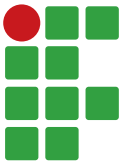

INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Goiás

Bacharelado em Sistemas de Informação

Manoel Augusto Dos Santos Ferreira

Comparativo de Desempenhos de Armazenamento de Dados de Dispositivos IoT Usando Banco de Dados Relacionais e Não Relacionais

Luziânia
2025

TERMO DE AUTORIZAÇÃO PARA DISPONIBILIZAÇÃO NO REPOSITÓRIO DIGITAL DO IFG - ReDi IFG

Com base no disposto na Lei Federal nº 9.610/98, AUTORIZO o Instituto Federal de Educação, Ciência e Tecnologia de Goiás, a disponibilizar gratuitamente o documento no Repositório Digital (ReDi IFG), sem ressarcimento de direitos autorais, conforme permissão assinada abaixo, em formato digital para fins de leitura, download e impressão, a título de divulgação da produção técnico-científica no IFG.

Identificação da Produção Técnico-Científica

[ ] Tese	[ ] Artigo Científico
[ ] Dissertação	[ ] Capítulo de Livro
[ ] Monografia – Especialização	[ ] Livro
[X] TCC - Graduação	[ ] Trabalho Apresentado em Evento
[ ] Produto Técnico e Educacional - Tipo: _____	

Nome Completo do Autor: Manoel Augustos dos Santos Ferreira
Matrícula: 20191080080269
Título do Trabalho: Comparativo de Desempenhos de Armazenamento de Dados de Dispositivos IoT Usando Banco de Dados Relacionais e Não Relacionais

Autorização - Marque uma das opções

1. (X) Autorizo disponibilizar meu trabalho no Repositório Digital do IFG (acesso aberto);
2. ( ) Autorizo disponibilizar meu trabalho no Repositório Digital do IFG somente após a data ___/___/_____ (Embargo);
3. ( ) Não autorizo disponibilizar meu trabalho no Repositório Digital do IFG (acesso restrito).

Ao indicar a opção **2 ou 3**, marque a justificativa:

( ) O documento está sujeito a registro de patente.
( ) O documento pode vir a ser publicado como livro, capítulo de livro ou artigo.
( ) Outra justificativa: _____

DECLARAÇÃO DE DISTRIBUIÇÃO NÃO-EXCLUSIVA

O/A referido/a autor/a declara que:

i. o documento é seu trabalho original, detém os direitos autorais da produção técnico-científica e não infringe os direitos de qualquer outra pessoa ou entidade;
ii. obteve autorização de quaisquer materiais inclusos no documento do qual não detém os direitos de autor/a, para conceder ao Instituto Federal de Educação, Ciência e Tecnologia de Goiás os direitos requeridos e que este material cujos direitos autorais são de terceiros, estão claramente identificados e reconhecidos no texto ou conteúdo do documento entregue;
iii. cumpriu quaisquer obrigações exigidas por contrato ou acordo, caso o documento entregue seja baseado em trabalho financiado ou apoiado por outra instituição que não o Instituto Federal de Educação, Ciência e Tecnologia de Goiás.

Luziânia, 10/09/25
Local Data

gov.br
Documento assinado digitalmente
MANOEL AUGUSTO DOS SANTOS FERREIRA
Data: 10/09/2025 08:54:49-0300
Verifique em https://validar.iti.gov.br

_____
Assinatura do Autor e/ou Detentor dos Direitos Autorais

Instituto Federal de Educação, Ciência e Tecnologia de Goiás
Bacharelado em Sistemas de Informação

Manoel Augusto Dos Santos Ferreira

Comparativo de Desempenhos de Armazenamento de Dados de Dispositivos IoT Usando Banco de Dados Relacionais e Não Relacionais

Trabalho de Conclusão de Curso 2 apresentado à Coordenação do Curso de Bacharelado em Sistemas de Informação do Instituto Federal de Educação, Ciência e Tecnologia de Goiás - Câmpus Luziânia, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Orientador: Me. Audir da Costa Oliveira Filho

Coorientador: Dr. Ulisses Rodrigues Afonseca

Luziânia

2025

F383c Ferreira, Manoel Augusto dos Santos

Comparativo de desempenhos de armazenamento de dados de dispositivos IoT usando banco de dados relacionais e não relacionais / Manoel Augusto dos Santos Ferreira. – Luziânia: IFG, 2025.
64 f. : il. color.

Orientador: Prof. Me. Audir da Costa Oliveira Filho.
Coorientador: Prof. Dr. Ulisses Rodrigues Afonseca.
TCC (Graduação - Bacharelado em Sistemas de Informação). Instituto Federal de Goiás - IFG, Câmpus Luziânia, 2025.

1. Computação 2. Armazenamento de dados 3. Internet das coisas 4. Banco de dados relacionais 5 Banco de dados não relacionais I. Título. II. Oliveira Filho, Audir da Costa, orient. III. Afonseca, Ulisses Rodrigues, coorient.

CDD 004

Bibliotecário: Helio Lino Delfino – CRB-1/3031

MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE GOIÁS
CÂMPUS LUZIÂNIA

Manoel Augusto dos Santos Ferreira

Comparativo de Desempenho de Armazenamento de Dados de Dispositivos IoT Usando Banco de Dados Relacionais e Não Relacionais

Trabalho de Conclusão de Curso apresentado à Coordenação do Curso de Bacharelado em Sistemas de Informação do Instituto Federal de Goiás - Campus Luziânia, como parte das exigências para obtenção do título de Bacharel em Sistemas de Informação.

Aprovada em 20 de fevereiro de 2025.

Professor Orientador: Me. Audir da Costa Oliveira Filho - IFG/Luziânia

Professor Coorientador: Dr. Ulisses Rodrigues Afonseca - IFG/Luziânia

Professor: Dr. Lucas de Almeida Ribeiro - IFG/Luziânia

Professor: Me. Luila Moraes de Oliveira - IFG/Luziânia

A todos que contribuíram para a realização deste projeto, com especial reconhecimento aos familiares, amigos e professores que ofereceram apoio, orientação e encorajamento ao longo desta jornada. Agradecimentos são também dirigidos a todos que, de forma direta ou indireta, participaram do processo de elaboração e conclusão deste trabalho.

AGRADECIMENTOS

Agradeço ao Instituto Federal de Goiás (IFG) pela infraestrutura e apoio acadêmico oferecidos. Um agradecimento ao meu orientador, Audir, e ao coorientador, Ulisses, pela orientação e suporte fundamentais na elaboração deste trabalho.

Expresso também minha gratidão aos colegas e à minha família pelo apoio e incentivo constantes.

"O sucesso nada mais é que ir de fracasso em fracasso sem que se perca o entusiasmo."
(Winston Churchill)

RESUMO

FERREIRA, M. A. S.. **Comparativo de Desempenhos de Armazenamento de Dados de Dispositivos IoT Usando Banco de Dados Relacionais e Não Relacionais. Luziânia, 2025. 64p. Trabalho de Conclusão de Curso 2. Instituto Federal de Educação, Ciência e Tecnologia de Goiás**

Este trabalho tem como objetivo comparar o desempenho de bancos de dados relacionais e não relacionais no armazenamento de dados gerados por dispositivos IoT (Internet das Coisas). A escolha do tipo de banco de dados é crucial para garantir eficiência e escalabilidade em aplicações IoT, uma vez que os bancos relacionais, embora amplamente utilizados, podem apresentar limitações no gerenciamento de grandes volumes de dados e na demanda por recursos. Por outro lado, os bancos não relacionais são projetados para lidar com grandes quantidades de dados, destacando-se pela eficiência e facilidade de uso em cenários de alta escalabilidade. Para isso, o estudo adota uma abordagem abrangente, combinando revisão sistemática da literatura com experimentos práticos. A revisão sistemática analisa de forma rigorosa os estudos existentes sobre o desempenho desses bancos de dados, enquanto os experimentos práticos avaliam o comportamento de soluções como PostgreSQL (banco relacional) e MongoDB (banco não relacional) em cenários simulados de coleta e consulta de dados IoT. As métricas de desempenho incluem tempo de resposta, taxa de transferência e consumo de recursos. Os resultados obtidos fornecem uma base para a escolha adequada de tecnologias de armazenamento, considerando as particularidades e demandas específicas de cada aplicação IoT.

Palavras-chave: Banco de Dados Relacionais, IoT, Banco de Dados Não Relacionais, NoSQL, SQL, PostgreSQL, MongoDB.

ABSTRACT

This study aims to compare the performance of relational and non-relational databases in storing data generated by IoT (Internet of Things) devices. The choice of database type is critical to ensure efficiency and scalability in IoT applications, as relational databases, although widely used, may present limitations in managing large volumes of data and resource demands. On the other hand, non-relational databases are designed to handle large amounts of data, standing out for their efficiency and ease of use in high-scalability scenarios. To achieve this, the study adopts a comprehensive approach, combining a systematic literature review with practical experiments. The systematic review rigorously analyzes existing studies on the performance of these databases, while the practical experiments evaluate the behavior of solutions such as PostgreSQL (relational database) and MongoDB (non-relational database) in simulated IoT data collection and query scenarios. Performance metrics include response time, throughput, and resource consumption. The results provide a basis for selecting appropriate storage technologies, considering the specificities and demands of each IoT application.

Keywords: **Keywords**: Rel Relational Databases, IoT, Non-Relational Databases, NoSQL, SQL, PostgreSQL, MongoDB.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE CÓDIGOS

LISTA DE ABREVIATURAS E SIGLAS

ACID	Atomicidade, Consistência, Isolamento, Durabilidade
BDNR	Banco de Dados de Não Relacionais
BDR	Banco de Dados de Relacionais
CoAP	Constrained Application Protocol
IOT	Internet of Things
JSON	JavaScript Object Notation
MQTT	Message Queuing Telemetry Transport
NFC	Near Field Communication
NoSQL	Not Structured Query Languag
RFID	Radio Frequency Identification
SGBDR	Sistema de Gestão de Dados Relacional
SQL	Structured Query Language
SRL	Systematic Literature Review

SUMÁRIO

1 INTRODUÇÃO

O crescente uso de dispositivos da Internet das Coisas (IoT, do inglês *Internet of Things*) tem gerado um aumento exponencial na quantidade de dados produzidos, o que revisita a necessidade de estratégias eficientes para o armazenamento e gerenciamento dessas informações. Com o volume de dados gerado por esses dispositivos se expandindo rapidamente, torna-se crucial identificar as plataformas de armazenamento mais adequadas para atender a essa demanda. Nesse contexto, a escolha entre bancos de dados relacionais e não relacionais emerge como uma decisão fundamental, uma vez que cada abordagem apresenta características distintas que podem impactar diretamente o desempenho e a escalabilidade das aplicações IoT.

Os bancos de dados relacionais, amplamente adotados em diversos cenários, são conhecidos por sua estrutura rígida e pela exigência de normalização, o que garante consistência e integridade dos dados. No entanto, em ambientes IoT, onde o volume de dados é massivo e dinâmico, essa rigidez pode se tornar um obstáculo, limitando a escalabilidade e a flexibilidade necessárias para lidar com a diversidade e a velocidade dos dados gerados. Por outro lado, os bancos de dados não relacionais, projetados para lidar com grandes volumes de informações não estruturadas ou semiestruturadas, destacam-se pela eficiência em cenários de alta escalabilidade e pela facilidade de adaptação a mudanças na estrutura dos dados.

A pesquisa proposta visa investigar as vantagens e limitações dos bancos de dados relacionais e não relacionais no desempenho em ambientes de IoT. Serão comparadas soluções representativas de cada categoria, como o PostgreSQL (banco relacional) e o MongoDB (banco não relacional), em cenários simulados de coleta e inserção de dados IoT. A análise se concentrará em métricas de desempenho, como tempo de resposta, taxa de transferência, consumo de recursos (CPU, memória e I/O de disco) e utilização de banda, com o objetivo de avaliar a eficiência de cada abordagem no contexto de aplicações IoT.

Além disso, a modelagem de dados para IoT impõe desafios distintos dependendo do tipo de banco de dados utilizado. Enquanto os bancos relacionais podem enfrentar dificuldades relacionadas à escalabilidade e flexibilidade, os bancos não relacionais podem oferecer vantagens significativas em cenários que exigem alta disponibilidade e adaptabilidade. Este estudo busca, portanto, fornecer uma análise comparativa detalhada, destacando as diferenças e semelhanças entre os dois tipos de banco de dados, com o intuito de oferecer percepções valiosas para a escolha de soluções mais eficientes e escaláveis no armazenamento e processamento de dados em sistemas IoT.

Por fim, a relevância deste trabalho reside na contribuição para a discussão aca-

dêmica e prática sobre a seleção de tecnologias de armazenamento em ambientes IoT, oferecendo uma base sólida para decisões técnicas que visam otimizar o desempenho e atender às demandas específicas de cada aplicação.

1.1 Objetivo geral

O objetivo principal deste estudo é explorar como diferentes tipos de bancos de dados lidam com o armazenamento dos dados gerados por dispositivos IoT, analisando o desempenho dos bancos de dados relacionais e não relacionais em termos de uso de CPU, I/O de disco, memória, taxa de transferência e utilização de banda.

1.2 Objetivos Específicos

A seguir, apresentam-se os objetivos específicos deste trabalho, os quais orientarão as etapas da pesquisa e contribuirão para o alcance dos resultados esperados.

- Realizar uma revisão sistemática do tema, com base em artigos científicos relevantes.
- Conduzir testes práticos para levantar evidências sobre o desempenho de bancos de dados relacionais e não relacionais.
- Modelar o banco de dados para cada abordagem de armazenamento de dados.
- Gerar dados por meio da coleta sistemática utilizando sensores ou simuladores.
- Executar testes de benchmarking para comparar o desempenho dos bancos de dados relacionais e não relacionais.

1.3 Problema e Hipótese

O conceito de Internet das Coisas surgiu nos anos 1990, quando Kevin Ashton, um dos pioneiros da tecnologia, utilizou o termo para descrever a conexão de dispositivos físicos à internet com o objetivo de coletar e compartilhar dados (ASHTON, 2009a). Desde então, o IoT experimentou um crescimento exponencial, impulsionado pela integração de sensores e dispositivos em setores variados, como saúde, transporte e manufatura (GUBBI et al., 2013). Esse crescimento vertiginoso resultou em uma crescente necessidade de estratégias de armazenamento e proteção de dados, uma vez que a coleta massiva de informações levanta preocupações substanciais quanto à segurança e privacidade (KUMAR; MALLICK; GUPTA, 2016). A proteção e a integridade dos dados tornaram-se essenciais para garantir a confiabilidade e a confidencialidade das informações geradas por dispositivos conectados,

exigindo, assim, a adoção de soluções robustas por parte das organizações (ZHANG et al., 2020).

O aumento significativo no uso de dispositivos IoT implicou na produção de volumes imensos de dados, o que gerou a demanda por soluções de armazenamento mais eficientes e escaláveis. Tradicionalmente, bancos de dados relacionais, com suas estruturas bem definidas e um modelo baseado em tabelas, têm sido o padrão adotado pela indústria por décadas. No entanto, a natureza dos dados gerados por dispositivos IoT, frequentemente semi-estruturados ou não estruturados, e sua constante evolução desafiam os modelos tradicionais de armazenamento (CHEN et al., 2017). Nesse contexto, os bancos de dados relacionais e não relacionais (NoSQL) apresentam vantagens e desvantagens que devem ser analisadas para determinar a solução mais adequada a diferentes tipos de aplicação em IoT.

Os bancos de dados relacionais são amplamente reconhecidos pela sua capacidade de garantir a integridade e a consistência dos dados, atributos cruciais para muitas aplicações que dependem da precisão e confiabilidade das informações. De acordo com Elmasri (ELMASRI; NAVATHE, 2020), a estrutura tabular e o uso de esquemas predefinidos tornam os bancos de dados relacionais ideais para cenários onde a realização de consultas complexas e a manutenção da integridade referencial são essenciais, como em sistemas de monitoramento de saúde ou no controle de processos industriais. Essa abordagem rígida é vantajosa em ambientes onde a consistência e a estrutura dos dados são prioridades.

No entanto, embora os bancos de dados relacionais ofereçam muitas vantagens, eles apresentam desafios significativos ao lidar com o armazenamento e processamento de dados provenientes de dispositivos IoT. A rigidez dos esquemas pode ser uma limitação importante, pois os dados gerados por esses dispositivos são frequentemente semi-estruturados ou não estruturados, variando consideravelmente em formato e complexidade (HE; HU; ZHONG, 2016). Além disso, a escalabilidade vertical, característica dos bancos de dados relacionais, pode se tornar um obstáculo quando se trata de processar a grande quantidade de dados gerados em sistemas distribuídos de IoT, prejudicando o desempenho em operações de leitura e escrita em grandes volumes de dados.

Por outro lado, os bancos de dados NoSQL oferecem uma maior flexibilidade em termos de modelagem de dados, permitindo o armazenamento de dados semi-estruturados ou não estruturados sem a necessidade de um esquema fixo (CATTELL, 2011). Embora essa flexibilidade seja uma vantagem para IoT, a ausência de um esquema rígido pode acarretar desafios em termos de integridade e consistência dos dados. Além disso, apesar de sua escalabilidade horizontal ser um ponto forte, a complexidade na gestão e manutenção desses bancos pode aumentar, já que o modelo distribuído e a natureza flexível podem levar a problemas de consistência, especialmente em ambientes de grande escala (STONEBRAKER et al., 2010).

