aprendizado de máquina. Eficiência de consulta. Desempenho do sistema.

## ABSTRACT

This study proposes the use of artificial intelligence (AI) techniques to optimize the performance of database systems, exploring how machine learning algorithms can be applied to enhance efficiency in both data management and querying. The main objective is to develop and implement AI models, such as artificial neural networks and genetic algorithms, aimed at automating essential processes like data indexing, SQL query optimization, and workload prediction for database systems. By applying these techniques, the study seeks to achieve significant improvements in query response times and a substantial increase in overall system efficiency. This enables more effective utilization of computational resources, allowing systems handling large volumes of data to operate faster and more efficiently. Automating these processes can lead to more agile and reliable performance, resulting in a simplified database management experience. As a result, database administrators can focus on more strategic activities, leaving repetitive and complex tasks to be automatically managed by AI. Furthermore, the study discusses potential challenges and limitations, such as the complexity of implementing these techniques and their compatibility with traditional database systems. Finally, the research suggests future directions for additional studies, emphasizing how the continued use of AI has the potential to transform and revolutionize the field of database management, fostering increasingly innovative and efficient solutions.



**Keywords:** Artificial Intelligence. Database Optimization. Machine Learning. Query Efficiency. System Performance.

## LISTA DE ILUSTRAÇÕES

Figura 1 - Lista de stopwords ..... 26
Figura 2 - Pré-Processamento do texto ..... 26
Figura 3 - Carregando o JSON ..... 27
Figura 4 - Criação da matriz de características ..... 27
Figura 5 - Configurando o pipeline ..... 28
Figura 6 - Consulta dos produtos ..... 28
Figura 7 - Solicitação das categorias ..... 29
Figura 8 - Exibição dos produtos relacionados ..... 29
Figura 9 - Escolha do produto ..... 29
Figura 10 - Valor máximo ..... 30
Figura 11 - Configuração dos filtros ..... 30
Figura 12 - Resultados ..... 30
Figura 13 - Escolha das categorias ..... 31
Figura 14 - Escolha do produto ..... 31
Figura 15 - Valor máximo ..... 32
Figura 16 - Resultado da busca ..... 32
Figura 17 - Segundo exemplo ..... 32

## LISTA DE ABREVIATURAS

| | |
|---|---|
| AGI | Inteligência Artificial Geral |
| ASI | Superinteligência Artificial |
| PLN | Processamento de Linguagem Natural |
| RL | Aprendizado por reforço |
| VAEs | Autoencoders variacionais |

## LISTA DE SIGLAS

| | |
|---|---|
| AI | Artificial Intelligence (Inteligência Artificial) |
| ANI | Artificial Narrow Intelligence (Inteligência Artificial Estreita) |
| IA | Inteligência Artificial |
| IDF | Inverse Document Frequency (Frequência Inversa de Documento) |
| JSON | JavaScript Object Notation (Notação de Objeto JavaScript) |
| ML | Machine Learning (Aprendizado de Máquina) |
| TF- | Term Frequency - (Frequência de Termo) |
| NoSQL | Not Only SQL" (Não Apenas SQL) |
| SQL | Structured Query Language, (Linguagem de Consulta Estruturada) |

**SUMÁRIO**

**1.INTRODUÇÃO ..... 13**

**2. REFERENCIAL TEORICO ..... 15**

**2.1 Banco de dados não relacional ..... 15**

**2.2 Otimização de consulta ..... 16**

**2.3 Inteligência Artificial ..... 17**

2.3.1 Aprendizado de Máquina ..... 18

2.3.2 Tipos de inteligência artificial: IA fraca vs. IA forte ..... 20

2.3.3 A ascensão dos modelos generativos ..... 21

2.3.4 História da inteligência artificial: datas e nomes marcantes. ..... 22

**3 METODOLOGIA ..... 25**

**4. DESENVOLVIMENTO ..... 26**

**5. RESULTADOS ..... 31**

**6. CONSIDERAÇÕES FINAIS ..... 33**

**REFERÊNCIAS ..... 36**

# 1.INTRODUÇÃO

Nos últimos anos, as organizações têm gerado uma quantidade cada vez maior de dados, o que fez com que fosse necessário melhorar o desempenho e a eficiência dos sistemas que gerenciam essas informações. Uma das soluções mais promissoras para esse problema é otimizar as consultas usando técnicas de Machine Learning (ML). Com o uso de algoritmos de aprendizado de máquina, é possível prever padrões de consulta, o que torna o acesso e o processamento dos dados muito mais rápidos. Além disso, a aplicação de modelos preditivos ajuda a antecipar o comportamento de usuários e sistemas, facilitando a tomada de decisões em tempo real, como destaca Marcus, M., & Scholl, M. H. (2019) em "Query Optimization Using Machine Learning Techniques: A Survey".

A automação de tarefas internas também é fundamental, pois torna as rotinas repetitivas mais eficientes e com menos intervenção humana. Quando combinada com o machine learning, a automação ajuda a otimizar os fluxos de trabalho e reduzir erros operacionais. Além disso, integrar dados de diferentes sistemas e plataformas se torna cada vez mais essencial para garantir consultas eficientes, já que essas fontes de dados são variadas. Para isso, é necessária uma infraestrutura robusta e flexível, como apontam Mohapatra, S., & Samal, B. (2020) em "Automation and Machine Learning Integration in Modern Enterprises: Enhancing Workflow Efficiency and Reducing Operational Errors".

Ao mesmo tempo, o uso de ferramentas de backup é fundamental para garantir a segurança e a disponibilidade dos dados. Quando essas ferramentas são combinadas com modelos de otimização e automação, os sistemas continuam funcionando mesmo quando ocorrem falhas, mantendo a integridade e a continuidade dos serviços, como ressaltado por W. Stallings (2018) em "Data and Computer Communications".

