

Governança de TI com COBIT: Eficiência e Controle em Ambientes Terceirizados

BERNARDO, P. R. A.¹, VERGA, M. D.², GRAZIOSI, S. E.³

1 Docente em Administração e Sistemas de Informação no Instituto Matonense Municipal de Ensino Superior (IMMES). 2 Docente em Administração e Sistemas de Informação no Instituto Matonense Municipal de Ensino Superior (IMMES) 3 Docente em Sistemas de Informação no Instituto Matonense Municipal de Ensino Superior (IMMES).

RESUMO: Este artigo apresenta uma análise comparativa do desempenho de diferentes Sistemas de Gerenciamento de Bancos de Dados (SGBDs), incluindo soluções relacionais (PostgreSQL, MySQL, SQL Server), não relacionais (MongoDB, Neo4j) e bancos em memória. O estudo visa avaliar como essas tecnologias se comportam em cenários distintos, considerando operações de inserção, recuperação, agregação e consultas complexas, especialmente em grandes volumes de dados. A metodologia inclui a revisão sistemática de literatura e a análise experimental baseada em testes de benchmark documentados em trabalhos acadêmicos, como os de Silva (2023), Ferreira (2013) e Souza (2023). Foram utilizados critérios como tempo de resposta, consumo de recursos e escalabilidade em diferentes plataformas e linguagens de programação (C e Python). Os resultados indicam que bancos relacionais apresentam melhor desempenho em consultas estruturadas com alto índice de normalização, enquanto bancos NoSQL se destacam em escalabilidade e flexibilidade para dados semiestruturados. O Neo4j demonstrou superioridade em consultas de grafos, mas com limitações em cargas massivas. Por sua vez, variações no desempenho entre linguagens foram observadas, com C superando Python em tarefas de inserção intensiva. Conclui-se que a escolha do SGBD ideal depende fortemente do tipo de aplicação, volume de dados e perfil de consulta, reforçando a importância de critérios técnicos na tomada de decisão arquitetural em projetos de software.

Palavras-chave: Banco de Dados, SQL, NoSQL, Desempenho, PostgreSQL, MongoDB, Neo4j.

ABSTRACT: *This paper presents a comparative analysis of the performance of different Database Management Systems (DBMSs), including relational (PostgreSQL, MySQL, SQL Server), non-relational (MongoDB, Neo4j) and in-memory databases. The study aims to evaluate how these technologies behave in different scenarios, considering insertion, retrieval, aggregation and complex query operations, especially in large volumes of data. The methodology includes a systematic literature review and experimental analysis based on benchmark tests documented in academic works, such as those by Silva (2023), Ferreira (2013) and Souza (2023). Criteria such as response time, resource consumption and scalability on different platforms and programming languages (C and Python) were used. The results indicate that relational databases perform better in structured queries with a high normalization index, while NoSQL databases stand out in scalability and flexibility for semi-structured data. Neo4j demonstrated superiority in graph queries, but with limitations in massive loads. In turn, variations in performance between languages were observed, with C outperforming Python in intensive insertion tasks. It is concluded that the choice of the ideal DBMS depends strongly on the type of application, data volume and query profile, reinforcing the importance of technical criteria in architectural decision-making in*

software projects.

Keywords: *Database, SQL, NoSQL, Performance, PostgreSQL, MongoDB, Neo4j.*

1. INTRODUÇÃO

O avanço exponencial da geração de dados tem exigido soluções cada vez mais eficientes para o armazenamento, processamento e recuperação de informações. Em um cenário marcado por aplicações distribuídas, big data e sistemas interconectados, a escolha adequada do Sistema de Gerenciamento de Bancos de Dados (SGBD) torna-se um fator estratégico. Tradicionalmente, os bancos de dados relacionais (SQL) têm dominado o mercado, porém a crescente demanda por escalabilidade e flexibilidade impulsionou o uso de bancos NoSQL, que adotam modelos diferentes de armazenamento e acesso aos dados. Essa diversidade tecnológica exige análises criteriosas sobre o desempenho e a aplicabilidade de cada arquitetura em diferentes cenários. Estudos como os de Ferreira (2013) e Silva (2023) comparam diferentes SGBDs quanto ao tempo de resposta, consumo de recursos e eficiência em operações de leitura e escrita. Enquanto os bancos relacionais como MySQL e PostgreSQL demonstram alta consistência e desempenho em consultas estruturadas, os bancos NoSQL como MongoDB e Neo4j destacam-se na manipulação de grandes volumes de dados semiestruturados. No entanto, autores como Zimmer (2023) apontam que o desempenho pode variar significativamente conforme a linguagem de programação utilizada, os drivers adotados e a complexidade das operações. Ainda há lacunas sobre a comparação direta entre diferentes arquiteturas sob condições experimentais equivalentes.