Entretanto, essa flexibilidade vem acompanhada de desafios, particularmente em relação à manutenção da consistência e integridade dos dados. A ausência de um esquema predefinido pode complicar a modelagem dos dados, exigindo maior especialização no gerenciamento e aumentando a complexidade dos sistemas (HAN; KAMBER; PEI, 2011). Em alguns casos, essas dificuldades podem resultar em uma perda de desempenho e confiabilidade, especialmente em equipes com menos experiência em gerenciar bancos de dados NoSQL. A justificativa para essa pesquisa reside na necessidade de compreender de forma aprofundada as vantagens e limitações de cada tipo de banco de dados, de modo a fornecer uma orientação clara para a escolha da solução de armazenamento mais eficaz em contextos de IoT. Considerando a crescente diversidade e volume de dados gerados por dispositivos conectados, é essencial que as organizações possam escolher a plataforma mais adequada para gerenciar essas informações, garantindo ao mesmo tempo, a eficiência, a confiabilidade e a escalabilidade necessárias para atender às demandas específicas de cada aplicação IoT.

1.4 Justificativa

A comparação do desempenho de armazenamento de dados de dispositivos IoT utilizando bancos de dados relacionais e não relacionais nos fornece informações cruciais sobre a eficácia e viabilidade de cada abordagem. O objetivo desta pesquisa é identificar a solução mais apropriada para o gerenciamento do volume e da complexidade dos dados coletados, levando em consideração as crescentes exigências de armazenamento e processamento associadas aos dispositivos da Internet das Coisa.

Os bancos de dados relacionais oferecem importantes vantagens no contexto da consistência e integridade dos dados. Eles fornecem um ambiente estruturado e robusto para o gerenciamento das informações, com uma capacidade intrínseca de organizar dados em tabelas e estabelecer relações complexas entre diferentes conjuntos de informações. Essa estruturação é essencial para manter relações claras e bem definidas, garantindo a confiabilidade dos resultados e facilitando a realização de consultas complexas. Os bancos de dados não relacionais se destacam por sua capacidade de lidar com excesso de dados não estruturados, como os gerados por sensores e dispositivos IoT, que frequentemente produzem informações diversificadas e não padronizadas. Esses bancos de dados representam uma escolha prática para lidar com a escalabilidade e a velocidade de processamento exigidas pelos ambientes de IoT em constante mudança, graças à sua capacidade de expansão horizontal e facilidade de distribuição.

A coleta e análise dos dados resultará em evidências empíricas que desempenharão um papel fundamental na tomada de decisão sobre a escolha do banco de dados mais apropriado. Ao investigar o desempenho de armazenamento de dados de dispositivos

IoT, tanto em ambientes relacionais quanto não relacionais, suprirá novas perspectivas que permitirão uma melhor compreensão e eficácia dessas abordagens. Isso aprimorará a capacidade de tomar decisões informadas sobre a seleção de bancos de dados específicos, pois os dados empíricos oferecerão uma visão crítica de como cada sistema responde às demandas específicas dos dispositivos IoT. Em última análise, os resultados desta pesquisa não apenas contribuirão para o conhecimento sobre o tema, mas também fornecerão uma base sólida para a escolha de soluções de armazenamento de dados que otimizem o desempenho e a eficiência em cenários relacionados a dispositivos IoT.

1.5 Metodologia

A metodologia é fundamental para garantir a robustez e a validade dos resultados obtidos. O primeiro passo consiste na realização de uma revisão sistemática da literatura, uma abordagem rigorosa e estruturada que foi realizada para identificar, avaliar e sintetizar as evidências existentes sobre o tema escolhido. A revisão sistemática permitirá uma compreensão aprofundada do estado atual da pesquisa, destacando tendências, lacunas e direções futuras.Para isso, foram utilizadas as principais bases de dados acadêmicas, incluindo IEEE Xplore, Scopus, Google Scholar e ACM Digital Library. A busca foi conduzida com strings de busca cuidadosamente formuladas para garantir a relevância e a abrangência dos artigos selecionados.

A metodologia inclui a definição de critérios claros para a inclusão e exclusão de estudos, uma triagem inicial dos títulos e resumos, e uma análise completa dos artigos selecionados. A extração de dados será realizada com o auxílio da ferramenta Parsifal, que permitirá a organização eficiente dos estudos e a análise detalhada das informações. A síntese dos dados permitirá a identificação de padrões e tendências, fornecendo uma base sólida para a discussão dos resultados e a formulação de conclusões. Esta abordagem metodológica garantirá a integridade e a profundidade da revisão sistemática, estabelecendo um alicerce sólido para as demais etapas da pesquisa.

Para avaliar o desempenho de bancos de dados relacionais e não relacionais em aplicativos IoT, é fundamental conduzir testes práticos que forneçam evidências concretas para sustentar comparações e análises. Esses testes devem ser conduzidos com base em objetivos específicos, como avaliar a eficiência no armazenamento e a capacidade de lidar com grandes volumes de dados gerados por dispositivos IoT. Além disso, será necessário estabelecer critérios claros para os testes, a fim de garantir que as comparações entre os diferentes tipos de bancos de dados sejam justas, consistentes e relevantes, considerando a necessidade de persistência dos dados em IoT, essencial para a análise contínua e tomada de decisões baseadas nas informações geradas por sensores e dispositivos conectados.

Os testes práticos envolverão a implementação e execução de uma série de cenários

que simulam diferentes cargas de trabalho e tipos de consultas que esses bancos de dados devem gerenciar. Para bancos de dados relacionais, como MySQL e *PostgreSQL*, os testes incluirão operações como inserção, atualização, exclusão e consultas complexas envolvendo múltiplas tabelas e relacionamentos. Já para bancos de dados não relacionais, como MongoDB e Cassandra, os testes focarão em operações de leitura e escrita em grande escala, consultas por chave-valor e a eficiência no manejo de grandes volumes de dados sem esquema fixo.

Durante os testes, serão coletados dados sobre tempos de resposta, *throughput* (taxa de processamento de operações), e uso de recursos como CPU e memória. Além disso, será avaliada a escalabilidade de cada banco de dados sob diferentes cargas de trabalho para entender como o desempenho varia conforme o aumento da demanda. Os resultados obtidos serão analisados para identificar pontos fortes e fracos de cada tipo de banco de dados, fornecendo evidências empíricas que ajudarão a escolher a melhor solução de armazenamento para diferentes necessidades e cenários.

Para modelar um banco de dados de forma eficaz, é essencial adaptar a abordagem conforme o tipo de banco de dados, seja relacional ou não relacional. No caso de bancos de dados relacionais, o processo começa com a identificação das entidades e seus relacionamentos, como clientes e pedidos, e a criação de tabelas correspondentes. Cada tabela deve ter uma chave primária e pode incluir chaves estrangeiras para definir as relações entre tabelas. A normalização é aplicada para evitar redundâncias e garantir a integridade dos dados, e índices são criados para otimizar o desempenho das consultas. Um diagrama entidade-relacionamento pode ser desenvolvido para visualizar a estrutura e facilitar a implementação.

Para bancos de dados não relacionais, como os modelos de documentos, chave-valor, coluna ou grafos, a modelagem é adaptada às suas características específicas. Em um banco de dados de documentos, os dados são armazenados como documentos JSON ou BSON, que podem incluir dados embutidos ou referenciados. Em um banco de dados chave-valor, cada chave única é associada a um valor simples ou complexo, enquanto em bancos de dados de coluna, os dados são armazenados em famílias de colunas, permitindo consultas eficientes em grandes volumes de dados. Para bancos de dados de grafos, a modelagem envolve nós e relacionamentos, refletindo as interconexões entre entidades. Em todos os casos, a modelagem deve considerar as necessidades específicas de consulta e armazenamento para garantir a eficiência e a adequação do banco de dados ao propósito pretendido.

A geração de dados por meio da coleta sistemática, utilizando sensores ou simuladores, requer um planejamento rigoroso para assegurar a qualidade e a relevância das informações obtidas. A primeira etapa desse processo envolve a definição clara dos objetivos da coleta de dados, especificando quais parâmetros precisam ser monitorados e

quais hipóteses serão testadas. Esta fase orientará a seleção dos sensores ou simuladores mais adequados para atender às necessidades do estudo em questão.

A utilização de sensores será fundamental para medir com precisão os parâmetros de interesse, como temperatura, umidade, pressão, entre outros fatores específicos ao contexto da pesquisa. Esses sensores devem ser estrategicamente posicionados e devidamente calibrados para garantir a precisão das medições. A coleta de dados deverá ocorrer em intervalos regulares e em condições controladas, a fim de garantir a consistência e a comparabilidade dos dados ao longo do tempo, favorecendo a integridade dos resultados e a análise posterior.

A coleta sistemática deve incluir o registro detalhado de todas as condições experimentais e metodológicas, o que facilitará a análise e interpretação dos dados. Além disso, é importante garantir a integridade dos dados por meio de verificações e validações regulares, além de adotar procedimentos de armazenamento e gerenciamento adequados para proteger e organizar os dados coletados.

Para comparar o desempenho de bancos de dados relacionais e não relacionais, é fundamental realizar uma análise prática em condições reais de uso. Este processo envolve a execução de uma série de testes que simulam as operações mais comuns de cada tipo de banco de dados, para entender como eles se comportam sob diferentes tipos de carga e condições.

Esses testes devem incluir operações típicas, como inserções e atualizações, além de atividades mais desafiadoras, como transações grandes. Para bancos de dados relacionais, por exemplo, é interessante testar o desempenho em transações com múltiplas inserções de dados e a complexidade envolvida, enquanto bancos de dados não relacionais podem ser testados focando em operações com grandes volumes de dados em leitura e escrita.

O objetivo é observar o comportamento de cada sistema em cenários variados e identificar qual banco de dados se adapta melhor a diferentes tipos de demandas, oferecendo informações sobre a eficiência em tarefas específicas e sua capacidade de lidar com escalabilidade e grandes volumes de dados.

Documentar os resultados e fazer uma análise crítica. Apresente os dados de forma clara, utilizando gráficos e tabelas para ilustrar as comparações. Discuta as implicações dos resultados, considerando fatores como a escalabilidade, a flexibilidade e a adequação de cada banco de dados ao seu caso de uso específico.

Essa abordagem sistemática permitirá uma comparação objetiva e detalhada do desempenho de bancos de dados relacionais e não relacionais, ajudando a tomar decisões informadas sobre qual sistema é mais adequado para suas necessidades.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, será apresentada a fundamentação teórica que embasa o desenvolvimento da pesquisa. Serão discutidos os conceitos fundamentais relacionados aos bancos de dados relacionais e não relacionais, suas principais características, vantagens e desvantagens, além das abordagens e metodologias existentes para análise de desempenho. O objetivo é fornecer uma base sólida para compreender as questões e os desafios que serão abordados ao longo do trabalho.

2.1 Internet das Coisas (IoT)

A Internet das Coisas transcende o conceito de uma mera tendência tecnológica, configurando-se como uma revolução que transforma a interação entre os indivíduos e o ambiente em que vivem. Originada pela convergência de diversas tecnologias, como sensores inteligentes, redes sem fio e computação em nuvem, a IoT estabelece um ecossistema dinâmico no qual objetos físicos estão interconectados, permitindo a coleta, troca e análise de dados em tempo real (GUBBI et al., 2013).

A chave para o sucesso da IoT reside na capacidade de coletar dados significativos em tempo real e transformá-los em Compreensão acionáveis. A IoT está se tornando cada vez mais onipresente em nossas vidas, impulsionando inovações em uma variedade de setores, desde saúde e agricultura até transporte e varejo. No entanto, com essas oportunidades surgem desafios significativos, como preocupações com privacidade e segurança, interoperabilidade de dispositivos e padrões de comunicação (ATZORI; IERA; MORABITO, 2010a).

A Figura 1 apresenta o conceito de Internet das Coisas (IoT), delimitando seus principais eixos de atuação. Ao desmembrar a IoT em três grandes áreas de comunicação, a imagem nos permite compreender a amplitude e a profundidade dessa tecnologia, que está revolucionando a forma como nos comunicamos com o mundo ao nosso redor. A seguir, serão apresentados os três principais eixos de comunicação da IoT:

- **Comunicação a qualquer momento:** essa primeira área enfatiza a característica de ubiquidade da IoT. Seja em movimento, durante o dia ou à noite, os dispositivos conectados estão sempre disponíveis para comunicação. Isso possibilita, por exemplo, o monitoramento remoto da saúde, o controle de eletrodomésticos por aplicativos e a otimização de rotas em tempo real.
- **Comunicação em qualquer lugar:** A segunda área demonstra a natureza global

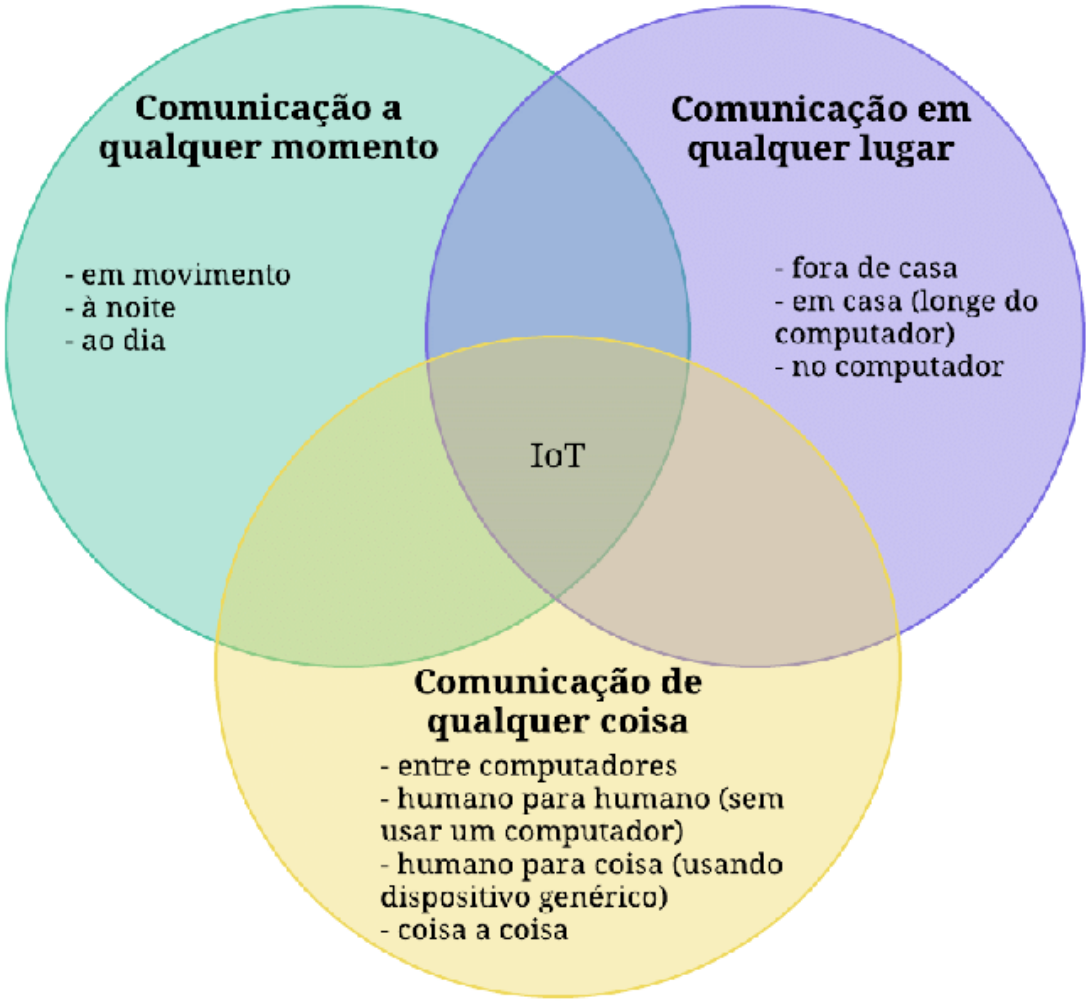

Figura 1 – Principais características que um sistema IoT.

Fonte: https://www.researchgate.net/.

da IoT. Seja dentro ou fora de casa, no computador ou em qualquer outro dispositivo, a comunicação é possível. Essa característica permite a criação de cidades inteligentes, a gestão eficiente de recursos e a otimização de processos em diversas áreas, como indústria e agricultura.

- **Comunicação de qualquer coisa:** por fim, a terceira área destaca a diversidade de elementos que podem ser interconectados na IoT. A comunicação pode ocorrer entre computadores, entre pessoas (sem a necessidade de um computador) e entre pessoas e coisas (utilizando dispositivos genéricos). Além disso, a comunicação entre dispositivos (coisa a coisa) possibilita a criação de sistemas autônomos e a automação de processos.

À medida que a IoT continua a se expandir e evoluir, é fundamental abordar esses desafios de maneira proativa, garantindo que as tecnologias sejam desenvolvidas e implementadas de forma ética e responsável.

2.1.1 Origens e Motivações

A concepção da Internet das Coisas remonta a uma ideia simples, mas revolucionária: conectar objetos do cotidiano à internet, equipando-os com sensores e capacidade de comunicação. Em 1999, Kevin Ashton, um pioneiro no campo da tecnologia, cunhou o termo IoT, vislumbrando um futuro no qual os objetos ao nosso redor estariam interconectados e poderiam compartilhar informações de forma autônoma (ASHTON, 2009b).

Segundo Naskar, a abordagem da Internet das Coisas (IoT) revolucionou a gestão da cadeia de suprimentos, otimizando processos e reduzindo custos ao fornecer informações em tempo real sobre produtos e ativos. A visão central da IoT, em seus estágios iniciais, era impulsionar tecnologias emergentes, como a Identificação por Radiofrequência (RFID), com o objetivo de viabilizar o rastreamento e a gestão de objetos em tempo real. Por exemplo, seria possível monitorar a localização de um pacote em trânsito ou o estado de máquinas industriais à distância. Essa proposta visava oferecer uma compreensão mais profunda e imediata do ambiente físico, permitindo que empresas e indivíduos tomassem decisões mais informadas e eficientes (NASKAR; BASU; SEN, 2017).