Este trabalho explora como a união dessas tecnologias — otimização de consultas, machine learning, modelos preditivos, automação de tarefas, integração de dados e ferramentas de backup — pode transformar a maneira como as organizações gerenciam e acessam seus dados, criando sistemas mais rápidos, seguros e escaláveis. Além disso, o estudo inclui uma abordagem experimental, com

o desenvolvimento de um sistema baseado em banco de dados não relacional para verificar, na prática, os benefícios apresentados pela teoria.

A metodologia adotada combina duas abordagens principais: a pesquisa bibliográfica, que envolveu a análise de artigos científicos e outras fontes para fundamentar teoricamente o trabalho, e a pesquisa experimental, que consistiu no desenvolvimento e teste de um código prático para validar os conceitos e hipóteses discutidos ao longo do estudo.

## 2. REFERENCIAL TEORICO

Neste capítulo, são apresentados os fundamentos teóricos que sustentam este trabalho, abordando conceitos essenciais e tecnologias relacionadas ao tema. Primeiramente, são explorados os bancos de dados não relacionais, suas características, vantagens e limitações, oferecendo uma visão detalhada sobre as diferentes categorias e suas aplicações. Em seguida, discute-se o processo de otimização de consultas, destacando estratégias e ferramentas que melhoram a eficiência no acesso e manipulação de dados. Por fim, é examinada a inteligência artificial (IA), com foco no aprendizado de máquina, tipos de IA e modelos generativos, além de um panorama histórico que contextualiza os avanços na área. Este capítulo fornece uma base sólida para compreender o contexto e as soluções propostas ao longo do estudo.

### 2.1 Banco de dados não relacional

Na era da informação, onde a diversidade e a complexidade dos dados crescem de forma exponencial, os sistemas de gerenciamento de banco de dados enfrentam o desafio de oferecer soluções flexíveis e eficientes para atender às demandas de diferentes contextos. Nesse cenário, surgem os bancos de dados não relacionais, conhecidos como NoSQL, que se apresentam como uma alternativa inovadora aos tradicionais bancos relacionais. Com estruturas menos rígidas e uma abordagem projetada para lidar com dados não estruturados ou semiestruturados, esses sistemas atendem a necessidades específicas de armazenamento e consulta, marcando uma mudança significativa na forma como os dados são organizados e acessados.

De acordo com a AWS (2023), "os bancos de dados NoSQL são criados especificamente para modelos de dados específicos e armazenam dados em esquemas flexíveis que se escalam facilmente para aplicações modernas. Os bancos de dados NoSQL são amplamente reconhecidos por sua facilidade de desenvolvimento, funcionalidade e performance em escala."

Um banco de dados não relacional funciona de forma diferente dos bancos tradicionais que utilizam tabelas organizadas em linhas e colunas. Ele organiza os dados de um jeito mais flexível, que se adapta melhor ao tipo de informação

armazenada. Por exemplo, é possível guardar dados como pares chave/valor, em documentos JSON, ou até como grafos que mostram conexões entre diferentes elementos.

O que todos esses métodos têm em comum é que não seguem o modelo relacional. Eles são voltados para lidar com tipos específicos de dados e para consultas que atendem a necessidades específicas. Por exemplo, bancos de séries temporais são ideais para dados organizados por períodos, enquanto bancos de grafos são perfeitos para mapear relações entre entidades. Apesar disso, nenhum deles é a melhor escolha para lidar com dados transacionais de forma mais ampla.

O termo "NoSQL" se refere a bancos de dados que não dependem do SQL para realizar consultas. Em vez disso, eles utilizam outras linguagens ou métodos para acessar as informações. Na prática, "NoSQL" é sinônimo de "banco de dados não relacional", embora muitos deles suportem consultas parecidas com SQL. A diferença está na forma como processam essas consultas, que é bem distinta de um sistema relacional tradicional.

Assim como nos bancos relacionais, os bancos NoSQL também têm diferentes tipos e especializações. Cada um traz vantagens específicas, mas também exigem aprendizado e estratégias próprias de uso. A seguir, vamos detalhar as principais categorias de bancos de dados não relacionais ou NoSQL.

## 2.2 Otimização de consulta

Conforme descrito no LinkedIn (2023), "o aprendizado de máquina é uma ferramenta poderosa para otimização de consultas, pois pode ajudar a automatizar, aprimorar e adaptar o processo de recuperação de informações, melhorando a eficiência, a precisão e a relevância das consultas."

A otimização de consulta é o processo de encontrar a maneira mais eficiente de executar uma consulta de banco de dados. Imagine que você tem uma grande biblioteca de livros, cada um representando uma entrada no banco de dados. Agora, quando você faz uma pergunta, como "Quais são todos os livros escritos por este autor específico?", o banco de dados precisa procurar entre todos os livros para encontrar a resposta.

Mas, em uma biblioteca grande, isso pode ser demorado. Aqui é onde a otimização de consulta entra em jogo. É como ter um bibliotecário inteligente que sabe exatamente onde cada livro está e como encontrá-lo rapidamente. Em termos de banco de dados, isso significa que o sistema tenta encontrar a maneira mais eficiente de buscar os dados que você precisa, levando em consideração coisas como as restrições que você colocou na sua consulta (como encontrar livros de um autor específico) e os recursos disponíveis (como o poder de processamento do computador).