Apesar do grande volume de estudos individuais sobre bancos de dados específicos, ainda são escassas as análises que exploram simultaneamente múltiplos paradigmas (relacional, NoSQL, grafo, em memória) em contextos comparáveis. Essa ausência limita a compreensão sobre qual tecnologia é mais adequada para cada tipo de aplicação, considerando fatores como inserção massiva, travessia de relacionamentos, consultas agregadas e desempenho sob diferentes linguagens. Além disso, poucos trabalhos integram aspectos como custo computacional e facilidade de desenvolvimento nos critérios de comparação.

Diante desse panorama, o presente artigo tem como objetivo comparar o desempenho de diferentes SGBDs — relacionais e não relacionais — em uma perspectiva multiplataforma. Serão considerados bancos como PostgreSQL, MySQL, MongoDB e Neo4j, avaliando-os sob tarefas distintas com uso das linguagens C e Python. A análise incluirá indicadores de tempo, consumo de memória e adequação ao tipo de consulta, contribuindo para auxiliar profissionais e pesquisadores na escolha do banco mais eficiente para suas necessidades específicas.

2. OBJETIVOS

Objetivo Geral:

Comparar o desempenho de diferentes Sistemas de Gerenciamento de Bancos de Dados (SGBDs) — relacionais e não relacionais — em uma perspectiva multiplataforma, considerando distintos tipos de consultas, volumes de dados e linguagens de programação.

Objetivos Específicos

- Avaliar o desempenho de SGBDs relacionais (PostgreSQL, MySQL) em tarefas de inserção, recuperação e agregação de dados.
- Analisar a performance de bancos NoSQL e orientados a grafos (MongoDB, Neo4j) em consultas semiestruturadas e de travessia.
- Comparar o impacto das linguagens de programação C e Python na execução de operações em um mesmo banco de dados.
- Identificar os cenários em que cada arquitetura apresenta maior eficiência, considerando também fatores como escalabilidade e custo computacional.

Segundo Ferreira (2013), a escolha do SGBD deve considerar o perfil da aplicação e o volume de dados, pois variações de desempenho podem ser significativas entre sistemas. Já Zimmer (2023) destaca que o desempenho não depende apenas do banco utilizado, mas também da linguagem e dos drivers de conexão, o que justifica a análise conjunta desses fatores neste estudo.

3. REVISÃO DA LITERATURA

O crescimento exponencial dos dados digitais nas últimas décadas tem impulsionado a necessidade por sistemas de armazenamento mais eficientes e flexíveis. Segundo Silberschatz, Korth e Sudarshan (2010), os bancos de dados relacionais são projetados para garantir integridade, consistência e desempenho em operações estruturadas. No entanto, com a diversificação dos dados, especialmente em formatos não tabulares, surgiram alternativas como os bancos NoSQL, que oferecem maior flexibilidade estrutural. Como apontam Han et al. (2011), a escalabilidade horizontal dos sistemas NoSQL os torna mais adequados a contextos de big data e aplicações em tempo real.

Os bancos relacionais, como MySQL, PostgreSQL e SQL Server, ainda são amplamente utilizados em aplicações corporativas, principalmente pela maturidade, suporte a transações ACID e compatibilidade com linguagens SQL padrão. Ferreira (2013) realizou uma análise de desempenho entre esses sistemas, demonstrando que o PostgreSQL obteve melhores resultados em inserções e o MySQL em consultas simples. Por outro lado, Rodrigues e Trad Júnior (2013) alertam que esses bancos, quando mal configurados, não aproveitam todo seu potencial, comprometendo o desempenho em grandes cargas de dados.