A motivação por trás da Internet das Coisas vai além da conveniência cotidiana. Ela está profundamente enraizada na busca por eficiência operacional e automação, visando melhorar processos existentes e explorar novas oportunidades de negócios. A IoT permite a conexão de objetos do dia a dia à internet, facilitando a coleta de dados em uma escala sem precedentes. Esses dados geram percepções que podem ser utilizadas para otimizar operações, aprimorar a qualidade de vida e impulsionar a inovação tecnológica (BRáS; PEREIRA; MORO, 2023).

2.1.2 Principais Características

A IoT é uma tecnologia que se destaca por sua capacidade de coletar, transmitir e analisar dados provenientes de uma variedade de dispositivos conectados à internet. Uma das principais características que define a IoT é sua interoperabilidade, que permite a comunicação entre diferentes dispositivos e plataformas, independentemente do fabricante ou do protocolo de comunicação utilizado (SHENG; TANG; YAO, 2013). Isso significa que um sensor de temperatura fabricado por uma empresa pode facilmente enviar dados para uma plataforma de análise operada por outra empresa, permitindo uma integração perfeita entre sistemas diversos.

A Internet das Coisas traz uma capacidade avançada de monitoramento em tempo real, essencial para a manutenção preventiva em ambientes industriais. Sensores conectados a equipamentos podem monitorar continuamente o desempenho e alertar sobre falhas iminentes, permitindo intervenções antes que ocorram paradas não programadas. Este tipo de monitoramento é crucial para aumentar a eficiência e minimizar o tempo de inatividade,

como evidenciado por pesquisas sobre redes de sensores em tempo real aplicadas na indústria 4.0 (KOULAMAS; LAZARESCU, 2020).

A automação baseada em dados coletados pela IoT oferece uma maneira poderosa de otimizar processos em diversos setores. Por exemplo, sistemas de climatização podem ser ajustados automaticamente com base nas condições ambientais em tempo real, enquanto rotas de entrega de veículos comerciais podem ser otimizadas dinamicamente para economizar tempo e combustível. Essa automação é fundamental para reduzir custos e melhorar a eficiência operacional, destacando o papel central da IoT na transformação digital das operações industriais (LEE, 2019).

2.1.3 Principais Usos

A IoT tem se mostrado extremamente versátil, encontrando aplicações em uma ampla gama de setores, desde saúde e agricultura até transporte, indústria e urbanismo. Seus benefícios potenciais são diversos e impactantes, transformando fundamentalmente a maneira como interage com o mundo ao nosso redor.

No setor da saúde, a IoT está revolucionando como os pacientes são monitorados e tratados. Sensores conectados podem ser implantados em dispositivos vestíveis ou mesmo em pacientes diretamente, permitindo o monitoramento remoto de sinais vitais, como frequência cardíaca, pressão arterial e níveis de glicose (SHENG; TANG; YAO, 2013). Essa capacidade de monitoramento contínuo não apenas melhora a qualidade do cuidado, mas também pode identificar precocemente problemas de saúde e facilitar intervenções médicas oportunas.

A Internet das Coisas está revolucionando a agricultura através da chamada "agricultura de precisão". Este conceito envolve o uso de sensores instalados em campos agrícolas para monitorar variáveis como umidade do solo, temperatura ambiente e níveis de nutrientes. Com esses dados, os agricultores podem otimizar o uso de recursos, como água e fertilizantes, e tomar decisões mais informadas sobre o momento ideal para plantio, colheita e irrigação. Isso não apenas aumenta a eficiência da produção agrícola, mas também reduz o desperdício e o impacto ambiental. A agricultura de precisão é um sistema de gerenciamento agrícola que utiliza tecnologias avançadas para otimizar o uso de insumos agrícolas, considerando a variabilidade espacial e temporal das áreas de cultivo (INAMASU et al., 2015; TSCHIEDEL; FERREIRA, 2000; MIRANDA; VERíSSIMO; CEOLIN, 2017).

2.1.4 Tecnologias Habilitadoras

A Internet das Coisas (IoT) é possibilitada por uma variedade de tecnologias que trabalham em conjunto para viabilizar sua implementação e funcionamento eficaz. Essas tecnologias desempenham papéis específicos e complementares, permitindo a interconexão

de dispositivos e a coleta de dados em tempo real.

Entre as principais tecnologias habilitadoras da IoT, destacam-se os protocolos de comunicação, como *MQTT (Message Queuing Telemetry Transport)* e *CoAP (Constrained Application Protocol).* Esses protocolos são projetados para facilitar a transmissão eficiente de dados entre dispositivos IoT e servidores, garantindo uma comunicação confiável e de baixa latência (ATZORI; IERA; MORABITO, 2010b).

Além dos protocolos de comunicação, as tecnologias de identificação desempenham um papel fundamental na IoT. *RFID (Identificação por Radiofrequência)* e *NFC (Near Field Communication)* são exemplos de tecnologias de identificação que permitem a rastreabilidade de objetos em tempo real. Essas tecnologias são amplamente utilizadas em cadeias de suprimentos, logística e gerenciamento de estoques, permitindo o acompanhamento preciso do fluxo de produtos e materiais ao longo de toda a cadeia de valor.

A proliferação de dispositivos com conectividade sem fio é outra tendência importante que impulsiona a expansão da IoT. Tecnologias como Bluetooth e 5G oferecem uma ampla cobertura de rede e alta velocidade de comunicação, possibilitando a conexão de dispositivos IoT em uma variedade de ambientes e cenários de uso. Essas redes sem fio fornecem a infraestrutura necessária para a transmissão de dados em tempo real e a interação entre dispositivos distribuídos geograficamente.

2.2 Banco de Dados Relacionais

Os Bancos de Dados Relacionais (BDR) representam uma mudança paradigmática na gestão de dados, cujas raízes remontam à década de 1970. Antes da sua introdução, os modelos de banco de dados predominantes eram os hierárquicos e em rede. Esses modelos, embora úteis em certos contextos, apresentavam limitações significativas em termos de flexibilidade e facilidade de uso. No modelo hierárquico, por exemplo, os dados eram organizados em uma estrutura de árvore, o que dificultava a representação de relações complexas entre entidades e muitas vezes tornava a estrutura inadequada para sistemas mais dinâmicos. Da mesma forma, o modelo em rede permitia relações mais complexas, mas sua estrutura intricada era difícil de compreender e gerenciar, limitando a eficiência na manipulação de dados (SILBERSCHATZ; KORTH; SUDARSHAN, 2011; DATE, 2004; OLIVEIRA et al., 2015).

Diante dessas limitações, Edgar F. Codd propôs um novo paradigma: o modelo relacional. Em seu influente artigo "A Relational Model of Data for Large Shared Data Banks" de 1970, Codd delineou uma abordagem na qual os dados são organizados em tabelas relacionadas, proporcionando uma visão mais lógica e consistente da estrutura dos dados. Esse modelo revolucionário permitia a representação de relações complexas de maneira simples e intuitiva, facilitando a manipulação e consulta dos dados. A introdução

da linguagem de consulta estruturada *SQL (Structured Query Language)* simplificou ainda mais a interação com os bancos de dados relacionais, consolidando a mudança proposta por Codd (CODD, 1970).

Assim, os Bancos de Dados Relacionais surgiram como uma resposta direta às limitações dos modelos anteriores, oferecendo uma solução mais flexível, intuitiva e eficiente para a gestão de dados. Desde então, eles se tornaram a base de muitos sistemas de informação, impulsionando avanços significativos em uma ampla gama de aplicações e setores industriais (GARCIA; SOTTO, 2019).

2.2.1 Origens e Motivações

Na época da introdução do modelo relacional, os modelos de banco de dados predominantes eram o hierárquico e o em rede. Embora esses modelos tenham sido eficazes em contextos específicos, apresentavam limitações significativas.

O modelo hierárquico organizava os dados em uma estrutura de árvore, o que dificultava a representação de relações complexas entre entidades. Cada entidade tinha um relacionamento pai-filho, limitando a flexibilidade para modelar interações que não se encaixavam facilmente nessa estrutura. A consulta e atualização de dados eram mais complicadas devido à rigidez da hierarquia, tornando a manipulação de dados menos eficiente (DATE, 2004).

Por outro lado, o modelo em rede oferecia uma abordagem mais flexível ao permitir relacionamentos mais complexos entre entidades. Contudo, sua estrutura complexa e interconectada tornava a compreensão e o gerenciamento mais difíceis. A necessidade de conhecer a estrutura exata do banco de dados para realizar consultas e manutenções tornava seu uso menos intuitivo (KORTH; SILBERSCHATZ, 2019).

O modelo relacional, introduzido por Edgar F. Codd na década de 1970, revolucionou o gerenciamento de dados ao organizar informações em tabelas e utilizar operações matemáticas para manipular essas tabelas. Essa abordagem simplificou a representação de relacionamentos complexos e proporcionou uma maneira mais intuitiva e flexível de gerenciar dados (CODD, 1970).

Desde então, os BDRs se tornaram a base de muitos sistemas de informação modernos, desempenhando um papel crucial em uma ampla gama de aplicações e promovendo avanços significativos em áreas como negócios, ciência, saúde e tecnologia (SILBERSCHATZ; KORTH; SUDARSHAN, 2010)

2.2.2 Principais Características

Os Bancos de Dados Relacionais (BDR) são caracterizados por uma série de elementos fundamentais que contribuem para sua eficiência e robustez. Uma das características

mais distintivas dos BDRs é a sua estrutura organizada em tabelas. Cada tabela representa uma entidade ou conceito da realidade, e as associações entre essas entidades são estabelecidas por meio de chaves primárias e estrangeiras. (DATE, 2004).

As chaves primárias são atributos únicos que identificam exclusivamente cada registro nessa tabela. Por exemplo, em uma tabela de clientes, o número de identificação do cliente pode ser definido como a chave primária, garantindo que cada cliente tenha um identificador único.

As chaves estrangeiras são atributos que estabelecem uma relação entre duas tabelas, permitindo a criação de associações entre os dados. Essas chaves são essenciais para manter a integridade referencial no banco de dados, assegurando que os relacionamentos entre tabelas sejam consistentes.

De acordo com (SILBERSCHATZ; KORTH; SUDARSHAN, 2010), as chaves primárias e estrangeiras desempenham papéis cruciais na modelagem de dados e na estruturação de bancos de dados relacionais. Além disso, (DATE, 2004) destaca a importância dessas chaves na normalização de bancos de dados, um processo que visa reduzir a redundância e melhorar a integridade dos dados.

Os BDRs são projetados para manter a integridade referencial dos dados. Isso significa que as relações entre as entidades são mantidas consistentes ao longo do tempo. Por exemplo, se um registro em uma tabela relacionada for excluído, as referências a esse registro em outras tabelas também serão atualizadas ou removidas para manter a integridade dos dados (DATE, 2004).

Para interagir e manipular os dados armazenados em um BDR, é utilizada uma linguagem de consulta estruturada, comumente conhecida como *SQL (Structured Query Language)*. O SQL fornece uma maneira padronizada de realizar operações como inserção, atualização, exclusão e consulta de dados em um BDR. Sua sintaxe intuitiva e poderosa facilita a realização de tarefas complexas de manipulação de dados (SILBERSCHATZ; KORTH; SUDARSHAN, 2010).

2.2.3 Principais Usos

Os Bancos de Dados Relacionais (BDR) têm se destacado como uma solução versátil e confiável em uma ampla gama de aplicações, desde sistemas de gerenciamento empresarial até projetos web e científicos. Sua estrutura tabular proporciona uma abordagem intuitiva para representar relações complexas entre conjuntos de dados diversos, tornando-os particularmente eficazes em cenários que exigem consistência e integridade.

A popularidade dos BDRs é evidente na variedade de setores que os adotam como solução para suas necessidades de gerenciamento de dados. Em empresas, eles servem como a espinha dorsal dos sistemas de informação, garantindo a organização e a acessibilidade

dos dados cruciais para operações cotidianas. Na esfera científica, os BDRs desempenham um papel fundamental na organização e análise de grandes volumes de dados, facilitando a condução de pesquisas e estudos complexos.

A evolução dos BDRs ao longo dos anos acompanhou de perto o avanço da tecnologia. Uma série de sistemas de gerenciamento de banco de dados (SGBDR) foram desenvolvidos, cada um com suas próprias características distintas. Exemplos notáveis incluem o MySQL, *PostgreSQL*, Oracle Database e Microsoft SQL Server. À medida que a demanda por escalabilidade e desempenho aumentou, surgiram inovações como o armazenamento em nuvem e sistemas distribuídos, oferecendo soluções otimizadas para lidar com conjuntos de dados gradativos e mais complexos (DATE, 2004).

2.2.4 ACID

O conceito ACID (Atomicidade, Consistência, Isolamento, Durabilidade) é essencial para compreender a robustez e confiabilidade dos Bancos de Dados Relacionais (BDR). Ele representa um conjunto de propriedades que garantem a integridade e consistência dos dados armazenados (DATE, 2004).

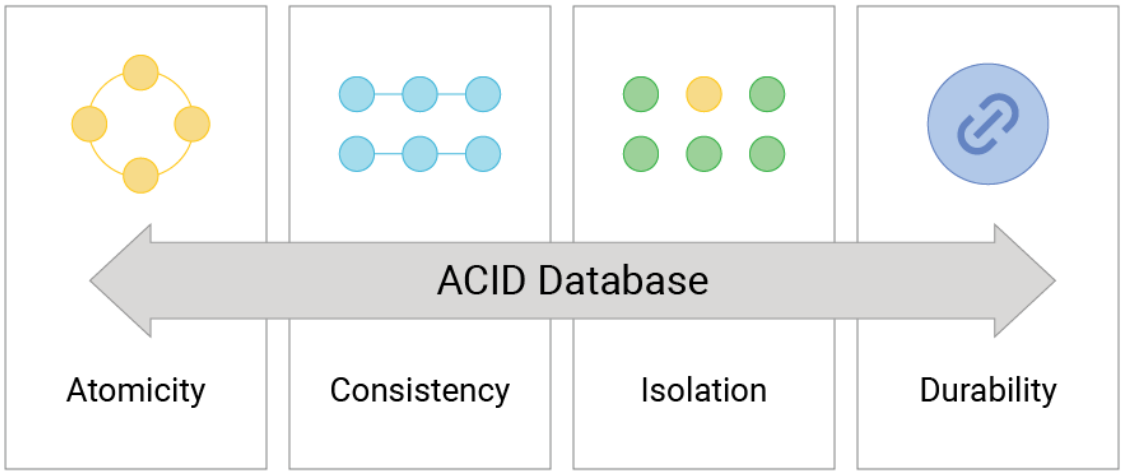

Figura 2 – ACID de bancos de dados SQL.

A Figura 2 ilustra o conceito de ACID, fundamental para a integridade das transações em bancos de dados. A partir deste ponto, serão abordados de forma detalhada os quatro princípios que constituem o conceito de ACID: Atomicidade, Consistência, Isolamento e Durabilidade, essenciais para garantir a integridade das transações em bancos de dados.

Atomicidade assegura que todas as operações de uma transação sejam concluídas com sucesso ou, em caso de falha, nenhuma delas seja executada. Isso evita estados

inconsistentes no banco de dados. Por exemplo, ao transferir fundos entre contas bancárias, é crucial que tanto o débito quanto o crédito ocorram integralmente, sem deixar a transação em um estado intermediário e arriscado.

Consistência garante que o banco de dados permaneça em um estado válido antes e após a execução de uma transação. Isso significa que todas as operações devem obedecer às restrições e regras definidas pelo esquema do banco de dados, preservando a integridade dos dados. Por exemplo, se uma transação viola uma restrição de chave estrangeira, ela não será concluída, mantendo o banco de dados em um estado consistente.

Isolamento assegura que transações concorrentes não interfiram umas nas outras. Em ambientes onde várias transações ocorrem simultaneamente, é essencial que cada transação seja executada como se fosse a única no sistema, evitando conflito e garantindo a consistência dos dados.

Durabilidade assegura que as alterações realizadas por uma transação sejam permanentes, mesmo em caso de falha do sistema. Isso significa que, uma vez que uma transação seja confirmada, suas alterações devem ser armazenadas de forma segura e persistente no banco de dados, mesmo que ocorra uma queda de energia ou falha de hardware.

2.3 Bancos de Dados Não-Relacionais

Os Bancos de Dados Não-Relacionais (BDNR), popularmente conhecidos como NoSQL, representam uma evolução marcante na área da gestão de dados, e sua ascensão foi catalisada pelas limitações percebidas nos modelos tradicionais de Bancos de Dados Relacionais (BDR). Na década de 2000 trouxe consigo uma explosão exponencial na geração de dados, alimentada pelo crescimento exponencial da internet e pela multiplicidade de fontes de informação disponíveis. Esse contexto desafiou os paradigmas estabelecidos de armazenamento e recuperação de dados, estimulando a busca por soluções mais flexíveis e escaláveis (SADALAGE; FOWLER, 2012).

Os BDNRs surgiram como uma resposta a esses desafios, oferecendo uma abordagem alternativa e inovadora para a gestão de dados em escala. Ao contrário dos BDRs, que se baseiam em um modelo tabular e em linguagens de consulta estruturada, os BDNRs adotam uma variedade de modelos de dados, como documentos, grafos e chave-valor, proporcionando uma maior flexibilidade para lidar com diferentes tipos de dados e estruturas (STRAUCH, 2013).

Um dos principais impulsionadores por trás da popularidade dos BDNRs é sua capacidade de lidar eficientemente com grandes volumes de dados não estruturados e semiestruturados, comuns em ambientes como mídias sociais, análises de Big Data e

Internet das Coisas. Esses sistemas são projetados para oferecer escalabilidade horizontal, distribuindo os dados entre vários nós de maneira eficiente e permitindo um processamento paralelo de consultas e análises (HAN; KAMBER; PEI, 2011).

Além disso, os BDNRs são frequentemente elogiados por sua tolerância a falhas e capacidade de recuperação automática. Em ambientes distribuídos e altamente dinâmicos, onde falhas de hardware e rede são inevitáveis, essa capacidade de manter a disponibilidade dos dados e a continuidade das operações é essencial (STRAUCH, 2013).