A otimização de consulta envolve estratégias inteligentes, como usar índices (que são como índices na parte de trás de um livro que ajudam a encontrar informações mais rapidamente), escolher a ordem correta de operações e minimizar a quantidade de dados que precisam ser examinados. Quando tudo é feito corretamente, sua consulta é executada rapidamente e você obtém a informação que precisa sem esperar muito tempo.

Um dos primeiros passos na otimização de consultas é analisar a consulta e entender sua intenção e contexto. Essa abordagem se baseia em técnicas avançadas de processamento de linguagem natural (PLN) para entender melhor a intenção e o contexto por trás de cada consulta.

## 2.3 Inteligência Artificial

A Inteligência Artificial (IA) é uma área da tecnologia focada em criar sistemas que imitam ou reproduzem habilidades humanas, como pensar, aprender, tomar decisões e resolver problemas. Em outras palavras, a ideia é desenvolver máquinas e algoritmos capazes de realizar tarefas sozinhos, adaptando-se a diferentes situações e lidando com informações complexas. Essas tarefas podem ser simples, como identificar padrões, ou mais complicadas, como tomar decisões rápidas em cenários imprevisíveis. "A Inteligência Artificial (IA) é uma tecnologia que permite que computadores e máquinas simulem a capacidade de resolução de problemas e a inteligência humana" (IBM, 2024).

A IA já está presente no nosso dia a dia, muitas vezes de forma discreta. Assistentes virtuais como Siri e Alexa, sistemas de GPS e até carros que dirigem sozinhos são exemplos disso. O que torna esses sistemas tão bons é a capacidade de aprenderem e melhorarem ao longo do tempo, com base nos dados que

recebem. Isso acontece graças a técnicas como aprendizado de máquina (machine learning) e deep learning, que permitem que os sistemas se adaptem e fiquem cada vez mais eficientes à medida que acumulam experiência.

O aprendizado de máquina cria algoritmos que analisam dados, identificam padrões e fazem previsões, sem precisar ser programado para cada tarefa específica. Já o deep learning, uma evolução do aprendizado de máquina, usa redes neurais avançadas para trabalhar com grandes volumes de dados, possibilitando tarefas como reconhecimento de imagens ou tradução de textos com alta eficiência. Sistemas como o ChatGPT são um exemplo claro disso, melhorando continuamente sua capacidade de interpretar contextos e oferecer respostas cada vez mais precisas.

A IA também avançou muito com as tecnologias gerativas, que criam coisas novas como textos, imagens, vídeos e até códigos ou moléculas. Essa habilidade de criar, como no caso do ChatGPT, é um marco importante, pois mostra que as máquinas não só reagem a comandos, mas também podem inovar a partir das instruções dos usuários. Isso amplia o alcance da IA, trazendo possibilidades criativas e práticas em diversas áreas.

Por outro lado, esses avanços trazem desafios importantes. O uso responsável da IA está no centro de debates que envolvem questões como privacidade, transparência, impacto no mercado de trabalho e possíveis discriminações geradas por decisões automatizadas. Com a IA se tornando cada vez mais integrada aos diferentes setores, é fundamental criar regras claras e práticas éticas para que seus benefícios sejam amplos e seus riscos, bem controlados.

### 2.3.1 Aprendizado de Máquina

O aprendizado de máquina é uma ferramenta incrível para melhorar as buscas. Ele utiliza técnicas como análise, tokenização, derivação, lematização e reconhecimento de entidades para identificar as palavras-chave, frases e ideias mais importantes de uma consulta. Isso facilita conectar a busca aos dados e campos corretos, garantindo resultados mais certeiros.

Além disso, ele resolve problemas de consultas confusas, incompletas ou complicadas. Com estratégias como expansão, reformulação, sugestão e

classificação de consultas, o processo fica mais claro e eficiente, enriquecendo ou simplificando o que foi perguntado. Isso dá ao usuário mais opções e deixa a interação com o sistema muito mais prática.

No geral, o aprendizado de máquina é indispensável para entregar respostas mais relevantes e tornar as buscas mais rápidas e assertivas. "O aprendizado de máquina 'profundo' pode usar conjuntos de dados rotulados, também conhecidos como aprendizado supervisionado, para informar seu algoritmo, mas não exige necessariamente um conjunto de dados rotulado. O processo de deep learning pode ingerir dados não estruturados em sua forma bruta (por exemplo, texto ou imagens) e pode determinar automaticamente o conjunto de recursos que distinguem diferentes categorias de dados umas das outras" (IBM, 2024).

Outra parte essencial da otimização de buscas é planejar e executar bem as consultas. Nesse ponto, o aprendizado de máquina se destaca com técnicas de aprendizado por reforço, como Q-learning, redes Q profundas e métodos de gradiente de políticas. Essas ferramentas permitem que o sistema aprenda com experiências anteriores, ajustando parâmetros e decisões para melhorar sua eficiência.

Ele também é muito bom em lidar com mudanças constantes nos dados e no ambiente. Usando aprendizado online, ativo ou por transferência, o sistema consegue se atualizar com novos dados e aproveitar conhecimentos já adquiridos em outros contextos.

Com isso, o uso inteligente de aprendizado de máquina nessa etapa melhora bastante a eficiência das buscas e garante resultados mais relevantes, deixando os usuários mais satisfeitos.

Por fim, avaliar os resultados da busca é uma fase crucial. O aprendizado de máquina entra em cena com técnicas como classificação, ordenação e agrupamento para medir a qualidade e relevância das respostas. Ele também personaliza os resultados, levando em conta as preferências e comportamentos de cada usuário, e utiliza recomendações e feedback para melhorar ainda mais a experiência.