Os bancos NoSQL, como MongoDB e Cassandra, adotam modelos orientados a documentos, colunas ou grafos, permitindo a manipulação eficiente de dados semiestruturados. Segundo Silva (2023), o MongoDB destaca-se em aplicações que demandam flexibilidade e desempenho em leitura distribuída. Já Neo4j, orientado a grafos, é recomendado para aplicações que exigem navegação entre entidades conectadas, como redes sociais ou sistemas de recomendação. Contudo, Robinson, Webber e Eifrem (2015) alertam que esse tipo de banco pode sofrer gargalos em operações de escrita em massa ou com grafos densos.

A escolha da linguagem de programação também influencia diretamente o desempenho das operações em SGBDs. Zimmer (2023) comparou C e Python em consultas ao MySQL e observou que, embora as diferenças em leitura fossem pequenas, o tempo de inserção em C foi significativamente menor. Isso confirma as observações de Alomari et al. (2015), que destacam o impacto da tipagem, gerenciamento de memória e estratégias de execução no tempo de resposta. A combinação banco-linguagem, portanto, deve ser cuidadosamente avaliada conforme o perfil do sistema.

Apesar das diversas análises disponíveis na literatura, poucos estudos realizam comparações abrangentes entre diferentes tipos de bancos (relacional, NoSQL, grafos) utilizando múltiplas linguagens de programação sob condições experimentais equivalentes. Como apontam Reinsel et al. (2017), o aumento no volume de dados exige decisões arquiteturais mais criteriosas, que considerem não apenas o desempenho, mas também a escalabilidade, complexidade de implementação e custos operacionais. Essa lacuna justifica a proposta deste artigo, que busca integrar diferentes perspectivas em uma análise comparativa de desempenho mais ampla e aplicada.

4. METODOLOGIA

Este estudo adota uma abordagem exploratória e descritiva com base em revisão sistemática da literatura e análise comparativa de resultados empíricos já documentados em estudos acadêmicos. Segundo Gil (2008), a abordagem exploratória é adequada para temas ainda pouco consolidados, permitindo a investigação de múltiplas variáveis sob diferentes condições. Foram utilizados como referência os trabalhos de Silva (2023), Ferreira (2013), Zimmer (2023) e Rodrigues (2013), que apresentam análises experimentais de desempenho em bancos de dados relacionais e não relacionais.

A seleção dos estudos considerou três critérios principais: (1) foco em desempenho de SGBDs em diferentes contextos; (2) aplicação de testes com métricas claras e reproduzíveis (tempo de inserção, consulta e uso de recursos); e (3) uso de linguagens de programação distintas ou tecnologias de banco variadas (SQL, NoSQL, grafos). A triangulação dos resultados entre diferentes fontes busca aumentar a validade dos achados, como sugerem Patton (2002) e Marconi e Lakatos (2003). Não foram incluídos trabalhos com dados incompletos ou sem descrição metodológica detalhada.

Foram considerados os seguintes SGBDs nas análises: MySQL e PostgreSQL (relacionais), MongoDB (documento/NoSQL) e Neo4j (grafo). Além disso, foram analisados os impactos da utilização de duas linguagens de programação (C e Python) na execução das mesmas operações, com base nos testes realizados por Zimmer (2023). As operações observadas incluíram inserções simples e massivas, consultas com filtros, agregações e travessia de relacionamentos. O foco esteve em medir o tempo de execução e avaliar o consumo de recursos computacionais, quando disponível.

A comparação entre os bancos foi organizada em três eixos principais: (1) desempenho de leitura e escrita; (2) escalabilidade e resposta sob cargas elevadas; (3) adaptabilidade às linguagens utilizadas. Conforme Bardin (2011), a categorização temática foi empregada para organizar os dados em torno dessas dimensões, permitindo a construção de uma análise crítica integrada. Foram utilizadas tabelas e gráficos comparativos quando presentes nos artigos para melhor visualização dos resultados.

Embora o estudo se baseie em dados secundários, a combinação de diferentes autores e abordagens amplia o escopo da análise e permite maior robustez interpretativa. Como destacam Guba e Lincoln (1994), a validade de estudos qualitativos depende da consistência das fontes e da clareza nos critérios de análise. Reconhece-se, contudo, que a ausência de experimentos próprios limita a generalização dos resultados, sendo esse um aspecto tratado nas considerações finais.