2.3.1 Origens e Motivações

A ascensão dos Bancos de Dados Não-Relacionais (BDNR) se deu em resposta a uma necessidade premente: lidar com a explosão de informações que inundou o cenário digital nas últimas décadas. Imagine o cenário: redes sociais fervilhantes como Facebook, Twitter e Instagram, onde cada curtida, compartilhamento ou tweet contribui para um volume imenso de dados não estruturados. Acrescente a isso os dispositivos inteligentes em nossas casas e escritórios, que geram dados constantemente, desde registros de temperatura até informações de saúde. É uma verdadeira avalanche de informações!

Essa enxurrada de dados rapidamente sobrecarregou os modelos tradicionais de Bancos de Dados Relacionais (BDR), que não estavam preparados para lidar com essa diversidade e magnitude de informações. Foi aí que os BDNRs entraram em cena, oferecendo uma solução mais flexível e dinâmica para esse desafio.

O *MongoDB*, por exemplo, surgiu em 2007, trazendo uma nova abordagem para o armazenamento e recuperação de dados. Em seguida, o Cassandra, desenvolvido pelo Facebook e lançado em 2008, foi uma peça fundamental nessa revolução, seguido pelo Couchbase, lançado em 2010 e que rapidamente se tornou uma ferramenta valiosa para muitas empresas.

Essa flexibilidade dos BDNRs permitiu que eles se adaptassem rapidamente às mudanças nos requisitos de dados, tornando-os ideais para ambientes onde a estrutura e o volume dos dados podem variar significativamente ao longo do tempo. Além disso, sua capacidade de distribuição e escalabilidade horizontal os torna altamente eficazes em lidar com cargas de trabalho intensivas e demandas imprevisíveis.

2.3.2 Principais Características

Os Bancos de Dados Não-Relacionais (BDNR) representam uma abordagem radicalmente diferente em comparação com os tradicionais Bancos de Dados Relacionais (BDR). Enquanto os BDRs seguem um modelo tabular rígido, onde os dados são organizados em tabelas com esquemas predefinidos, os BDNRs adotam uma abordagem mais flexível e dinâmica, incorporando o conceito de schema dinâmico (SADALAGE; FOWLER, 2012).

Essa característica fundamental dos BDNRs enorme flexibilidade, especialmente em ambientes onde a estrutura e o formato dos dados podem variar significativamente ao longo do tempo. Por exemplo, em um ambiente de desenvolvimento ágil de software, onde novos requisitos e funcionalidades são frequentemente adicionados e modificados, essa capacidade de adaptabilidade dos BDNRs é inestimável.

Além disso, os BDNRs são projetados para serem altamente escaláveis e distribuídos. Eles conseguem lidar com grandes volumes de dados e suportar um número crescente de usuários e solicitações, distribuindo os dados entre vários servidores eficientemente. Isso os torna ideais para ambientes onde a escalabilidade é uma preocupação primordial, como aplicações web de alta demanda e análises de Big Data (HAN; KAMBER; PEI, 2011).

Esses bancos de dados NoSQL oferecem alternativas altamente flexíveis e escaláveis, capazes de atender a uma ampla variedade de necessidades de aplicativos, superando as limitações dos bancos de dados relacionais tradicionais em termos de desempenho, escalabilidade e estrutura de dados.

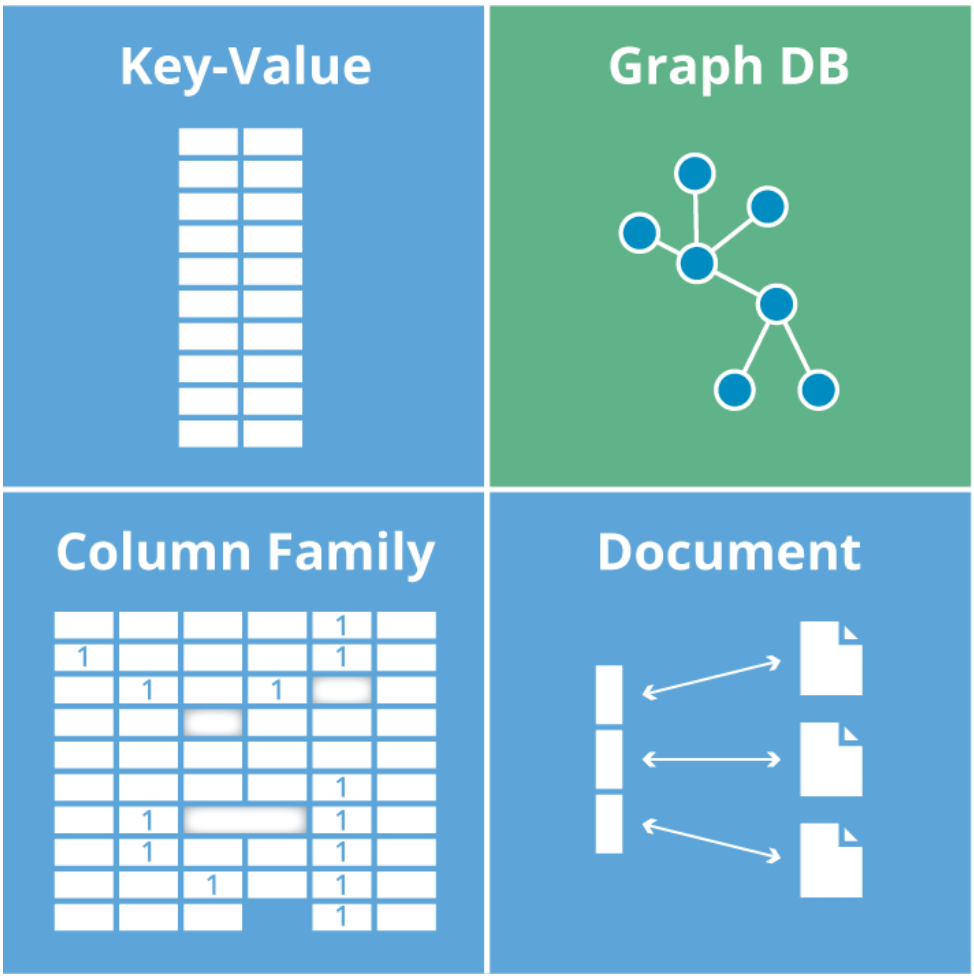

Figura 3 – Tipos de Modelos de bancos não relacionais

Fonte: https://www.rkimball.com/

A Figura 3 explica os diferentes modelos de banco de dados não relacionais e seus usos. A seguir, serão detalhados os principais tipos de bancos de dados NoSQL, destacando suas características, vantagens e cenários de aplicação.

- **Key-Value (Chave-Valor)**: Nesse modelo, os dados são armazenados como pares de chave e valor. Cada chave é única e identifica um valor associado. Esse tipo é especialmente útil para armazenar dados não estruturados ou para fins de cache.
- **Graph DB (Banco de Dados de Grafos)**: Neste modelo, os dados são representados como nós interconectados por relacionamentos. É ideal para modelar relações complexas, como redes sociais ou sistemas de recomendação.
- **Column Family (Família de Colunas)**: Organiza os dados em colunas agrupadas em famílias. Esse modelo é eficiente para consultas e armazenamento em grande escala.
- **Document (Documento)**: Armazena dados em formato de documentos, como JSON ou XML. Permite estruturas aninhadas e oferece boa escalabilidade.

Um exemplo marcante desse novo paradigma é o *MongoDB*, um sistema de gerenciamento de banco de dados que adota um modelo de documentos flexível. Neste modelo, os dados são organizados em documentos no formato JSON, o que permite uma representação mais próxima à estrutura natural das informações, sem a rigidez imposta pelas tabelas tradicionais dos bancos relacionais (SADALAGE; FOWLER, 2013).

O *MongoDB* oferece duas abordagens principais para modelar dados: a incorporação e as referências. A incorporação armazena dados relacionados dentro de um único documento, o que pode otimizar o desempenho de leitura e tornar as operações de atualização mais simples (CHODOROW, 2013). Por outro lado, as referências possibilitam armazenar os relacionamentos entre os dados por meio de links entre documentos, evitando redundâncias e representando de forma eficiente relacionamentos complexos (STRAUCH, 2017).

Outro exemplo relevante é o Apache Cassandra, que foi projetado para lidar com grandes volumes de dados distribuídos por diversos servidores, assegurando alta disponibilidade e tolerância a falhas. Utilizando um modelo de chave-valor, o Cassandra se destaca por sua escalabilidade e resiliência a falhas de hardware e rede (LAKSHMAN; MALIK, 2010). Sua arquitetura descentralizada, em que todos os nós do cluster possuem funções iguais, elimina pontos únicos de falha e possibilita a expansão do sistema sem interrupções no serviço (GEORGE, 2011).

2.3.3 Tecnologias e Evolução

Nos últimos anos, os bancos de dados NoSQL emergiram como uma alternativa robusta aos sistemas tradicionais de gerenciamento de bancos de dados relacionais (RDBMS). Essa evolução é impulsionada pela necessidade crescente de escalabilidade, flexibilidade e desempenho, que os modelos relacionais, com suas limitações, não conseguem atender adequadamente (WHITE, 2017).

Os bancos de dados NoSQL, que significa *"Not Only SQL",* foram projetados para superar as restrições dos bancos de dados relacionais, especialmente em termos de escalabilidade horizontal e flexibilidade de esquema. Diferentemente dos bancos de dados relacionais, que utilizam tabelas e esquemas fixos, os sistemas NoSQL oferecem uma variedade de modelos de dados, incluindo documentos, chave-valor, coluna e grafos. Essa diversidade permite uma adaptação mais eficaz a diferentes tipos de aplicações e requisitos (STRAUCH, 2013).

O universo NoSQL engloba uma gama diversificada de tecnologias, cada uma projetada para atender a necessidades específicas. Entre as mais reconhecidas e utilizadas estão o *MongoDB*, Cassandra, Couchbase e Redis. Cada uma dessas plataformas NoSQL possui características distintas que as tornam adequadas para diferentes cenários de uso, destacando-se em aspectos como armazenamento de documentos, chave-valor, famílias de colunas e armazenamento em memória (STRAUCH, 2013).

O universo NoSQL engloba uma gama diversificada de tecnologias, cada uma projetada para atender a necessidades específicas. Entre as mais reconhecidas e amplamente utilizadas estão *MongoDB*, Cassandra, Couchbase e Redis. Cada uma dessas plataformas NoSQL possui características distintas que as tornam adequadas para diferentes cenários de uso. A seguir, serão apresentados alguns exemplos de bancos de dados NoSQL, com uma breve descrição de suas principais características e aplicações:

- **MongoDB:** É amplamente elogiado por sua flexibilidade e escalabilidade. Baseado em documentos, o *MongoDB* permite que os desenvolvedores armazenem e manipulem dados sem um esquema rígido, facilitando o desenvolvimento de aplicações que requerem uma estrutura de dados adaptável e evolutiva (MongoDB, Inc., 2021).

- **Cassandra:** Desenvolvido inicialmente pelo Facebook e posteriormente tornado open source, o Cassandra é valorizado por sua capacidade de lidar com grandes volumes de dados distribuídos em vários servidores. Sua arquitetura distribuída e descentralizada oferece alta disponibilidade e resiliência a falhas, tornando-o ideal para aplicações que necessitam de alta disponibilidade e escalabilidade (Apache Cassandra, 2020).

- **Couchbase:** Oferece uma solução abrangente para armazenamento e recuperação

de dados em escala, combinando a flexibilidade do modelo de documentos com avançados recursos de consulta e indexação. É particularmente eficaz em ambientes que demandam alta carga de trabalho e baixa latência (Couchbase, Inc., 2021).

- **Redis:** Conhecido por sua velocidade e eficiência no armazenamento de dados em memória, o Redis é frequentemente utilizado em cenários que exigem alto desempenho, como cache de dados em tempo real, gerenciamento de sessões de usuário e filas de mensagens (Redis Labs, 2022).

Essas tecnologias NoSQL representam uma evolução significativa no campo do armazenamento de dados, oferecendo uma gama diversificada de opções para atender às necessidades variadas das aplicações modernas. Seu surgimento e adoção generalizada refletem a busca contínua por soluções mais eficientes e adaptáveis em um mundo cada vez mais orientado a dados.

3 REVISÃO SISTEMÁTICA DA LITERATURA

A revisão sistemática da literatura (SLR) é uma metodologia rigorosa e estruturada para coletar, avaliar e sintetizar pesquisas relevantes sobre um tema específico. Para realizar a SLR sobre o gerenciamento de dados em dispositivos IoT utilizando bancos de dados relacionais e não relacionais, delimita-se um protocolo de pesquisa claro, formulando perguntas específicas e estabelecendo critérios de inclusão e exclusão.

3.1 Método de Pesquisa

Utilizando bases de dados acadêmicas confiáveis, efetua-se buscas sistemáticas com palavras-chave relevantes. Os estudos encontrados foram registrados, avaliados quanto à qualidade e pontuados com base em critérios pré-definidos.

Após a seleção dos estudos, extraíram-se e sintetizaram-se os dados relevantes, comparando os resultados para identificar vantagens e desvantagens dos bancos de dados relacionais e não relacionais no contexto de IoT. A análise final incluiu uma discussão sobre os achados, destacando implicações práticas e teóricas, além de sugestões para pesquisas futuras. Todo o processo foi documentado detalhadamente, garantindo transparência e rigor na condução da revisão. A pesquisa visa responder questões cruciais sobre as melhores opções para o armazenamento de dados gerados por dispositivos IoT e como atender às suas exigências específicas. O foco está na comparação entre sistemas de bancos de dados relacionais e não relacionais, visando entender os prós e contras de cada abordagem, especialmente em relação à otimização da eficiência do armazenamento em aplicações IoT.

O objetivo central deste estudo é investigar como distintos tipos de bancos de dados gerenciam o armazenamento de dados originados de dispositivos IoT. Para isso, planeja-se entender melhor o desempenho de bancos de dados relacionais e não relacionais, identificando qual abordagem se mostra mais eficaz e viável no contexto específico da IoT. As etapas incluem realizar uma revisão sistemática da literatura, com base em artigos científicos relevantes, além da condução de testes práticos para reunir evidências sobre o desempenho dos dois tipos de banco de dados. Planeja-se modelar o banco de dados para cada abordagem de armazenamento e executar testes de benchmarking para comparar o desempenho efetivo dos bancos de dados relacionais e não relacionais.

3.2 Questões de Pesquisa.

Essas questões são formuladas para serem abordadas através do método de Revisão Sistemática da Literatura. O SLR é uma abordagem rigorosa e estruturada que tem em vista identificar, avaliar e sintetizar a literatura existente sobre um determinado tema, garantindo uma visão abrangente e imparcial do estado da arte. No contexto desta pesquisa, as perguntas fornecem a base para a revisão sistemática das evidências disponíveis sobre os bancos de dados relacionais e não relacionais no gerenciamento de dados de IoT, permitindo uma análise detalhada e fundamentada das práticas, desafios e oportunidades associados a cada tipo de banco de dados. A Tabela 2 apresenta um conjunto de questões de pesquisa elaboradas para comparar bancos de dados relacionais e não relacionais no contexto do armazenamento e processamento de dados gerados por dispositivos IoT. Cada pergunta aborda um aspecto específico dessa comparação.

Tabela 1 – Tabela de Questões de Pesquisa

Número	**Pergunta**
Q1	Como os requisitos de consumo de recursos, como uso de CPU, memória e armazenamento, variam entre bancos de dados relacionais e não relacionais para armazenar e processar dados de dispositivos IoT?
Q2	Qual é a confiabilidade percebida dos dados armazenados em bancos de dados relacionais em comparação aos bancos de dados não relacionais em ambientes de dispositivos IoT, considerando possíveis falhas e redundância?
Q3	Como as considerações de segurança e integridade dos dados diferem entre bancos de dados relacionais e não relacionais na gestão de dados sensíveis provenientes de dispositivos IoT?
Q4	Quais são os desafios e oportunidades específicos enfrentados ao usar bancos de dados relacionais e não relacionais para armazenar e processar dados de dispositivos IoT em tempo real?
Q5	Qual é o impacto da estrutura e modelagem de dados nos diferentes bancos de dados (relacionais e não relacionais) na eficiência e eficácia das análises de dados gerados por dispositivos IoT?

3.3 Processo de seleção de artigos

O processo de seleção de artigos consiste em três etapas principais:

- **Definição da de Busca e Bases de Dados**
  Nesta fase, são estabelecidas as palavras-chave e selecionadas as bases de dados onde será realizada a busca pelos artigos relevantes.

- **Definição de Critérios de Busca para Seleção dos Artigos**
  Aqui, são definidos os critérios específicos para filtrar os resultados, como período de publicação e idiomas, garantindo que apenas artigos pertinentes sejam considerados.

- **Seleção dos Artigos**
  Esta etapa envolve a análise dos artigos encontrados para verificar se atendem aos critérios de inclusão e exclusão estabelecidos.

Cada uma dessas etapas será detalhada nas sub-seções seguintes.

3.3.1 Etapa 1: Definição da de Busca e Bases de Dados

Na primeira fase da pesquisa, efetua-se uma busca automatizada utilizando uma *string* de pesquisa bem elaborada, visando maximizar a relevância dos resultados encontrados. A *string* utilizada foi a seguinte:

```
("Connected devices" OR "Dispositivos IoT" OR
 "Internet das Coisas" OR "IoT devices")
AND
("BDR" OR "Bancos de dados relacionais" OR
 "Relational databases" OR "SQL" OR "Traditional databases")
AND
("BDNR" OR "Bancos de dados não relacionais" OR
 "NoSQL" OR "NoSQL databases")
AND
("Data storage efficiency" OR "Storage performance" OR
 "Desempenho de armazenamento de dados")
```

Essa combinação de termos-chave foi cuidadosamente selecionada para abranger uma ampla gama de estudos e artigos que discutem a relação entre dispositivos IoT e os diferentes tipos de bancos de dados, com um foco especial na eficiência do armazenamento de dados. As fontes consultadas para a coleta de artigos incluíram:

- ACM Digital Library (<http://portal.acm.org>)
- IEEE Digital Library (<http://ieeexplore.ieee.org>)
- Scopus (<http://www.scopus.com>)
- Google Scholar.(<https://scholar.google.com/>)

Essas plataformas foram escolhidas pela sua credibilidade e pela qualidade dos artigos disponíveis, garantindo que nossa revisão sistemática da literatura se baseasse em informações confiáveis e atualizadas, fundamentais para o desenvolvimento do nosso trabalho.