Embora seja uma ferramenta poderosa, o aprendizado de máquina também enfrenta desafios, como qualidade dos dados, escalabilidade, privacidade e transparência. Por isso, é essencial equilibrar seus benefícios com os possíveis riscos, garantindo que essas tecnologias sejam usadas de forma ética e responsável para oferecer buscas mais eficientes, precisas e relevantes.

### 2.3.2 Tipos de inteligência artificial: IA fraca vs. IA forte

"A IA fraca – também conhecida como IA estreita ou inteligência estreita artificial (ANI) – é a IA treinada e focada para executar tarefas específicas. A IA fraca impulsiona a maior parte da IA que nos cerca hoje. 'Narrow' pode ser um descritor mais adequado para esse tipo de IA, pois é tudo menos fraco: permite alguns aplicativos muito robustos, como a Siri da Apple, a Alexa da Amazon, o IBM watsonx e veículos autônomos. A IA forte é composta por inteligência artificial geral (AGI) e superinteligência artificial (ASI)" (IBM, 2024).

A Inteligência Artificial (IA) pode ser dividida em dois tipos principais: IA fraca e IA forte, que têm diferenças bem marcantes em termos de objetivos e capacidades.

A IA fraca, também chamada de IA estreita (ANI - Artificial Narrow Intelligence), é feita para resolver tarefas bem específicas. Exemplos disso são assistentes virtuais como Siri e Alexa ou os sistemas que operam carros autônomos. Apesar de ser chamada de "fraca", ela é extremamente eficiente no que se propõe a fazer, mas não consegue sair do escopo para o qual foi programada. Esses sistemas não têm consciência nem são capazes de raciocinar de forma geral. Exemplos comuns incluem recomendações de filmes, diagnósticos médicos baseados em dados e automação de tarefas simples, sempre dentro de um contexto bem definido.

Já a IA forte é algo muito mais avançado. Dentro dela, temos a chamada Inteligência Artificial Geral (AGI - Artificial General Intelligence), que seria uma IA capaz de fazer praticamente qualquer coisa que um ser humano faz: aprender, se adaptar e atuar em diversas áreas sem precisar de reprogramação. Além disso, há a ideia da Superinteligência Artificial (ASI - Artificial Superintelligence), que seria uma IA ainda mais poderosa, superando as habilidades humanas em todos os sentidos. No entanto, esses conceitos ainda são teóricos ou aparecem em histórias de ficção científica, como o HAL 9000 do filme *2001: Uma Odisseia no Espaço*. Nenhuma delas foi desenvolvida até hoje.

A principal diferença entre esses dois tipos de IA estar no que elas conseguem fazer e em sua complexidade. Enquanto a IA fraca se limita a tarefas

específicas, a IA forte busca criar sistemas que pensem e ajam de maneira mais parecida com o raciocínio humano, sendo mais amplas e adaptáveis.

Quando falamos de aprendizado de máquina e *deep learning*, as abordagens também se diferenciam. O aprendizado de máquina tradicional usa redes neurais mais simples e depende de dados bem-organizados e pré-processados. Por outro lado, o *deep learning* vai além, utilizando redes neurais profundas que conseguem aprender sozinhas com grandes volumes de dados, sem precisar de tanta intervenção humana. Isso faz com que o *deep learning* seja perfeito para problemas mais complexos, como reconhecimento de imagens e tradução automática, enquanto o aprendizado de máquina tradicional é mais usado para tarefas que lidam com dados mais estruturados e com menos variáveis.

### 2.3.3 A ascensão dos modelos generativos

"Os VAEs abriram as comportas para a modelagem generativa profunda, tornando os modelos mais fáceis de escalar", afirmou Akash Srivastava, especialista em IA generativa do MIT-IBM Watson AI Lab. "Muito do que pensamos hoje como IA generativa começou aqui" (IBM, 2024).

A IA generativa usa modelos de deep learning para criar conteúdo a partir de dados brutos, como textos da Wikipedia ou até obras de arte. Em resumo, ela gera algo novo que é parecido com os dados originais, mas sem fazer uma cópia exata.

Esses modelos começaram sendo usados para fazer cálculos e análises, mas com o avanço do deep learning, eles passaram a criar imagens, sons e outros tipos de conteúdo mais complexos. Os primeiros que conseguiram fazer isso foram os autoencoders variacionais (VAEs), que apareceram em 2013 e já conseguiam gerar imagens e sons super-realistas.

Modelos como o GPT-3, BERT e DALL-E 2 mostraram todo o potencial dessa tecnologia. No futuro, esses modelos serão treinados com uma quantidade gigante de dados e poderão ser usados em diversas tarefas, sem ter que precisar de muito ajuste. Isso nos leva a um futuro em que sistemas de IA serão mais flexíveis e capazes de aprenderem e atuarem em várias áreas.

A ideia é que, com esses “modelos básicos”, a IA se torne mais acessível para empresas, já que não exigirá tantos dados organizados e poderá ser aplicada em diferentes situações. A IBM, por exemplo, espera que esses modelos sejam

usados por empresas de todos os tamanhos em plataformas de nuvem híbrida, facilitando o uso e a implementação da IA no mundo dos negócios.

Hoje, a IA já está presente em várias áreas, e um exemplo disso é o reconhecimento de fala, onde ela converte a voz em texto usando processamento de linguagem natural (PLN). Isso facilita a busca por voz em smartphones, como na Siri, e também ajuda a converter mensagens de voz para texto em vários idiomas. Um exemplo de como a IA tem sido útil é o IBM Watson Text to Speech, que tem ajudado a melhorar a acessibilidade nas salas de aula.