RESULTADOS

O estudo de Silva (2023), com foco em revisão sistemática sobre SGBDs relacionais e NoSQL, identificou que bancos relacionais apresentam melhor desempenho em operações estruturadas com volumes moderados de dados. O MySQL se destacou em consultas simples, enquanto o PostgreSQL apresentou estabilidade superior em operações com índices e agregações. Já o MongoDB obteve vantagem em operações com dados semiestruturados e em leitura distribuída. A pesquisa apontou ainda que o desempenho varia de acordo com a estrutura do dataset e a padronização das consultas.

A análise de Ferreira (2013) incluiu testes comparativos com os bancos Firebird 2.5, PostgreSQL 9.2.1, MySQL 5.5 e SQL Server 2008. Os resultados indicaram que o PostgreSQL teve o melhor tempo de resposta em inserções massivas, enquanto o SQL Server apresentou melhor desempenho em consultas com múltiplos joins. O Firebird, por sua vez, demonstrou desempenho inferior nas operações de leitura e escrita, mas com menor uso de memória. O estudo utilizou benchmarks como TPC-C e AS3AP para uniformizar os testes.

Zimmer (2023) realizou uma avaliação entre as linguagens C e Python, ambas acessando um banco MySQL. Os resultados mostraram que, apesar de as consultas apresentarem tempos similares em ambas as linguagens, a inserção de grandes volumes de dados foi significativamente mais rápida em C. Essa diferença foi atribuída à compilação nativa e ao gerenciamento de memória da linguagem. O uso de diferentes drivers também influenciou os resultados, com o MySQL Connector/Python apresentando desempenho inferior ao driver nativo C.

No contexto de bancos orientados a grafos, os dados do estudo de Robinson, Webber e Eifrem (2015) demonstraram que o Neo4j teve desempenho superior na realização de consultas com múltiplas travessias de relacionamento. Entretanto, o sistema apresentou queda de desempenho em cenários com alta concorrência de escrita e grafos muito densos. A estrutura de armazenamento baseada em nós e arestas mostrou-se ideal para sistemas de recomendação e redes sociais, mas menos eficiente para operações agregadas.

Por fim, os resultados compilados indicam que cada arquitetura de banco de dados apresenta desempenho superior em cenários específicos. Bancos relacionais lideram em integridade e operações estruturadas, enquanto os NoSQL se destacam na escalabilidade horizontal e flexibilidade de dados. A combinação banco-linguagem também influenciou os resultados, especialmente em inserções. Esses dados oferecem subsídios concretos para análises mais aprofundadas, que serão discutidas na próxima seção.

5. DISCUSSÃO

Os resultados obtidos mostram que a escolha do SGBD deve estar fortemente alinhada ao tipo de aplicação e ao perfil das operações envolvidas. A performance superior dos bancos relacionais em operações estruturadas, como verificado por Ferreira (2013) e Silva (2023), confirma a robustez dessas plataformas em ambientes que exigem integridade transacional e precisão em consultas agregadas. Isso está em consonância com Silberschatz et al. (2010), que destacam a maturidade dos sistemas relacionais para aplicações empresariais críticas. No entanto, seu desempenho pode ser comprometido em operações de leitura massiva e dados altamente variáveis.

Por outro lado, os bancos NoSQL como MongoDB e os orientados a grafos como Neo4j apresentaram desempenho superior em cenários de alta escalabilidade e dados semiestruturados, especialmente quando não há necessidade de modelagem relacional rígida. Os resultados encontrados por Silva (2023) e Robinson et al. (2015) validam a aplicabilidade dessas tecnologias em contextos como redes sociais, big data e internet das coisas. Contudo, como argumenta Alomari et al. (2015), sua utilização exige cuidado com integridade dos dados e limites de consistência, o que pode representar um desafio para sistemas financeiros ou críticos.