A Tabela 4 apresenta os critérios de inclusão, que definem as características que os artigos devem possuir para serem considerados na revisão. Estes critérios garantem que apenas estudos que efetivamente comparam o desempenho de armazenamento de dados em dispositivos IoT utilizando bancos de dados relacionais e não relacionais sejam incluídos. Por outro lado, a Tabela 3 lista os critérios de exclusão, que especificam as condições sob

Tabela 2 – Tabela de Questões de Pesquisa

Número	**Pergunta**
Q1	Como os requisitos de consumo de recursos, como uso de CPU, memória e armazenamento, variam entre bancos de dados relacionais e não relacionais para armazenar e processar dados de dispositivos IoT?
Q2	Qual é a confiabilidade percebida dos dados armazenados em bancos de dados relacionais em comparação aos bancos de dados não relacionais em ambientes de dispositivos IoT, considerando possíveis falhas e redundância?
Q3	Como as considerações de segurança e integridade dos dados diferem entre bancos de dados relacionais e não relacionais na gestão de dados sensíveis provenientes de dispositivos IoT?
Q4	Quais são os desafios e oportunidades específicos enfrentados ao usar bancos de dados relacionais e não relacionais para armazenar e processar dados de dispositivos IoT em tempo real?
Q5	Qual é o impacto da estrutura e modelagem de dados nos diferentes bancos de dados (relacionais e não relacionais) na eficiência e eficácia das análises de dados gerados por dispositivos IoT?

as quais os artigos serão descartados da revisão. Estes critérios ajudam a filtrar estudos irrelevantes, como aqueles duplicados, publicados antes de 2019, ou escritos em idiomas diferentes de português e inglês. Juntas, essas tabelas ajudam a garantir a relevância e a qualidade dos estudos incluídos na revisão, promovendo uma análise mais robusta e focada.

A Tabela 4 apresenta os critérios de inclusão, que definem as características que os artigos devem possuir para serem considerados na revisão. Estes critérios garantem que apenas estudos que efetivamente comparam o desempenho de armazenamento de dados em dispositivos IoT utilizando bancos de dados relacionais e não relacionais sejam incluídos.

3.3.2 Etapa 2: Definição de Critérios de Busca para Seleção dos Artigos

A segunda fase da pesquisa, efetua-se a seleção dos artigos com base em critérios específicos, incluindo o título, o resumo e a qualidade do periódico onde os trabalhos

Tabela 3 – Critérios de Exclusão para a revisão de artigos sobre desempenho de armazenamento em IoT

Número	**Critério**
E1	Estudos duplicados.
E2	Estudos publicados antes de 2019.
E3	Estudos em idiomas diferentes de português e inglês.
E4	Estudos secundários e terciários.
E5	Trabalhos que estão fora do escopo definido, ou seja, que não realizam a comparação entre os desempenhos de armazenamento de dados em dispositivos IoT usando bancos de dados relacionais e não relacionais.
E6	Literatura cinza, como teses, manuais, dissertações e revistas.
E7	Short paper com menos de quatro páginas.

Tabela 4 – Critérios de Inclusão para a revisão de artigos sobre desempenho de armazenamento em IoT

Número	**Critério**
I1	Apenas estudos que tratam do comparativo de desempenhos de armazenamento de dados de dispositivos IoT utilizando bancos de dados relacionais e não relacionais foram considerados.
I2	O artigo deve abordar a implementação prática e/ou testes empíricos de bancos de dados relacionais e não relacionais em um ambiente de IoT.
I3	O artigo deve comparar o desempenho de armazenamento de dados de dispositivos IoT usando ambos os tipos de bancos de dados.
I4	O estudo deve incluir experimentos ou análises empíricas que avaliem o desempenho de diferentes bancos de dados no contexto de IoT.

foram publicados. Primeiramente, Analisa os títulos dos artigos encontrados na busca automatizada para identificar aqueles que eram mais relevantes para a nossa temática, que envolve a comparação entre bancos de dados relacionais e não relacionais no contexto de dispositivos IoT.

Como resultado desse processo de seleção, encontra um total de 26 artigos relevantes nas diferentes plataformas consultadas:

- ACM Digital Library: 14 artigos
- Google Scholar: 1 artigo
- Scopus: 11 artigos

A Figura 4 apresenta os resultados da pesquisa realizada utilizando uma *string* de busca específica. Após a execução da pesquisa, foram encontrados um total de 26 artigos

distribuídos entre três bases de dados acadêmicas. Especificamente, a pesquisa resultou em 14 artigos provenientes da ACM, 11 artigos da Scopus e 1 artigo do Google Scholar.A distribuição dos artigos encontrados é evidenciada pela contribuição de cada base de dados para o conjunto total: 53,8% da ACM Digital Library, 42,3% do Scopus e 3,8% do Google Scholar.

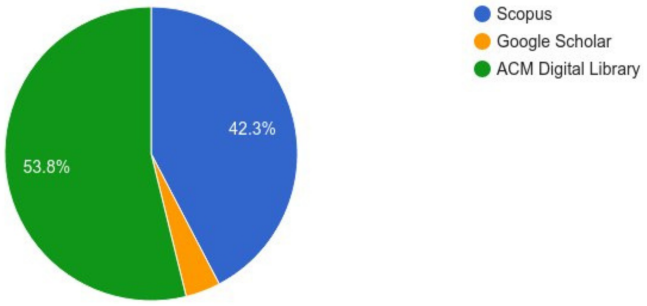

Figura 4 – Artigos por fonte

3.3.3 Etapa 3: Seleção dos Artigos

Nesta etapa, foi conduzida uma avaliação dos artigos selecionados para garantir que eles abordassem realmente a eficiência de armazenamento de dados e o desempenho dos diferentes tipos de bancos de dados em ambientes IoT. Essa análise seguiu a definição prévia da *string* de busca e das bases de dados, bem como a definição dos critérios de busca para a seleção dos artigos. Os critérios de inclusão e exclusão estabelecidos foram aplicados para refinar o conjunto de estudos, assegurando que apenas os artigos pertinentes fossem considerados.

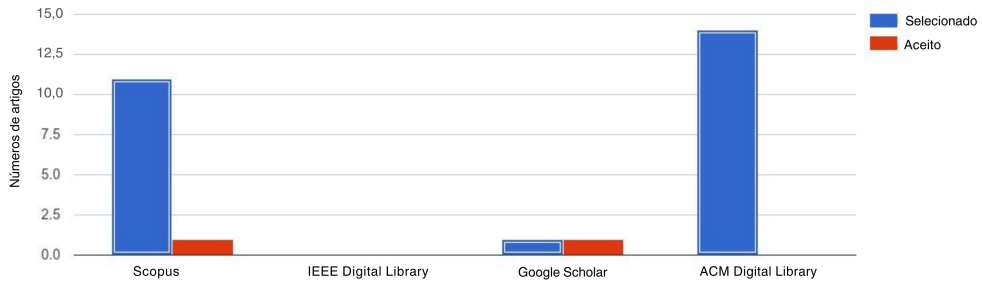

Figura 5 – Artigos Aceitos por Fonte

A Figura 5 apresenta os resultados do processo de seleção dos artigos, que foi conduzido com base em rigorosos critérios de inclusão e exclusão, conforme os procedimentos estabelecidos em uma Revisão Sistemática da Literatura (SLR). Inicialmente, foram identificados 26 artigos, com 14 provenientes da ACM, 11 da Scopus e 1 do Google Scholar. Após a aplicação dos critérios de seleção, restaram apenas dois artigos que se alinhavam

com os objetivos da pesquisa: um da Scopus e um do Google Scholar. Esta triagem criteriosa, fundamentada na Revisão Sistemática da Literatura, foi crucial para assegurar que a pesquisa se sustentasse em fontes de alta qualidade e relevância. A SLR permitiu uma avaliação detalhada e objetiva dos artigos, garantindo que estes atendem rigorosamente aos objetivos da pesquisa e estão conforme as melhores práticas no campo da Tecnologia da Informação.

Com a conclusão desta etapa de seleção, a pesquisa está preparada para avançar para a fase seguinte. Nesta fase, será realizada uma análise aprofundada dos estudos selecionados, examinando suas contribuições específicas para o tema abordado. Este exame detalhado é essencial para garantir que a revisão sistemática da literatura seja completa e bem fundamentada, oferecendo uma visão abrangente e substancial para a área de estudo.

3.4 Análise dos Artigos Selecionados

O estudo de (AZAD et al., 2020), *The Role of Structured and Unstructured Data Managing Mechanisms in the Internet of Things*, discute o papel dos mecanismos de gerenciamento de dados estruturados e não estruturados na Internet das Coisas. Utiliza uma abordagem metodológica abrangente, que combina revisão de literatura, estudos de caso e experimentação prática para analisar o gerenciamento de dados na IoT. A revisão de literatura ajuda a estabelecer uma base teórica sobre como os dados estruturados (organizados em tabelas e bancos de dados relacionais) e não estruturados (como textos, imagens e vídeos) são tratados. Os estudos de caso fornecem exemplos reais de como diferentes organizações implementam soluções para gerenciar esses dados. Além disso, experimentos práticos e simulações são utilizados para testar a eficácia de várias técnicas e ferramentas de gerenciamento de dados, proporcionando percepções sobre o desempenho das abordagens adotadas.

Entre as informações analisadas, o estudo ressalta que a principal dificuldade no gerenciamento de dados na IoT está na integração e análise eficiente dos enormes volumes e diferentes tipos de dados gerados. Dados estruturados são relativamente mais fáceis de gerenciar devido à sua organização pré-definida, enquanto dados não estruturados apresentam desafios significativos em termos de armazenamento, processamento e análise. O estudo revela que tecnologias emergentes, como processamento de dados em tempo real e análise de big data, são essenciais para lidar com a complexidade e o volume dos dados gerados pelos dispositivos IoT. Além disso, destaca a importância de implementar mecanismos robustos para a coleta e análise desses dados para garantir decisões informadas e precisas.

Os autores concluem que, para otimizar o gerenciamento de dados na IoT, é crucial adotar uma combinação de ferramentas e técnicas que abordem tanto dados estruturados

quanto não estruturados. Eles recomendam o uso de plataformas de análise avançada, algoritmos de aprendizado de máquina e técnicas de integração de dados para melhorar a eficiência e a precisão das análises. O estudo também aponta que a implementação de soluções adaptativas e escaláveis é fundamental para enfrentar os desafios dinâmicos associados à crescente quantidade e diversidade de dados na IoT. As sugestões dos autores visam melhorar a capacidade das organizações em processar e utilizar os dados de maneira mais eficaz, promovendo um avanço significativo na forma como a IoT é aplicada e gerida.

O segundo estudo de (YADAV, 2024), *Structuring SQL/ NoSQL Databases for IoT Data*, aborda as complexidades e oportunidades envolvidas no armazenamento e gerenciamento dos dados gerados pela Internet das Coisas (IoT). À medida que o volume, a velocidade e a variedade dos dados provenientes de dispositivos IoT aumentam rapidamente, é imperativo que as estruturas de banco de dados sejam altamente eficientes e adaptáveis para atender a essas crescentes demandas.

O estudo analisa de forma detalhada os aspectos de planejamento na criação de bancos de dados voltados para a Internet das Coisas (IoT), analisando as principais diferenças entre os modelos SQL e NoSQL. Os bancos de dados SQL são particularmente eficazes para dados transacionais, onde a consistência e a integridade são cruciais. No entanto, eles podem enfrentar limitações significativas em termos de escalabilidade e flexibilidade quando confrontados com a natureza heterogênea e em rápida evolução dos dados IoT. Por outro lado, os bancos de dados NoSQL oferecem uma flexibilidade e escalabilidade superiores, adequando-se bem ao gerenciamento de dados não estruturados e semi-estruturados. No entanto, a manutenção da consistência e da confiabilidade em ambientes complexos continua sendo um desafio para essa tecnologia.

Uma solução promissora que o estudo destaca é a adoção de uma estratégia de persistência poliglota, que combina os bancos de dados SQL e NoSQL. Essa abordagem permite que as organizações aproveitem as vantagens de ambas as tecnologias, maximizando a eficiência no armazenamento e na análise dos dados. A persistência poliglota tem-se mostrado eficaz especialmente em cenários que requerem tanto o gerenciamento de dados transacionais quanto a análise de grandes volumes de dados não estruturados, como leituras de sensores e telemetria.

No entanto, a implementação de uma arquitetura de persistência poliglota exige uma consideração cuidadosa de vários fatores, incluindo a modelagem de dados, o design de esquema, a otimização de consultas e as estratégias de migração de dados. As organizações devem avaliar suas necessidades específicas para projetar uma solução que equilibre escalabilidade, flexibilidade e desempenho. Estratégias híbridas de integração de dados são essenciais para garantir que diferentes tipos de dados possam ser armazenados e gerenciados coesamente, suportando as diversas cargas de trabalho exigidas pelas aplicações IoT.

O estudo também enfatiza a importância de avaliações de desempenho e bench-

marking para determinar a eficiência, escalabilidade e confiabilidade dos bancos de dados SQL e NoSQL. Tais avaliações auxiliam as organizações a escolher a tecnologia de banco de dados mais adequada para suas necessidades específicas. Comparações experimentais revelam que, embora cada tecnologia tenha suas vantagens, a escolha entre SQL e NoSQL deve ser guiada pelo tipo de dados e pelas exigências específicas da aplicação.

A comparação entre os dois artigos selecionados e a pesquisa desenvolvida neste trabalho oferece uma visão abrangente sobre o gerenciamento de dados em ambientes IoT, destacando tanto as soluções tradicionais quanto as emergentes. O artigo *"The role of structured and unstructured data managing mechanisms in the Internet of Things"* explora as características e desafios dos bancos de dados SQL, NoSQL e gráficos, abordando suas capacidades de gerenciar grandes volumes de dados estruturados e não estruturados, além de destacar a importância da consistência e escalabilidade. Por outro lado, *"Structuring SQL/NoSQL databases for IoT data*"aprofunda-se na estruturação e nas estratégias de persistência poliglota, sugerindo a combinação de bancos de dados relacionais e não-relacionais para maximizar a eficiência e a flexibilidade em cenários IoT. A pesquisa está focada em comparar com precisão o desempenho dos sistemas de armazenamento de dados no contexto da Internet das Coisas (IoT).O objetivo deste estudo é analisar e comparar aspectos essenciais relacionados ao funcionamento de sistemas, como a eficiência da compilação, o desempenho sob carga de trabalho, o uso de recursos (como memória, CPU e armazenamento em disco), o tempo de execução dos processos e a capacidade do sistema em gerenciar múltiplos aplicativos ao mesmo tempo. Para alcançar esses objetivos, a pesquisa envolve uma série de etapas práticas e metodológicas: primeiramente, modela-se o banco de dados para cada abordagem de armazenamento, seja ela relacional ou não relacional. Em seguida, dados são gerados por meio da coleta sistemática usando sensores ou simuladores. Finalmente, são executados testes de benchmarking utilizando softwares especializados para comparar o desempenho de ambos os tipos de banco de dados. Com isso, a pesquisa visa contribuir significativamente para a área, orientando a seleção de tecnologias de banco de dados que melhor atendam às necessidades específicas dos sistemas IoT, equilibrando desempenho e custo-benefício.

4 MODELAGENS DOS BANCOS DE DADOS PARA DISPOSITIVOS IOT

Este capítulo apresenta as modelagens de banco de dados para sistemas IoT, abordando tanto soluções relacionais quanto não relacionais. A modelagem relacional será exemplificada com o uso do *PostgreSQL*, enquanto a modelagem NoSQL será ilustrada com o *MongoDB*. Ambas as abordagens são discutidas com o intuito de demonstrar como diferentes estruturas de dados podem ser aplicadas ao armazenamento e gerenciamento de informações em sistemas de dispositivos IoT.

4.1 Descrição da Modelagem Relacional

A modelagem apresentada utiliza o banco de dados *PostgreSQL*, uma solução relacional robusta que oferece suporte a tipos de dados avançados e extensibilidade, características fundamentais para aplicações IoT. A Figura 6 ilustra o modelo de dados desenvolvido.