### 2.3.4 História da inteligência artificial: datas e nomes marcantes.

"A ideia de 'uma máquina que pensa' remonta à Grécia antiga. Mas desde o advento da computação eletrônica, eventos e marcos importantes na evolução da inteligência artificial incluem o trabalho de Alan Turing, que em 1950 publicou o artigo 'Computing Machinery and Intelligence', onde propôs a famosa questão 'As máquinas pensam?'" (IBM, 2024).

A ideia de “máquinas que pensam” remonta à Antiguidade, com suas primeiras menções na Grécia antiga. No entanto, foi com o advento dos computadores que marcos históricos e eventos importantes começaram a moldar o desenvolvimento da Inteligência Artificial (IA), transformando-a de uma ideia filosófica para uma realidade científica e tecnológica. A seguir, alguns dos momentos mais significativos dessa trajetória:

1950: Alan Turing, considerado um dos pais da computação moderna, publica o artigo Computing Machinery and Intelligence, onde propõe a famosa questão "As máquinas podem pensar?" e introduz o "Teste de Turing". Este teste visa determinar se uma máquina é capaz de imitar o comportamento humano de forma convincente, sendo considerado um dos primeiros marcos na pesquisa de IA.

1956: Na conferência de IA realizada no Dartmouth College, John McCarthy, um dos fundadores da área, cunha o termo "Inteligência Artificial". Esse evento é frequentemente apontado como o nascimento formal da IA como um campo de estudo. No mesmo ano, Allen Newell e Herbert Simon lançam o Logic Theorist, o primeiro programa de IA a ser desenvolvido, marcando o início de uma nova era no uso de algoritmos para resolver problemas complexos.

1967: Frank Rosenblatt cria o Mark 1 Perceptron, o primeiro computador a utilizar uma rede neural capaz de aprender por tentativa e erro, uma técnica fundamental para o desenvolvimento futuro da IA. Contudo, em 1968, Marvin Minsky e Seymour Papert publicam o livro Perceptrons, onde criticam as limitações das redes neurais da época, o que resultou em um retrocesso no interesse por esse tipo de abordagem durante a década de 1970.

Década de 1980: As redes neurais voltam a ganhar atenção com o avanço de algoritmos de retropropagação, que permitem a aprendizagem mais eficiente e a aplicação de redes neurais em problemas mais complexos, como reconhecimento de padrões e processamento de linguagem natural. Este período marca o renascimento da IA trazendo novas esperanças para sua aplicação prática.

1995: Stuart Russell e Peter Norvig publicam o livro Artificial Intelligence: A Modern Approach, que se tornaria uma das principais referências no estudo de IA abordando de forma abrangente os métodos e algoritmos fundamentais da área, consolidando-se como material de estudo essencial para pesquisadores e estudantes.

1997: O Deep Blue, supercomputador da IBM, derrota o campeão mundial de xadrez Garry Kasparov, um marco histórico que demonstra a capacidade da IA de resolver problemas altamente complexos e estratégicos, além de colocar a inteligência artificial sob os holofotes internacionais.

2004: John McCarthy, um dos pioneiros da IA, publica o livro What is Artificial Intelligence?, onde oferece uma definição amplamente adotada de IA, consolidando seu lugar como uma das referências essenciais no campo. Esse trabalho ajuda a esclarecer as discussões sobre o que constitui a inteligência artificial e como ela pode ser aplicada em diferentes contextos.

2011: O IBM Watson vence os campeões de Jeopardy!, Ken Jennings e Brad Rutter, demonstrando a incrível capacidade de um sistema de IA para processar linguagem natural e responder a perguntas de forma altamente precisa. Essa vitória representa um avanço significativo na capacidade de um sistema de IA lidar com informações não estruturadas e entender nuances linguísticas complexas.

2015: O supercomputador Minwa, desenvolvido pela Baidu, utiliza redes neurais convolucionais para categorizar imagens com uma precisão superior à humana, marcando uma grande conquista no campo da visão computacional e provando a eficácia do deep learning em tarefas de reconhecimento visual.

2016: O AlphaGo, da DeepMind, vence o campeão mundial de Go, Lee Sedol, em uma série de partidas. Esse feito, considerado uma das maiores conquistas da IA, demonstra a capacidade dos algoritmos de aprendizado profundo de lidar com jogos complexos, que exigem raciocínio estratégico avançado. A vitória também resulta na aquisição da DeepMind pelo Google por US$ 400 milhões.

2023: A chegada de grandes modelos de linguagem, como o ChatGPT, começa uma nova era na IA, com a IA generativa se tornando um fator transformador nas empresas e na sociedade. Esses modelos de deep learning, treinados com grandes quantidades de dados não rotulados, são capazes de criar textos, imagens e até códigos com uma habilidade impressionante, levando a IA a um novo patamar de aplicabilidade e popularidade.

A história da inteligência artificial, como podemos observar, é marcada por uma sucessão de descobertas, avanços e superações de limitações que permitiram à IA atingir um grau de sofisticação cada vez maior, com aplicações em uma variedade de campos, de jogos a diagnósticos médicos. Com o rápido desenvolvimento das tecnologias de aprendizado de máquina e deep learning, a IA continua a se expandir, prometendo uma revolução ainda maior nas próximas décadas.

# 3 METODOLOGIA

Este trabalho, quanto à sua natureza, caracteriza-se como uma pesquisa aplicada, cujo objetivo principal é utilizar o conhecimento científico para desenvolver uma solução prática para um problema específico. Esse tipo de pesquisa visa à aplicação direta dos resultados, contribuindo para a ciência e a tecnologia com uma finalidade prática. O estudo baseia-se em um estudo de caso, complementado pelo uso de materiais bibliográficos, como artigos científicos, que serviram como base para a fundamentação teórica e o desenvolvimento do sistema proposto.

A pesquisa bibliográfica desempenhou um papel essencial, reunindo conhecimento previamente produzido por meio da análise de artigos científicos. Esse processo possibilitou a construção de uma base teórica sólida, permitindo compreender os principais conceitos, ideias e teorias sobre o tema. Além disso, a revisão bibliográfica auxiliou na identificação de lacunas no conhecimento existente, permitindo direcionar o foco do estudo para preencher essas lacunas e contextualizar o trabalho dentro do panorama acadêmico atual.

A pesquisa experimental complementou a abordagem teórica, permitindo aplicar na prática os conceitos estudados. Por meio do desenvolvimento de um código, foi possível implementar um sistema que utiliza um banco de dados não relacional, explorando sua eficiência, escalabilidade e aplicabilidade. A experimentação foi fundamental para validar os fundamentos teóricos e demonstrar o funcionamento das tecnologias propostas, oferecendo uma visão prática de suas vantagens e limitações.

Essa abordagem combinada, que integra pesquisa bibliográfica e experimental, garantiu que o estudo fosse robusto e fundamentado, permitindo uma análise aprofundada que alia teoria e prática. A experimentação proporcionou resultados concretos e insights práticos, reforçando a relevância e a aplicabilidade dos conceitos apresentados.

## 4. DESENVOLVIMENTO

No começo do código foi feito uma sequência de *stopwords* que servem para evitar que o programa acabe mais cedo além da lista de categorias que é utilizada futuramente no programa

Figura 1 - Lista de stopwords

```
# Lista básica de stopwords em português
STOPWORDS_PORTUGUES = {
    "a", "o", "e", "de", "do", "da", "em", "um", "uma", "que", "com",
    "para", "por", "não", "no", "na", "os", "as", "se", "mais", "me",
    "te", "lhe", "nos", "vos", "eles", "elas", "este", "essa", "isso"
}

# Lista de categorias
categorias = [
    "Cereais", "Doces e Açúcares", "Bebidas", "Condimentos", "Limpeza",
    "Massas", "Laticínios", "Frios", "Padaria", "Doces e Snacks",
    "Enlatados", "Carnes", "Pet Shop", "Higiene"
]
```

Fonte: Próprio Autor

Após isso é feito um pré-processamento do texto. Essa função é utilizada no intuito de remover caracteres especiais além de filtrar as palavras que se encaixam no *stopword* ajudando assim a reduzir o ruído nos dados

Figura 2 - Pré-Processamento do texto

```
# Pré-processamento de texto
def preprocessar_texto(texto):
    # Remove caracteres especiais e coloca o texto em minúsculas
    texto = re.sub(r'[^\w\s]', '', texto.lower())
    # Tokeniza e remove stopwords
    palavras = texto.split()
    palavras_filtradas = [palavra for palavra in palavras if palavra not in STOPWORDS_PORTUGUES]
    return " ".join(palavras_filtradas)
```

Fonte: Próprio Autor

Nesta parte são carregados os dados do meu arquivo “.json” onde estão contidas as informações dos meus produtos.

Figura 3 - Carregando o JSON

```python
# Carregar os produtos do arquivo JSON
def carregar_produtos(arquivo_json):
    with open(arquivo_json, 'r', encoding='utf-8') as f:
        return json.load(f)
```

Fonte: Próprio Autor

Utilizando o Algoritmo TF-IDF (Term Frequency - Inverse Document Frequency) que transforma o texto de cada produto em um vetor numérico que pode ser usado para calcular a similaridade entre os produtos. Além de ajudar a ajustar a importância dos mesmos

Figura 4 - Criação da matriz de características

```python
# Criar matriz de características com pesos ajustáveis
def criar_matriz_caracteristicas(produtos, pesos=None):
    if pesos is None:
        pesos = {'nome': 1.0, 'descricao': 1.0, 'categoria': 1.0}

    textos = []
    for produto in produtos:
        nome = preprocessar_texto(produto['nome']) * int(pesos['nome'])
        descricao = preprocessar_texto(produto['descricao']) * int(pesos['descricao'])
        categoria = preprocessar_texto(produto['categoria']) * int(pesos['categoria'])
        textos.append(f"{nome} {descricao} {categoria}")

    tfidf = TfidfVectorizer(max_features=500)
    matriz_caracteristicas = tfidf.fit_transform(textos)
    return matriz_caracteristicas, tfidf
```

Fonte: Próprio Autor

O pipeline serve para poder organizar o fluxo de trabalho do modelo, garantindo que os dados sejam processados corretamente antes que sejam feitas as previsões. A normalização e a redução de dimensionalidade são especialmente importantes para melhorar a eficiência e a precisão da busca por produtos semelhantes.

Figura 5 - Configurando o pipeline

```python
# Configurar pipeline completo
def configurar_pipeline():
    pipeline = Pipeline([
        ('scaler', StandardScaler(with_mean=False)),  # Normalização dos dados
        ('reducer', TruncatedSVD(n_components=40)),  # Redução de dimensionalidade
        ('model', NearestNeighbors(n_neighbors=5, metric='cosine'))  # Modelo de busca
    ])
    return pipeline

# Treinar o pipeline
def treinar_pipeline(pipeline, matriz_caracteristicas):
    pipeline.fit(matriz_caracteristicas)
    return pipeline
```

Fonte: Próprio Autor

Neste print é criado a função consultar_produtos que executa a busca pelos produtos mais similares à consulta do usuário. Ela começa com o pré-processamento da consulta, convertendo-a em um vetor numérico com TF-IDF. Em seguida, o vetor da consulta passa pelo pipeline para transformação antes de ser comparado com os produtos usando o modelo NearestNeighbors. A função também aplica filtros adicionais, como categoria e preço máximo, para refinar os resultados da busca.