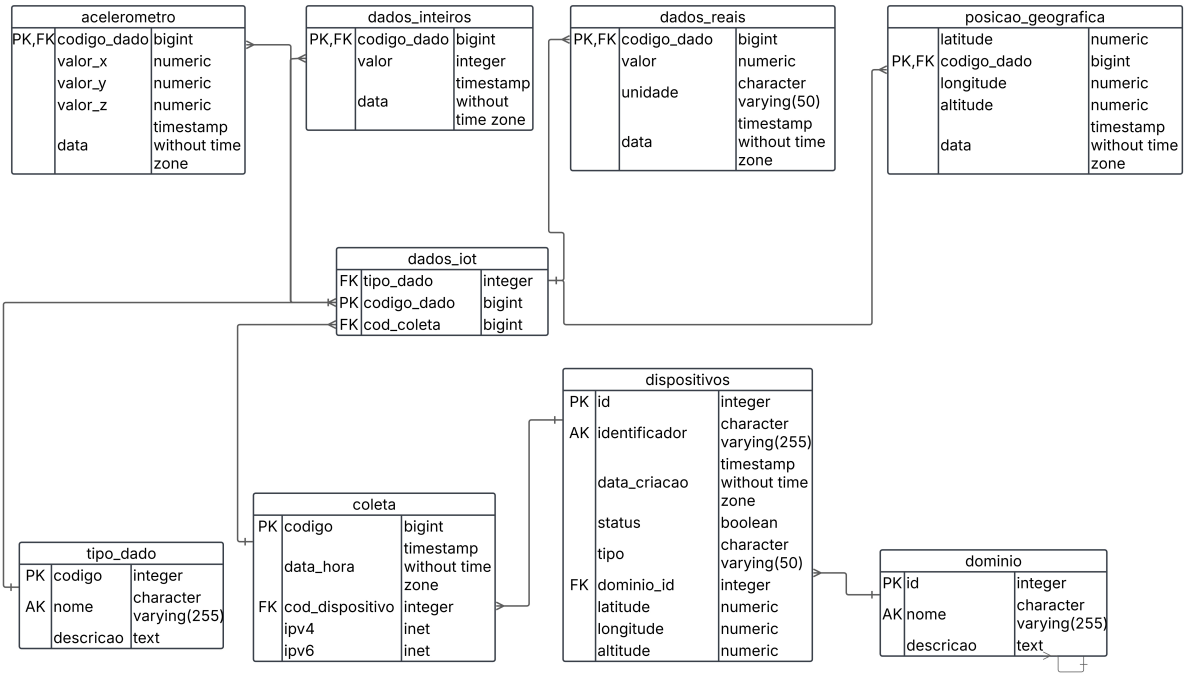

Figura 6 – Modelo de banco de dados para dispositivos IoT.

O modelo é composto pelas seguintes tabelas principais:

4.1.1 Tabela *Dominio*

A tabela *Dominio* organiza dispositivos em diferentes categorias ou contextos, facilitando a gestão. Um domínio pode representar, por exemplo, uma empresa, organização ou segmento, como "Residencial" ou "Industrial". Seus principais atributos incluem:

- `id`: Chave primária, gerada automaticamente.
- `nome`: nome do domínio, como "Residencial" ou "Industrial".
- `descricao`: descrição detalhada do domínio.

Essa tabela permite agrupar dispositivos por contexto, facilitando consultas e análises.

4.1.2 Tabela *Dispositivos*

A tabela *Dispositivos* registra informações sobre cada dispositivo conectado. Seus atributos incluem:

- `id`: identificador único do dispositivo, chave primária.
- `identificador`: um identificador externo, como o número de série.
- `data_criacao`: data e hora de registro no sistema.
- `status`: indica se o dispositivo está ativo ou inativo.
- `tipo`: define o tipo de dispositivo.
- `dominio_id`: relacionamento com a tabela *Dominio*.
- `latitude`, `longitude`, `altitude`: coordenadas geográficas da localização física do dispositivo.

Essa tabela centraliza a identificação e o contexto de cada dispositivo.

4.1.3 Tabela *Coleta*

A tabela *Coleta* armazena eventos de coleta de dados. Os Atributos dessa tabela são:

- `codigo`: Chave primária que identifica cada coleta.
- `data_hora`: data e hora da coleta.
- `cod_dispositivo`: relacionamento com a tabela *Dispositivos*.

- `ipv4`, `ipv6`: endereços IP utilizados pelo dispositivo na coleta.

Essa tabela vincula os dados coletados ao dispositivo e ao momento específico.

4.1.4 Tabela *Dados_IoT*

A tabela *Dados_IoT* permite o armazenamento de diferentes tipos de dados, integrando coletas com tabelas específicas. AOs Atributos dessa tabela são:

- `codigo_dado`: identificador único para os dados.
- `tipo_dado`: relacionamento com a tabela *Tipo_Dado*, indicando o tipo de dado armazenado.
- `cod_coleta`: relacionamento com a tabela *Coleta*, associando os dados a um evento de coleta.

Essa tabela facilita a flexibilidade no armazenamento de diferentes formatos de dados.

4.1.5 Tabela *Tipo_Dado*

A tabela *Tipo_Dado* especifica os tipos de dados armazenados, como aceleração ou temperatura. Atributos:

- `codigo`: identificador único do tipo de dado.
- `nome`: nome do tipo de dado, como "Acelerômetro".
- `descricao`: descrição detalhada do tipo de dado.

Essa tabela fornece flexibilidade e consistência na definição de tipos de dados.

4.1.6 Tabelas Específicas de Dados

Tabelas especializadas armazenam informações detalhadas sobre cada tipo de dado:

- ***Acelerometro***: Registra valores `X`, `Y`, `Z` e a data.
- ***Posicao_Geografica***: armazena latitude, longitude, altitude e data.
- ***Dados_Inteiros*** e ***Dados_Reais***: guardam valores numéricos e suas unidades.

Essas tabelas permitem maior especialização e eficiência no armazenamento de dados heterogêneos.

4.2 Modelagem em Banco de Dados NoSQL (MongoDB)

Os bancos de dados NoSQL, como o MongoDB, diferem significativamente dos bancos relacionais por serem projetados para lidar com grandes volumes de dados não estruturados ou semiestruturados. Em vez de tabelas e linhas, o MongoDB utiliza coleções e documentos no formato JSON (ou BSON), o que proporciona maior flexibilidade na modelagem de dados. Essa característica torna o MongoDB ideal para aplicações IoT, onde os dispositivos podem gerar dados heterogêneos em formatos imprevisíveis.

4.2.1 Estrutura e Criação das Coleções

No contexto do MongoDB, cada coleção representa uma entidade do sistema IoT, com os seguintes propósitos:

- `Dominio`: Armazena informações sobre diferentes categorias ou contextos de dispositivos, como "Residencial" ou "Industrial". Isso permite organizar dispositivos logicamente, facilitando análises e consultas contextuais.
- `Dispositivos`: Contém detalhes de cada dispositivo, incluindo identificadores, tipo, localização geográfica e status. Essa coleção centraliza a identificação e monitoramento de dispositivos em diferentes cenários IoT.
- `Coleta`: Representa eventos de coleta de dados realizados pelos dispositivos. Inclui informações como timestamp e endereços IP, vinculando os dados coletados aos dispositivos específicos.
- `Tipo_Dado`: Define os tipos de dados armazenados, como "Acelerômetro" ou "Temperatura". Isso permite flexibilidade na adição de novos tipos de dados sem alterar a estrutura do banco.
- `Dados_IoT`: Relaciona os dados coletados ao tipo e à coleta correspondente. Essa coleção abstrai a integração de dados específicos com suas coletas e dispositivos.

A criação das coleções no MongoDB é realizada através do comando `db.createCollection()`, que permite a definição de novas coleções no banco de dados. No **Código 1**, apresentamos um exemplo de como criar as coleções necessárias para o armazenamento dos dados dos sensores IoT. Esse código deve ser executado no shell do MongoDB ou em um script de inicialização do banco de dados. A seguir, você pode ver o código que cria as coleções `"Dominio"`, `"Dispositivos"`, `"Coleta"`, `"Tipo_Dado"` e `"Dados_IoT"`, que são essenciais para o processo de coleta e armazenamento de dados dos dispositivos IoT.

Código 1 – Criação das coleções no MongoDB

```
// Conectar ao banco de dados IoTDatabase
use IoTDatabase;

// Criar as coleções para armazenar os dados dos sensores IoT
db.createCollection("Dominio"); // Coleção para armazenar domínios de dispositivos
db.createCollection("Dispositivos"); // Coleção para armazenar informações dos
    dispositivos IoT
db.createCollection("Coleta"); // Coleção para armazenar as coletas de dados
    realizadas
db.createCollection("Tipo_Dado"); // Coleção para armazenar os tipos de dados
    coletados
db.createCollection("Dados_IoT"); // Coleção para armazenar os dados brutos dos
    sensores
```

5 INSERÇÃO DE DADOS NOS BANCOS RELACIONAL E NOSQL

Este capítulo descreve um sistema de simulação de coleta de dados provenientes de sensores IoT e a inserção desses dados em dois tipos de bancos de dados: *PostgreSQL* e MongoDB. O sistema é composto por scripts que simulam a coleta de dados, monitoram o desempenho do sistema e automatizam a execução dos processos. A Figura 7 ilustra o fluxo de dados e a interação entre os componentes do sistema.

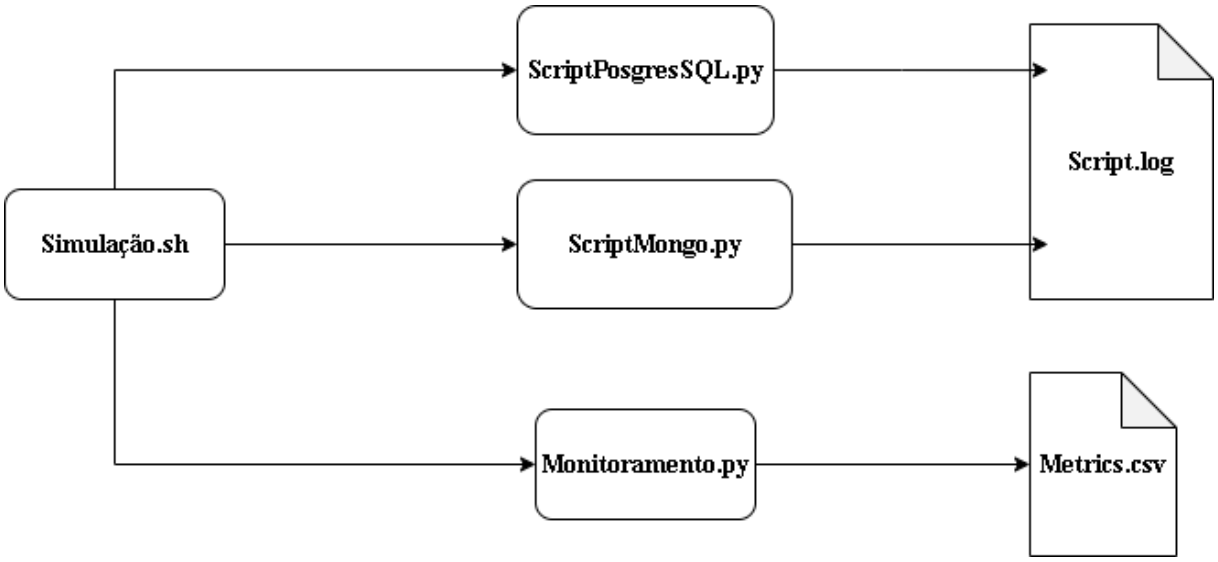

Figura 7 – Diagrama do sistema de simulação de sensores IoT e inserção de dados em *PostgreSQL* e *MongoDB*.

Fluxo de Execução do Sistema

O fluxo de execução do sistema envolve um conjunto de etapas estruturadas que garantem a coleta de métricas, a simulação de dispositivos IoT e o armazenamento dos dados em bancos de dados relacionais (PostgreSQL) ou NoSQL (MongoDB). Esse processo está representado na Figura 7 e pode ser descrito da seguinte forma:

1. **Execução do Script Principal**: O processo inicia com a execução do script `Simulacao.sh`, responsável por orquestrar a execução dos demais componentes do sistema. Esse script pode chamar diferentes processos, dependendo da configuração do experimento.

2. **Monitoramento de Recursos do Sistema**: O script `Monitoramento.py` é ativado para coletar métricas de desempenho, incluindo uso de CPU, consumo de memória, tempo de espera de I/O e tráfego de rede. Essas métricas são registradas periodicamente nos arquivos `1.csv`, `2.csv`, etc., permitindo análises posteriores sobre o comportamento do ambiente de execução.

3. **Simulação de Dispositivos IoT**: O script `Simulacao.sh` também pode chamar `ScriptPostgreSQL.py` ou `ScriptMongo.py`, dependendo da configuração do experimento. Esses scripts simulam a coleta de dados por sensores IoT, gerando leituras fictícias de variáveis como temperatura, umidade e pressão atmosférica.

4. **Armazenamento dos Dados**: Os scripts de simulação inserem os dados coletados em seus respectivos bancos de dados:

   - `ScriptPostgreSQL.py` insere os dados em tabelas estruturadas dentro do PostgreSQL, garantindo integridade referencial e concorrência controlada.
   - `ScriptMongo.py` insere os dados em coleções flexíveis no MongoDB, permitindo modelagem de dados mais adaptável a diferentes contextos de IoT.

   Durante esse processo, logs detalhados são gerados nos arquivos `ScriptPosgreSQL.log` e `ScriptMongo.log`, documentando as inserções realizadas e possíveis erros.

5. **Encerramento do Processo**: Após um período predefinido, o script `Simulacao.sh` finaliza a execução, encerrando os processos em segundo plano, incluindo os scripts de inserção de dados e monitoramento. Os arquivos de log gerados ao longo da simulação são armazenados para posterior análise de desempenho e comportamento do sistema.

O sistema proposto é composto por um conjunto de scripts que simulam um ambiente de Internet das Coisas (IoT), monitoram o desempenho do sistema e armazenam dados em bancos de dados relacionais (PostgreSQL) e não relacionais (MongoDB). A execução desses scripts é orquestrada por um script principal em Bash, que garante a sincronização e o controle do tempo de execução. A seguir, detalha-se o funcionamento de cada componente, sua lógica de execução e a justificativa para as escolhas tecnológicas.

Simulação.sh: O Orquestrador do Sistema

O script `Simulação.sh` é o núcleo de controle do sistema, responsável por iniciar, monitorar e finalizar a execução dos demais scripts. Escrito em Bash, ele é projetado para ser executado em sistemas Unix/Linux e possui as seguintes funcionalidades:

- **Inicialização do Ambiente**: O script define parâmetros como a duração da simulação e o número de instâncias de scripts que serão executadas. Esses parâmetros podem ser ajustados conforme a necessidade do experimento.
- **Execução Paralela**: Utilizando um loop, o script inicia múltiplas instâncias dos scripts de simulação de dispositivos IoT (`ScriptPostgreSQL.py` e `ScriptMongoDB.py`) em segundo plano. Isso permite a simulação de um grande número de dispositivos simultaneamente, aumentando a carga no sistema e gerando dados mais representativos.
- **Monitoramento em Tempo Real**: O script inicia o `Monitoramento.py` em segundo plano, que coleta métricas de desempenho do sistema, como uso de CPU, memória, I/O de disco e tráfego de rede. Essas métricas são armazenadas em um arquivo CSV para análise posterior.
- **Controle de Tempo**: O script aguarda um tempo pré-definido (em segundos) antes de encerrar a execução. Esse tempo é configurável e permite que a simulação seja executada por períodos variados, dependendo dos objetivos do experimento.
- **Finalização e Limpeza**: Ao término do tempo especificado, o script encerra todos os processos em execução, garantindo que nenhum recurso do sistema fique alocado indevidamente.

Monitoramento.py: Coleta de Métricas de Desempenho

O script `Monitoramento.py` é responsável por coletar métricas de desempenho do sistema durante a execução da simulação. Escrito em Python, ele utiliza a biblioteca `psutil` para acessar informações do sistema, como uso de CPU, memória, I/O de disco e tráfego de rede. O funcionamento do script pode ser dividido nas seguintes etapas:

- **Coleta de Dados**: O script acessa diretamente os arquivos do sistema operacional (como `/proc/stat`, `/proc/meminfo` e `/proc/loadavg`) para obter informações detalhadas sobre o uso de recursos. Além disso, ele utiliza comandos externos, como `iostat`, para coletar métricas de I/O de disco.
- **Processamento das Métricas**: As métricas brutas são processadas para calcular valores percentuais, como o uso de CPU e memória, e para converter unidades (por exemplo, tráfego de rede em MB ou GB).
- **Armazenamento em CSV**: As métricas são armazenadas em um arquivo CSV, com um novo arquivo sendo criado a cada execução do script. O arquivo CSV contém colunas para cada métrica coletada, juntamente com um timestamp que registra o momento da coleta.

- **Execução Contínua**: O script é executado em um loop infinito, coletando métricas a cada 5 segundos. Ele pode ser interrompido manualmente ou pelo script `Simulação.sh`.

A escolha do Python para esse script se deve à sua facilidade de manipulação de dados e à disponibilidade de bibliotecas como `psutil`, que simplificam a coleta de métricas do sistema. O armazenamento em CSV permite que os dados sejam facilmente importados para ferramentas de análise, como Pandas ou Excel.

ScriptPostgreSQL.py: Simulação de Dispositivos IoT com PostgreSQL

O script `ScriptPostgreSQL.py` simula a geração de dados de dispositivos IoT e os armazena em um banco de dados PostgreSQL. Escrito em Python, ele utiliza a biblioteca `psycopg2` para interagir com o banco de dados. O funcionamento do script pode ser descrito da seguinte forma:

- **Conexão com o Banco de Dados**: O script estabelece uma conexão persistente com o banco de dados PostgreSQL, utilizando credenciais pré-definidas. A conexão é configurada para não ser autocommit, garantindo que as transações sejam controladas manualmente.
- **Geração de Dados**: O script gera dados simulados para domínios e dispositivos IoT. Cada domínio é representado por um nome e uma descrição, enquanto os dispositivos possuem atributos como identificador, tipo (sensor, atuador, câmera), status (ativo/inativo) e coordenadas geográficas (latitude, longitude, altitude).
- **Inserção em Lote**: Para otimizar o desempenho, os dados são inseridos em lote no banco de dados. Isso reduz o número de operações de rede e melhora a eficiência da inserção.
- **Execução Paralela**: O script é projetado para ser executado em múltiplas instâncias simultaneamente, simulando um grande número de dispositivos IoT. Cada instância gera um conjunto único de dados, garantindo que não haja conflitos entre os registros.