Figura 6 - Consulta dos produtos

```python
# Consultar produtos com suporte para filtros adicionais
def consultar_produtos(query, produtos, pipeline, tfidf, filtros=None):
    query_processada = preprocessar_texto(query)
    query_tfidf = tfidf.transform([query_processada])

    # Passa o vetor da consulta pelo pipeline (exceto a última etapa: modelo)
    query_transformada = pipeline[:-1].transform(query_tfidf)

    # Obter vizinhos mais próximos
    distancias, indices = pipeline['model'].kneighbors(query_transformada)
    resultados = [produtos[i] for i in indices[0]]

    # Aplicar filtros adicionais, se fornecidos
    if filtros:
        if 'categoria' in filtros and filtros['categoria']:
            categoria_desejada = filtros['categoria'].lower()
            resultados = [produto for produto in resultados if produto['categoria'].lower() == categoria_desejada]
        if 'preco_maximo' in filtros and filtros['preco_maximo'] > 0:
            resultados = [produto for produto in resultados if produto['preco'] <= filtros['preco_maximo']]

    return resultados
```

Fonte: Próprio Autor

Nessa parte é mostrado ao usuário as possíveis categorias para que ele possa escolher as disponíveis

Figura 7 - Solicitação das categorias

```
# Solicitar categoria e preço máximo
while True:
    # Exibe as categorias
    print("\nCategorias disponíveis:")
    for idx, categoria in enumerate(categorias, 1):
        print(f"{idx}. {categoria}")

    # Entrada de categoria
    categoria_filtro = input("\nDigite a categoria do produto que você procura: ").strip()

    # Verifica se a categoria não foi deixada em branco e é válida
    if categoria_filtro and categoria_filtro.lower() in [cat.lower() for cat in categorias]:
        break
    else:
        print("\nCategoria inválida ou em branco. Por favor, tente novamente.")
```

Fonte: Próprio Autor

A seguir são exibidos os produtos relacionados a categoria escolhida pelo usuário

Figura 8 - Exibição dos produtos relacionados

```
# Exibir produtos relacionados à categoria escolhida
produtos_categoria = [produto for produto in produtos if produto['categoria'].lower() == categoria_filtro.lower()]

if produtos_categoria:
    print("\nProdutos disponíveis nesta categoria:")
    for idx, produto in enumerate(produtos_categoria, 1):
        print(f"{idx}. {produto['nome']}")
else:
    print("\nNão há produtos disponíveis nesta categoria.")
```

Fonte: Próprio Autor

A seguir o usuário pode digitar o nome do produto desejado

Figura 9 - Escolha do produto

```
# Permitir que o usuário digite um produto ou parte do nome
produto_filtro = input("\nDigite o nome do produto que você procura (ou pressione Enter para ignorar): ").strip()
```

Fonte: Próprio Autor

Logo após ele pode definir um valor máximo para a busca

Figura 10 - Valor máximo

```
# Entrada do preço máximo
preco_filtro = input("Digite o preço máximo (deixe vazio para ignorar): ").strip()
if preco_filtro:
    preco_filtro = float(preco_filtro)
```

Fonte: Próprio Autor

A seguir é criado um dicionário chamado filtro em que armazena todas as os filtros aplicados pelo usuário

Figura 11 - Configuração dos filtros

```
# Configurando filtros
filtros = {'categoria': categoria_filtro}
if produto_filtro:
    filtros['produto'] = produto_filtro
if preco_filtro:
    filtros['preco_maximo'] = preco_filtro
```

Fonte: Próprio Autor

Após isso é realizado a consulta e exibido os resultados

Figura 12 - Resultados

```
# Realizar consulta
resultados = consultar_produtos(produto_filtro, produtos, pipeline, tfidf, filtros=filtros)

# Exibir resultados
if resultados:
    print("\nProdutos encontrados:")
    for produto in resultados:
        print(f"Nome: {produto['nome']}, Categoria: {produto['categoria']}, Preço: {produto['preco']:.2f}")
else:
    print("\nNenhum produto encontrado com os filtros aplicados.")
```

Fonte: Próprio Autor

## 5. RESULTADOS

Após o programa rodar ele mostra essa tela onde classifica todas as categorias e pede para o usuário escolher uma

Figura 13 - Escolha das categorias

```
Categorias disponíveis:
1. Cereais
2. Doces e Açúcares
3. Bebidas
4. Condimentos
5. Limpeza
6. Massas
7. Laticínios
8. Frios
9. Padaria
10. Doces e Snacks
11. Enlatados
12. Carnes
13. Pet Shop
14. Higiene

Digite a categoria do produto que você procura:
```

Fonte: Próprio Autor

Depois de escolher uma categoria ele vai para a tela em que pede para escolher um dos produtos disponíveis

Figura 14 - Escolha do produto

```
Digite a categoria do produto que você procura: carnes

Produtos disponíveis nesta categoria:
1. Filé de Frango
2. Linguiça Calabresa

Digite o nome do produto que você procura: 
```

Fonte: Próprio Autor

Após escolher o produto é pedido para o cliente escolher um valor máximo caso ele queira

Figura 15 - Valor máximo

```
Digite o nome do produto que você procura: filé de frango
Digite o preço máximo (deixe vazio para ignorar):
```

Fonte: Próprio Autor

Caso houvesse outros itens similares ele iria ser mostrado junto do item escolhido

Figura 16 - Resultado da busca

```
Produtos encontrados:
Nome: Filé de Frango, Categoria: Carnes, Preço: 16.99
PS C:\Users\LLPGames\Documents\TCC>
```

Fonte: Próprio Autor

Aqui está um outro exemplo em que o resultado mostra dois possíveis produtos para o cliente

Figura 17 - Segundo exemplo

```
Digite a categoria do produto que você procura: bebidas

Produtos disponíveis nesta categoria:
1. Café
2. Refrigerante
3. Água Mineral
4. Suco de Laranja
5. Chá Romã
6. Chá Verde

Digite o nome do produto que você procura:chá
Digite o preço máximo (deixe vazio para ignorar):

Produtos encontrados:
Nome: Chá Verde, Categoria: Bebidas, Preço: 9.50
Nome: Chá Romã, Categoria: Bebidas, Preço: 7.99
PS C:\Users\LLPGames\Documents\TCC>
```

Fonte: Próprio Autor

## 6. CONSIDERAÇÕES FINAIS

Este trabalho explorou o uso da Inteligência Artificial (IA) como ferramenta estratégica para otimizar sistemas de armazenamento e recuperação de dados em bancos de dados não relacionais. A IA tem se mostrado essencial para lidar com a crescente complexidade dos dados em aplicações modernas, permitindo a criação de algoritmos e modelos que potencializam a eficiência de consultas e o desempenho geral do sistema.

Ao integrar técnicas de IA, foi possível implementar soluções que analisam padrões de consulta e comportamento do usuário, ajustando dinamicamente a estrutura e a indexação do banco de dados para reduzir tempos de resposta e melhorar a experiência do usuário. Além disso, algoritmos de aprendizado de máquina foram aplicados para prever consultas frequentes e personalizar a apresentação dos resultados com base nas necessidades específicas de cada contexto. Essas abordagens destacaram o papel da IA como um diferencial no desenvolvimento de sistemas inteligentes e adaptativos.

No contexto específico deste trabalho, a aplicação de bancos de dados não relacionais, como aqueles baseados no formato JSON, revelou-se fundamental para atender às demandas de sistemas de busca. Sua flexibilidade permitiu uma modelagem de dados intuitiva e alinhada às necessidades de sistemas dinâmicos e escaláveis, complementando as capacidades da IA. Por exemplo, a estrutura menos rígida desses bancos possibilitou a integração eficiente com algoritmos de IA, que exploraram dados de forma mais ágil e precisa, sem as limitações impostas por esquemas tradicionais.

Durante o desenvolvimento, foram investigados os benefícios e desafios inerentes ao uso de bancos de dados não relacionais. Uma das principais vantagens observadas foi a capacidade de adaptação rápida a mudanças nos requisitos do sistema, essencial em cenários onde a IA aprende e ajusta constantemente os modelos de operação. Essa característica foi crucial para a implementação de filtros e buscas personalizadas, bem como para a categorização de dados como produtos, otimizando a interação entre o banco e os algoritmos de IA.

No entanto, também foram enfrentados desafios importantes, como o controle da redundância de dados, que pode aumentar custos de armazenamento e complicar atualizações. A implementação de soluções baseadas em IA ajudou a mitigar parte desses desafios, automatizando o gerenciamento de redundâncias e aprimorando a eficiência de consultas complexas, características que são limitadas em abordagens puramente relacionais.

As otimizações realizadas ao longo do projeto incluíram a configuração cuidadosa de mecanismos de indexação e cache, os quais, aliados a técnicas de IA, potencializaram o desempenho do sistema. Essa integração é especialmente relevante em aplicações que lidam com volumes crescentes de dados, como plataformas de e-commerce ou sistemas de recomendação, onde a IA não apenas processa os dados, mas também gera insights para decisões automatizadas em tempo real.

Por fim, este trabalho demonstrou que a combinação de IA com bancos de dados não relacionais cria uma solução robusta e adaptável para sistemas modernos. A utilização de IA não apenas amplia as capacidades de processamento e consulta, mas também proporciona uma camada de inteligência que transforma a interação com os dados em algo mais eficiente e personalizado.

Como perspectivas futuras, destaca-se a exploração de algoritmos mais avançados para análise preditiva de consultas e otimização de índices. A integração com serviços de nuvem e técnicas de particionamento de dados também apresenta oportunidades significativas, especialmente em contextos de big data. Além disso, comparações mais detalhadas entre abordagens relacionais e não relacionais em cenários com uso intensivo de IA poderiam enriquecer a compreensão sobre as melhores práticas para diferentes aplicações.

Em síntese, este projeto reafirma a relevância da integração entre IA e bancos de dados não relacionais como uma abordagem indispensável para o desenvolvimento de soluções escaláveis e de alto desempenho, atendendo às demandas de sistemas que necessitam combinar inteligência, flexibilidade e eficiência no gerenciamento de dados.

**REFERÊNCIAS**

Linkedin. Como você usa o aprendizado de máquina para melhorar a otimização de consultas? https://pt.linkedin.com/advice/1/how-do-you-use-machine-learning-improve-query?lang=pt

IBM. O que é inteligência artificial? https://www.ibm.com/br-pt/topics/artificial-intelligence

IBM. O que é aprendizado de máquina (ML)? https://www.ibm.com/br-pt/topics/machine-learning

AWS, amazon. O que é NoSQL? https://aws.amazon.com/pt/nosql/

Tracker, Tipos de Inteligência Artificial: ANI, AGI e ASI https://iatracker.com.br/tipos-de-inteligencia-artificial-ani-agi-e-asi/

Santos, Vinicius dos. Term Frequency (TF) e Term Frequency – Inverse Document Frequency (TF-IDF) https://www.computersciencemaster.com.br/term-frequency-e-term-frequency-inverse-document-frequency/