ScriptMongoDB.py: Simulação de Dispositivos IoT com MongoDB

O script `ScriptMongoDB.py` tem uma função semelhante ao `ScriptPostgreSQL.py`, mas armazena os dados em um banco de dados MongoDB. Escrito em Python, ele utiliza a biblioteca `pymongo` para interagir com o banco de dados NoSQL. O funcionamento do script pode ser descrito da seguinte forma:

- **Conexão com o MongoDB**: O script estabelece uma conexão com o MongoDB, configurando um pool de conexões para suportar um grande número de operações simultâneas. **Geração de Dados**: O script simula a coleta de dados de sensores IoT, como temperatura, umidade e outros valores aleatórios. Cada registro contém o ID do sensor, o valor coletado e um timestamp.
- **Inserção em Lote**: Para otimizar o desempenho, os dados são inseridos em lote no MongoDB. O script utiliza a função `bulk_write` para realizar inserções em massa, reduzindo o overhead de rede e melhorando a eficiência.
- **Processamento em Paralelo**: O script utiliza múltiplas threads para processar os dados e inseri-los no banco de dados. Isso permite que a simulação lide com um grande volume de dados de forma eficiente.

O sistema proposto é uma solução para a simulação de ambientes IoT, combinando a geração de dados, o monitoramento de desempenho e o armazenamento em bancos de dados relacionais e não relacionais. A escolha das tecnologias utilizadas (Bash, Python, PostgreSQL e MongoDB) foi baseada em suas características de desempenho, flexibilidade e facilidade de integração. A execução coordenada dos scripts permite a coleta de dados estruturados para análise, facilitando a avaliação do desempenho do sistema em diferentes cenários.

O uso de Python para os scripts de monitoramento e inserção de dados demonstra a versatilidade da linguagem em tarefas de manipulação de dados e integração com bancos de dados. Já o Bash se mostrou ideal para a automação do fluxo de execução, garantindo que todos os componentes do sistema funcionem de forma sincronizada. Por fim, a combinação de PostgreSQL e MongoDB permite a comparação entre abordagens relacionais e não relacionais.

6 ANÁLISE DOS DESEMPENHOS DOS BANCOS DE DADOS

Neste capítulo, foi conduzida uma análise comparativa do desempenho dos bancos de dados MongoDB e PostgreSQL, com base em métricas como uso de CPU, espera de I/O, carga média do sistema, uso de memória e tráfego de rede. Os gráficos apresentados fornecem percepções sobre como cada banco de dados se comporta sob diferentes cargas de trabalho, destacando as características intrínsecas de cada sistema que influenciam seu desempenho. Para a realização da coleta de dados deste estudo, foi configurado um servidor virtualizado utilizando a plataforma *VirtualBox*. Esse servidor foi configurado com os seguintes recursos: 100 GB de armazenamento SSD, 6 *vCPUs* e 16 GB de memória RAM, utilizando o sistema operacional Debian 12. A escolha do Debian e da interface de linha de comando (*CLI*) foi feita visando minimizar o impacto no desempenho do servidor, garantindo que os dados coletados refletissem com maior precisão as condições experimentais, sem a interferência de processos gráficos ou de interface.

A coleta de dados foi realizada em incrementos de 100 dispositivos, começando em 100 e indo até 500 dispositivos. Cada rodada de coleta durou 10 minutos, permitindo que os bancos de dados atingissem um estado estável antes da medição. Essa abordagem permitiu observar como o aumento no número de dispositivos afetou o desempenho de cada banco de dados.

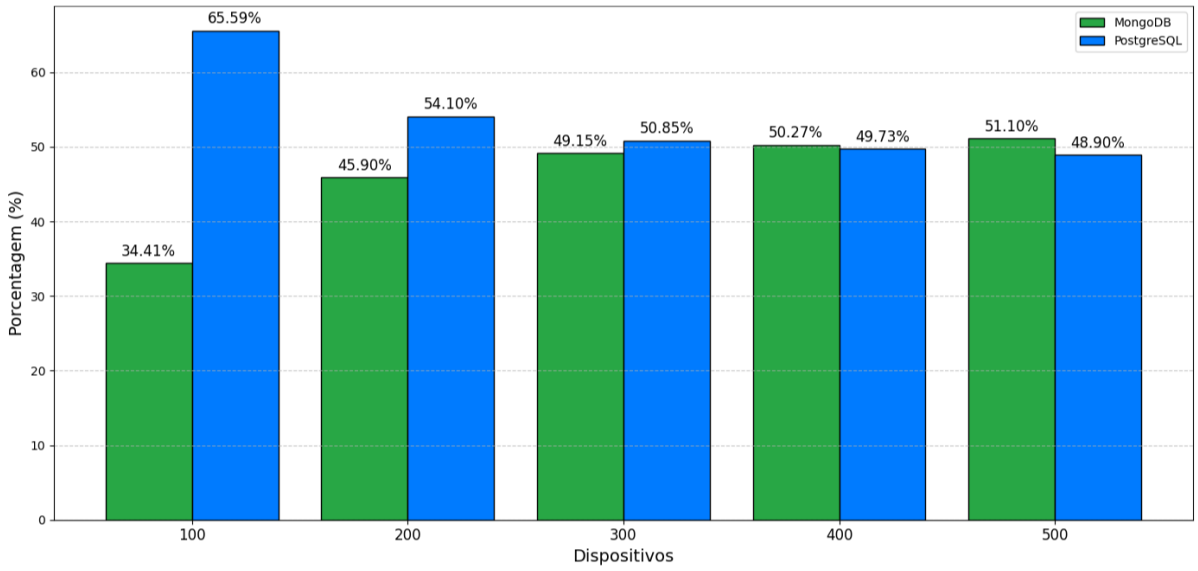

Figura 8 – Comparação do uso de CPU entre MongoDB e PostgreSQL.

A Figura 8 apresenta a comparação do uso de CPU entre o MongoDB e o Post-

greSQL. O gráfico mostra que o PostgreSQL tende a utilizar uma porcentagem maior da CPU, 65,59% em comparação com o MongoDB, 34,41%. Essa diferença pode ser atribuída à arquitetura do PostgreSQL, que é um banco de dados relacional tradicional, exigindo mais recursos em operações que envolvem consistência transacional e integridade referencial. Por outro lado, o MongoDB, por ser um banco de dados NoSQL orientado a documentos e armazenar dados em formato BSON, mostra-se mais eficiente nesse cenário, com menor consumo de CPU, especialmente em aplicações que requerem maior flexibilidade e escalabilidade.

Conforme o número de dispositivos aumentou de 100 para 500, o uso de CPU pelo MongoDB mostrou uma tendência de crescimento, atingindo picos em 500 dispositivos. Isso ocorre porque, com mais dispositivos, o MongoDB precisa gerenciar mais conexões e processar mais operações simultaneamente. O PostgreSQL, por outro lado, manteve um uso de CPU mais estável, demonstrando sua eficiência em operações estruturadas e transacionais, mesmo com o aumento da carga.

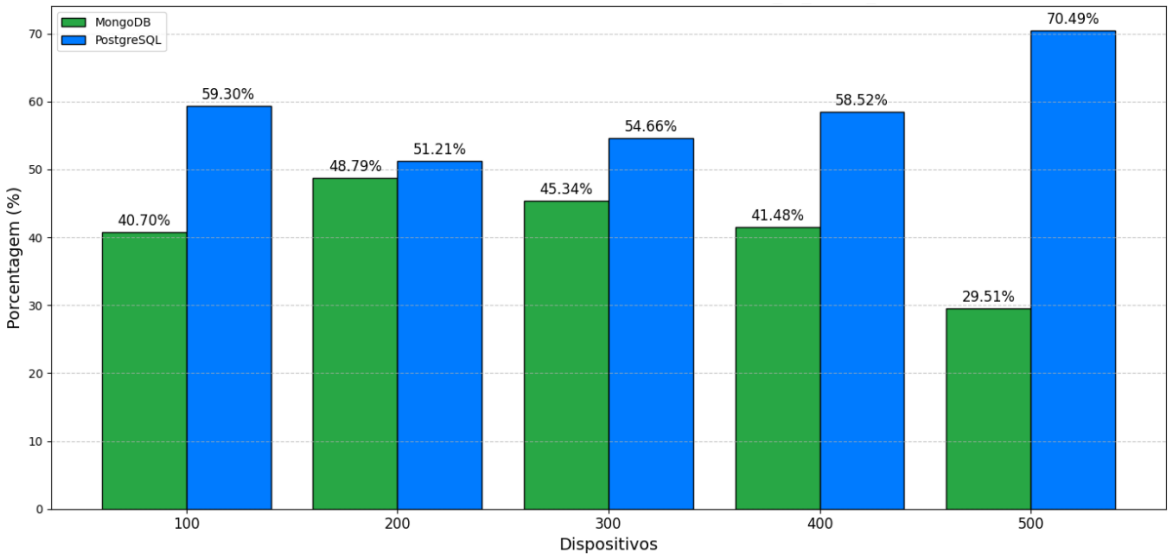

Figura 9 – Comparação da espera de I/O entre MongoDB e PostgreSQL.

A Figura 9 apresenta a comparação da espera de I/O entre o MongoDB e o PostgreSQL. O gráfico indica que o PostgreSQL apresenta um tempo de espera de I/O mais alto, 51,21% em comparação com o MongoDB, 48,79%. Esse resultado pode ser atribuído à natureza relacional do PostgreSQL, que, ao lidar com transações complexas e operações que envolvem integridade referencial, tende a realizar acessos mais intensivos ao disco. Em contrapartida, o MongoDB, por utilizar um modelo de dados orientado a documentos e otimizado para operações de leitura e escrita em grande escala, apresenta menor latência de I/O nesse cenário, especialmente quando os dados são distribuídos e parcialmente mantidos em memória. Com o aumento no número de dispositivos, a espera

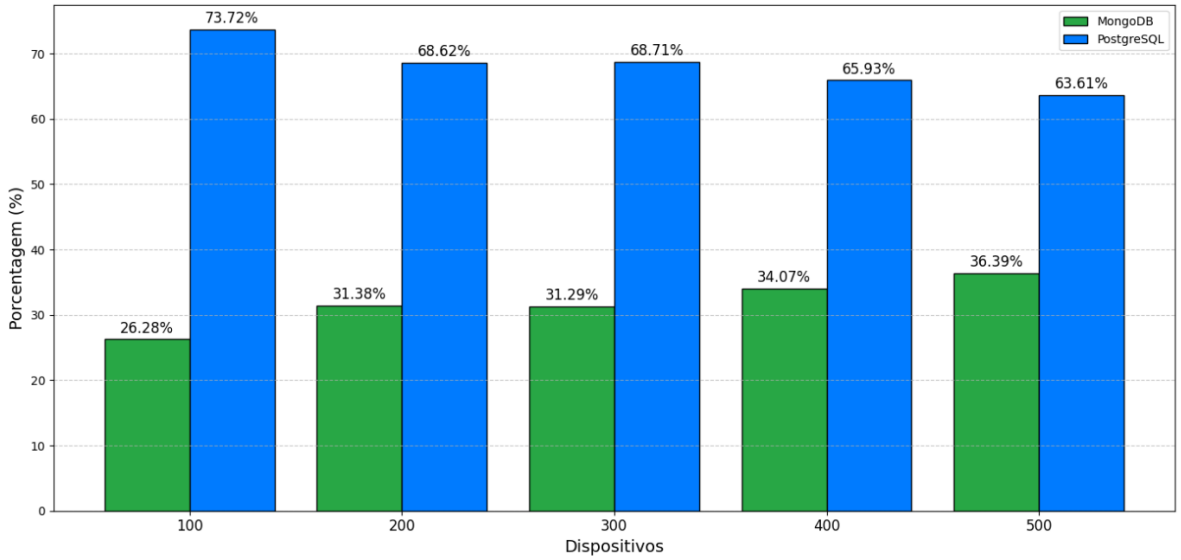

Figura 10 – Comparação do uso de memória entre MongoDB e PostgreSQL.

de I/O do MongoDB aumentou significativamente, especialmente em 500 dispositivos. Isso se deve ao maior volume de operações de leitura e escrita necessárias para gerenciar mais dispositivos. O PostgreSQL, por outro lado, manteve uma espera de I/O mais baixa e estável, graças à sua capacidade de otimizar operações de disco e ao uso eficiente de índices B-tree.

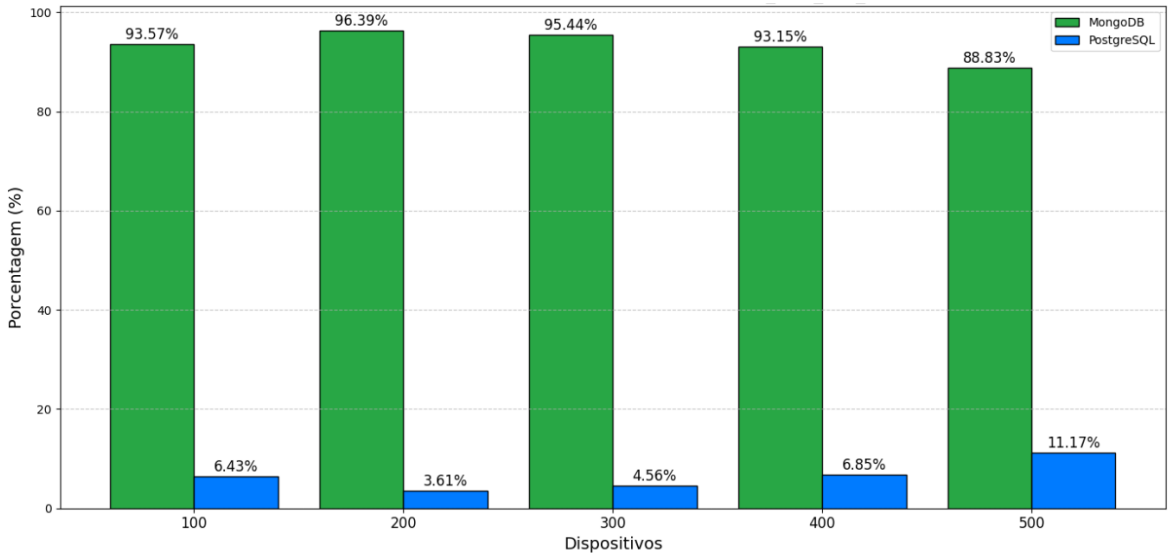

Figura 11 – Comparação da carga média do sistema entre MongoDB e PostgreSQL.

A Figura 11 apresenta a comparação da carga média do sistema entre o MongoDB e o PostgreSQL. A carga média do sistema é uma métrica que reflete a quantidade de trabalho que o sistema está executando. O PostgreSQL apresenta uma carga média do

sistema mais alta, 96,39% em comparação com o MongoDB, 3,61%. Esse resultado pode ser explicado pelas características do PostgreSQL, que, ao seguir um modelo relacional tradicional, realiza diversas operações com forte controle transacional e verificação de integridade, o que pode aumentar a carga do sistema. Em contraste, o MongoDB, com sua arquitetura orientada a documentos e foco em escalabilidade e desempenho em tempo real, consegue manter uma carga de sistema significativamente menor mesmo sob alta demanda. Com o aumento no número de dispositivos, a carga média do sistema do PostgreSQL

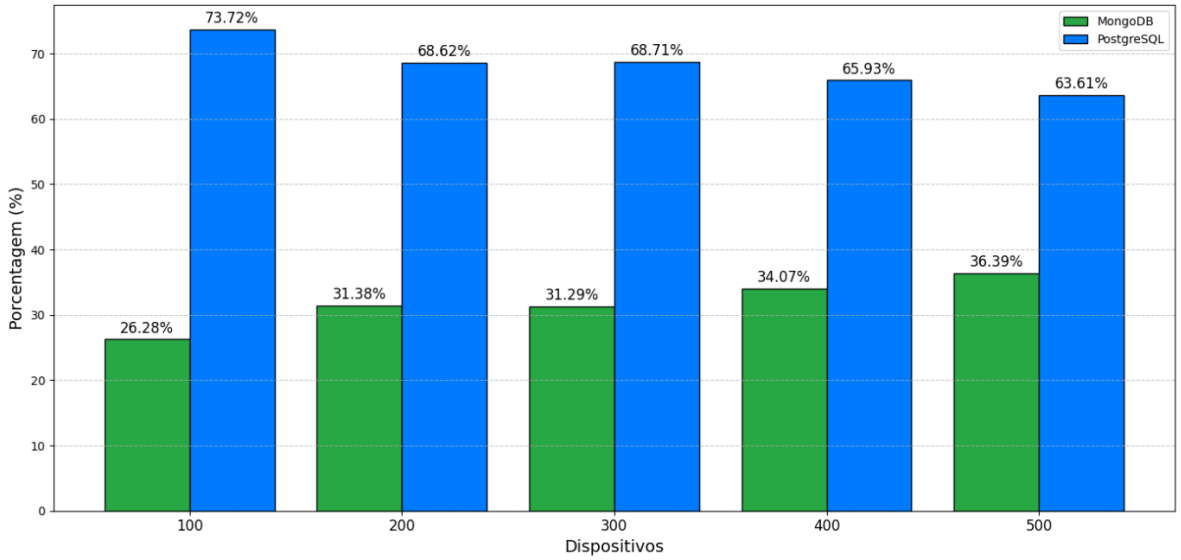

Figura 12 – Comparação do uso de memória entre MongoDB e PostgreSQL.

aumentou de forma mais acentuada, especialmente com 500 dispositivos. Isso ocorre porque o PostgreSQL precisa lidar com um número maior de conexões simultâneas, mantendo a integridade transacional e a estrutura relacional dos dados, o que pode sobrecarregar o sistema. O MongoDB, por outro lado, manteve uma carga média do sistema mais estável, evidenciando sua capacidade de lidar com grandes volumes de operações paralelas de forma mais eficiente, graças à sua arquitetura orientada a documentos e otimizada para escalabilidade horizontal.

A Figura 12 apresenta a comparação do uso de memória entre o MongoDB e o PostgreSQL. O gráfico mostra que o PostgreSQL utiliza uma porcentagem maior de memória, 73,72% em comparação com o MongoDB, 26,28%. Esse resultado pode ser atribuído à natureza relacional do PostgreSQL, que demanda mais recursos para manter estruturas como tabelas, índices complexos e buffers de transações. Além disso, o PostgreSQL utiliza técnicas avançadas de caching e otimização de consultas, o que, embora beneficie o desempenho, também contribui para um maior consumo de memória.

Com o aumento no número de dispositivos, o uso de memória pelo PostgreSQL cresceu de forma significativa, atingindo os maiores picos com 500 dispositivos. Isso

reflete o aumento na complexidade das operações e no volume de dados estruturados que precisam ser processados. O MongoDB, por sua vez, manteve um uso de memória mais estável, beneficiando-se de sua arquitetura orientada a documentos e de um modelo de gerenciamento de memória mais leve e escalável.

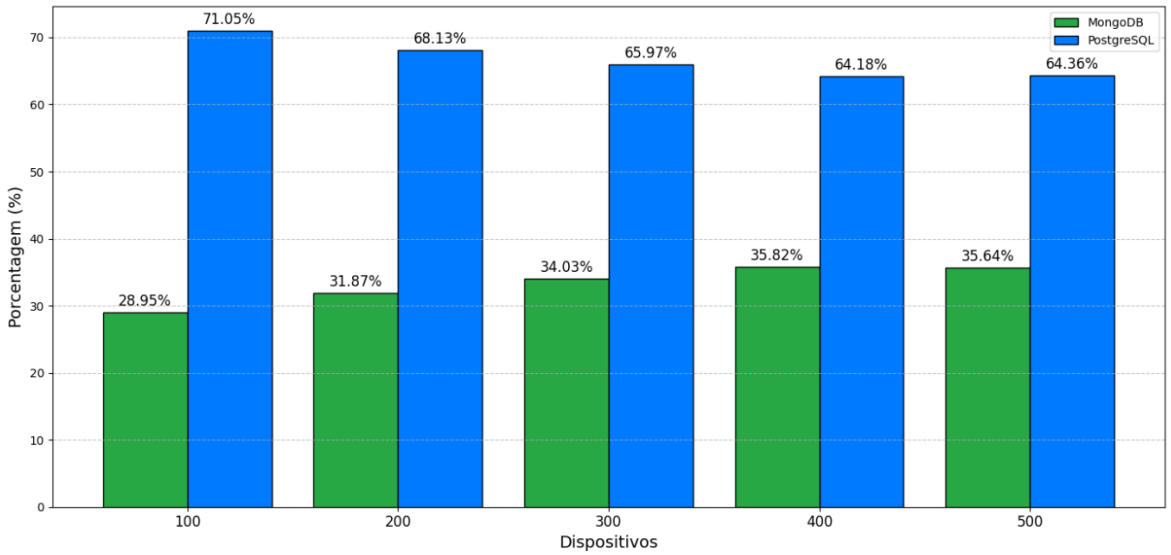

Figura 13 – Comparação do tráfego de rede de entrada entre MongoDB e PostgreSQL.

A Figura 13 apresenta a comparação do tráfego de rede de entrada entre o MongoDB e o PostgreSQL. O gráfico mostra que o PostgreSQL e o MongoDB têm porcentagens semelhantes de tráfego de entrada, com o PostgreSQL ligeiramente à frente em alguns dispositivos. Isso pode ser explicado pelo fato de que o PostgreSQL, operando como um sistema centralizado, concentra o recebimento de dados em um único ponto, o que pode resultar em maior tráfego de entrada.

Com o aumento no número de dispositivos, o tráfego de entrada do PostgreSQL aumentou de forma mais acentuada, especialmente com 500 dispositivos, devido à necessidade de receber, validar e armazenar grandes volumes de dados de forma estruturada. O MongoDB, por outro lado, manteve um tráfego de entrada mais estável, beneficiando-se de sua arquitetura distribuída, que permite o balanceamento da carga de rede entre diferentes nós do cluster.

A Figura 14 apresenta a comparação do tráfego de rede de saída entre o MongoDB e o PostgreSQL. O gráfico indica que o PostgreSQL tem um tráfego de saída mais alto, 62,30% em comparação com o MongoDB, 37,70%. Isso ocorre porque o PostgreSQL, mesmo sendo um banco de dados centralizado, realiza comunicações intensivas com clientes e serviços externos para manter a consistência e fornecer respostas detalhadas a consultas complexas. Essas interações elevam o tráfego de saída. O MongoDB, por outro lado, apesar de ser distribuído, otimiza a replicação entre nós, o que contribui para um tráfego de saída

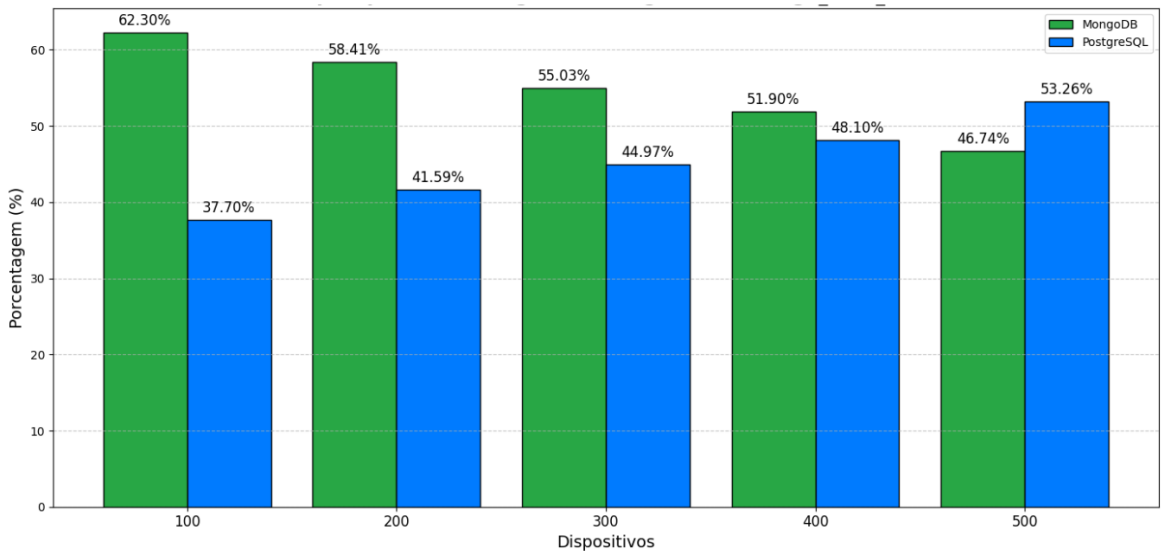

Figura 14 – Comparação do tráfego de rede de saída entre MongoDB e PostgreSQL.

mais contido.

Com o aumento no número de dispositivos, o tráfego de saída do PostgreSQL aumentou significativamente, especialmente em 500 dispositivos, reflexo do maior volume de dados processados e transmitidos para múltiplas requisições. O MongoDB manteve um tráfego de saída mais estável, graças à sua arquitetura escalável e eficiente em ambientes distribuídos.

A análise dos gráficos revela que o PostgreSQL tende a utilizar mais recursos (CPU, memória e rede) em comparação com o MongoDB. Isso está relacionado à sua arquitetura relacional, que exige maior processamento para garantir integridade, transações e consultas complexas. Como resultado, o PostgreSQL é ideal para aplicações que demandam precisão, controle e consistência em operações estruturadas.

Por outro lado, o MongoDB demonstra maior eficiência no uso de recursos em cenários com grandes volumes de dados não estruturados. Sua arquitetura orientada a documentos e distribuída permite escalabilidade e flexibilidade, mantendo um consumo de recursos mais equilibrado.

A escolha entre PostgreSQL e MongoDB deve, portanto, considerar as demandas específicas da aplicação — como estrutura dos dados, volume, escalabilidade e requisitos de desempenho. Ambos oferecem vantagens claras, e a decisão ideal depende de alinhar essas características às necessidades do projeto.

7 CONCLUSÃO

A análise comparativa do desempenho dos bancos de dados MongoDB e PostgreSQL mostrou claramente as características distintas de cada sistema em relação ao uso de recursos como CPU, memória, I/O, carga média do sistema e tráfego de rede.

O MongoDB, por ser um banco de dados NoSQL orientado a documentos, apresenta um uso mais intensivo de CPU, memória e tráfego de rede, especialmente em ambientes de alto volume de dados e conexões simultâneas. Isso se deve à sua arquitetura distribuída, que permite maior escalabilidade e flexibilidade, mas também implica em maior custo computacional e maior complexidade na gestão de dados em tempo real. A alta carga média do sistema e o aumento significativo no tempo de espera de I/O, especialmente com o aumento no número de dispositivos, indicam que o MongoDB pode ser mais exigente em termos de recursos à medida que a carga de trabalho aumenta.

Por outro lado, o PostgreSQL, um banco de dados relacional, demonstrou ser mais eficiente no uso de recursos, com menor uso de CPU e memória, bem como um tráfego de rede mais controlado. Sua arquitetura relacional e otimizações para consultas complexas o tornam ideal para ambientes que exigem alta consistência, integridade de dados e transações complexas, sem demandar uma quantidade excessiva de recursos computacionais. A carga média do sistema e a espera de I/O se mantiveram mais estáveis, mesmo com o aumento no número de dispositivos, o que sugere que o PostgreSQL é mais adequado para cenários de processamento de dados estruturados e workloads transacionais.

A escolha entre MongoDB e PostgreSQL, portanto, depende das necessidades específicas da aplicação. O MongoDB é mais indicado para projetos que exigem escalabilidade horizontal, flexibilidade no esquema de dados e alta disponibilidade, enquanto o PostgreSQL é mais vantajoso em cenários onde a consistência e a integridade dos dados, bem como a eficiência em transações complexas, são prioritárias.

Ambos os bancos de dados têm seu papel e vantagens dependendo do tipo de aplicação e dos requisitos de desempenho. A decisão sobre qual banco de dados utilizar deve ser baseada em uma análise cuidadosa das características de cada sistema e das demandas específicas do ambiente de produção.

REFERÊNCIAS

ASHTON, K. That 'internet of things' thing. *RFID Journal*, 2009. Accessed: 2024-08-27. Citado na página 16.

_____. That 'internet of things' thing. *RFID Journal*, v. 22, n. 7, p. 97, 2009. Citado na página 24.

ATZORI, L.; IERA, A.; MORABITO, G. The internet of things: A survey. *Computer networks*, Elsevier, v. 54, n. 15, p. 2787–2805, 2010. Citado na página 22.

_____. The internet of things: A survey. *Computer networks*, Elsevier, v. 54, n. 15, p. 2787–2805, 2010. Citado na página 26.

AZAD, P.; NAVIMIPOUR, N. J.; RAHMANI, A. M.; SHARIFI, A. The role of structured and unstructured data managing mechanisms in the internet of things. *Cluster Computing*, Kluwer Academic Publishers, USA, v. 23, n. 2, p. 1185–1198, jun. 2020. ISSN 1386-7857. Disponível em: <https://doi.org/10.1007/s10586-019-02986-2>. Citado na página 42.

BRáS, J.; PEREIRA, R.; MORO, S. Intelligent process automation and business continuity: Areas for future research. *Information*, v. 14, n. 2, 2023. ISSN 2078-2489. Disponível em: <https://www.mdpi.com/2078-2489/14/2/122>. Citado na página 24.

CATTELL, R. G. G. *Scalable SQL and NoSQL Data Stores*. [S.l.]: ACM Computing Surveys, 2011. v. 43. 1-39 p. Citado na página 17.

CHEN, M.; MAO, S.; ZHANG, Y.; CHEN, J.; WU, H.; DAS, S. K. A survey of the internet of things (iot) and its applications. *IEEE Access*, v. 5, p. 8196–8218, 2017. Citado na página 17.

CHODOROW, K. *MongoDB: The Definitive Guide: Powerful and Scalable Data Storage*. [S.l.]: O'Reilly Media, 2013. Citado na página 33.

CODD, E. F. A relational model of data for large shared data banks. *Communications of the ACM*, v. 13, n. 6, p. 377–387, 1970. Citado na página 27.

DATE, C. J. *An Introduction to Database Systems*. [S.l.]: Addison-Wesley, 2004. Citado 4 vezes nas páginas 26, 27, 28 e 29.

ELMASRI, R.; NAVATHE, S. B. *Database System Concepts*. 7. ed. [S.l.]: McGraw-Hill Education, 2020. Citado na página 17.

GARCIA, V. S.; SOTTO, E. C. S. Comparativo entre os modelos de banco de dados relacional e não-relacional. *Revista Interface Tecnológica*, Faculdade de Tecnologia de Catanduva, v. 16, n. 2, p. 673, 2019. Citado na página 27.

GEORGE, L. *Cassandra: The Definitive Guide: Distributed Data at Web Scale*. [S.l.]: O'Reilly Media, 2011. Citado na página 33.

GUBBI, J.; BUYYA, R.; MARUSIC, S.; PALANISWAMI, M. Internet of things (iot): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, v. 29, n. 7, p. 1645–1660, 2013. Citado 2 vezes nas páginas 16 e 22.

HAN, J.; KAMBER, M.; PEI, J. *Data Mining: Concepts and Techniques*. [S.l.]: Morgan Kaufmann, 2011. Citado 3 vezes nas páginas 18, 31 e 32.

HE, H.; HU, S.; ZHONG, Y. Modeling and optimization for the internet of things: Challenges and solutions. *ACM Computing Surveys*, v. 49, n. 2, p. 1–39, 2016. Citado na página 17.

INAMASU, R. Y.; NAIME, J. d. M.; RESENDE, A. V. d.; BASSOI, L. H.; BERNARDI, A. C. d. C. *Agricultura de precisão: um novo olhar*. [S.l.]: Embrapa, 2015. Citado na página 25.

KORTH, H. F.; SILBERSCHATZ, A. *Database System Concepts*. 7. ed. [S.l.]: McGraw-Hill Education, 2019. Citado na página 27.

KOULAMAS, C.; LAZARESCU, M. T. Real-time sensor networks in industry 4.0. *Journal of Industrial Information Integration*, Elsevier, v. 18, p. 100131, 2020. Citado na página 25.

KUMAR, N.; MALLICK, P. K.; GUPTA, H. Security and privacy issues in internet of things (iot): A survey. *Journal of Computing and Security*, v. 58, p. 1–15, 2016. Citado na página 16.

LAKSHMAN, A.; MALIK, P. Cassandra: A decentralized structured storage system. *ACM SIGOPS Operating Systems Review*, v. 44, n. 2, p. 35–40, 2010. Citado na página 33.

LEE, J. *Industrial IoT: Real-time sensor networks and automation*. [S.l.]: Springer, 2019. Citado na página 25.

MIRANDA, A. C. C. d.; VERíSSIMO, A. M.; CEOLIN, A. C. Agricultura de precisão: um mapeamento da base da scielo. *Gestão Produção*, UFPE, v. 15, n. 1, p. 129–137, 2017. Citado na página 25.

NASKAR, S.; BASU, P.; SEN, A. K. A literature review of the emerging field of iot using rfid and its applications in supply chain management. In: LEE, I. (Ed.). *The Internet of Things in the Modern Business Environment*. [S.l.]: IGI Global, 2017. p. 1–27. Citado na página 24.

OLIVEIRA, M. M. A.; CARLOS, D. G.; SOUSA, A. R. V. O.; CASTRO, A. F. Um estudo comparativo entre banco de dados orientado a objetos, banco de dados relacionais e framework para mapeamento objeto/relacional, no contexto de uma aplicação web. *Holos*, Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte, v. 1, p. 182–198, 2015. Citado na página 26.

SADALAGE, P. J.; FOWLER, M. *NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence*. [S.l.]: Addison-Wesley, 2012. Citado 2 vezes nas páginas 30 e 31.

_____. *NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence*. [S.l.]: Addison-Wesley, 2013. Citado na página 33.

SHENG, Q. Z.; TANG, S.; YAO, L. Data management in the internet of things: challenges and solutions. *IEEE Internet of Things Journal*, IEEE, v. 1, n. 5, p. 430–446, 2013. Citado 2 vezes nas páginas 24 e 25.

SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. *Database System Concepts*. [S.l.]: McGraw-Hill, 2010. Citado 2 vezes nas páginas 27 e 28.

_____. *Sistemas de Banco de Dados*. [S.l.]: McGraw-Hill, 2011. Citado na página 26.

STONEBRAKER, M.; CEYLAN, U.; DEWITT, D.; S., D. P. The end of an architectural era (it's time for a complete rewrite). *Communications of the ACM*, v. 53, n. 6, p. 46–55, 2010. Citado na página 17.

STRAUCH, C. *Mastering MongoDB 3.x: A Practical Guide to the Latest Version of MongoDB*. [S.l.]: Packt Publishing, 2017. Citado na página 33.

STRAUCH, S. *NoSQL for Dummies*. [S.l.]: Wiley, 2013. Citado 3 vezes nas páginas 30, 31 e 34.

TSCHIEDEL, M.; FERREIRA, M. F. Introdução à agricultura de precisão: conceitos e vantagens. *Ciência Rural*, SciELO, v. 30, n. 4, p. 731–738, 2000. Citado na página 25.

WHITE, T. *Hadoop: The Definitive Guide*. 4. ed. [S.l.]: O'Reilly Media, 2017. Citado na página 34.

YADAV, H. Structuring sql/ nosql databases for iot data. *International Journal of Machine Learning and Artificial Intelligence*, v. 5, n. 5, p. 1–12, Apr. 2024. Disponível em: <https://jmlai.in/index.php/ijmlai/article/view/27>. Citado na página 43.

ZHANG, L.; ZHAO, Z.; ZHANG, H.; LIANG, W. A survey of security and privacy issues in internet of things. *IEEE Internet of Things Journal*, v. 7, n. 5, p. 4000–4016, 2020. Citado na página 17.