A comparação entre linguagens, conforme realizada por Zimmer (2023), destaca a relevância da camada de acesso ao banco de dados na definição do desempenho final do sistema. O desempenho superior da linguagem C em operações de inserção sugere que aplicações com grande volume de dados podem se beneficiar de linguagens compiladas e drivers otimizados. Por outro lado, Python mostrou-se adequada para tarefas com menor demanda de desempenho bruto, mantendo a vantagem da simplicidade e da legibilidade do código.

Outro ponto relevante foi a constatação de que bancos orientados a grafos como o Neo4j superam bancos relacionais em travessias complexas de relacionamentos. No entanto, o desempenho sofre queda quando a estrutura de grafo se torna muito densa ou em situações com muitas operações concorrentes de escrita. Como argumentam Robinson, Webber e Eifrem (2015), esses sistemas são ideais para aplicações altamente conectadas, mas requerem um bom entendimento do modelo para evitar degradações.

Em síntese, a discussão revela que não existe um banco de dados universalmente superior, mas sim soluções mais adequadas a contextos específicos. A decisão entre SQL, NoSQL ou grafos deve considerar não apenas o volume e tipo de dados, mas também a linguagem utilizada, os requisitos de desempenho, a natureza das consultas e o perfil do sistema. Assim, profissionais de TI devem adotar uma abordagem crítica e contextualizada ao escolher a arquitetura de banco de

dados ideal para seus projetos.

6. CONCLUSÃO

Este artigo comparou o desempenho de diferentes tipos de bancos de dados — relacionais e não relacionais — em uma abordagem multiplataforma, considerando também o impacto de diferentes linguagens de programação. Os resultados indicaram que SGBDs relacionais, como MySQL e PostgreSQL, continuam a oferecer excelente desempenho em consultas estruturadas, especialmente quando aplicados a contextos com regras rígidas de integridade e agregações complexas. Já soluções NoSQL, como MongoDB, e bancos de grafos, como Neo4j, mostraram vantagens significativas em escalabilidade e na manipulação de dados não tabulares ou fortemente interconectados.

A análise também evidenciou que o desempenho pode variar significativamente de acordo com a linguagem utilizada para acessar os dados. A linguagem C apresentou vantagens em operações intensivas de escrita, enquanto Python mostrou-se eficiente em consultas simples, com melhor produtividade no desenvolvimento. Essa variação reforça a importância de considerar o ambiente de execução completo — banco, linguagem e drivers — ao planejar sistemas que exigem alta performance.

Entre as limitações do estudo, destaca-se o uso exclusivo de dados secundários, o que impede a padronização completa dos testes. Além disso, aspectos como complexidade de manutenção, curva de aprendizado e custo operacional não foram contemplados nos testes de desempenho, mas devem ser considerados em decisões práticas. Também não foi possível explorar a performance de bancos em nuvem ou soluções híbridas.

Para futuras pesquisas, recomenda-se a realização de estudos empíricos com coleta de dados primários, explorando também aspectos de segurança, escalabilidade horizontal e custo-benefício. Investigações focadas em bancos de dados em ambientes de big data, edge computing ou com suporte a inteligência artificial também se mostram promissoras, dada a crescente diversidade de aplicações que demandam arquiteturas de dados específicas e otimizadas.

REFERÊNCIAS

- ALOMARI, E.; ALSALAMAH, M.; ALSALEH, M. Performance Evaluation of Programming Languages. *International Journal of Computer Applications*, v. 112, n. 15, p. 1–7, 2015.
- BARDIN, L. *Análise de conteúdo*. Lisboa: Edições 70, 2011.
- FERREIRA, E. R. *Análise de desempenho de bancos de dados*. Barbacena: UNIPAC, 2013.
- GIL, A. C. *Como elaborar projetos de pesquisa*. 6. ed. São Paulo: Atlas, 2008.
- GUBA, E. G.; LINCOLN, Y. S. Competing paradigms in qualitative research. In: DENZIN, N. K.; LINCOLN, Y. S. (Orgs.). *Handbook of qualitative research*. Thousand Oaks: Sage, 1994.
- HAN, J.; KAMBER, M.; PEI, J. *Data mining: concepts and techniques*. 3. ed. San Francisco: Morgan Kaufmann, 2011.
- MARCONI, M. A.; LAKATOS, E. M. *Fundamentos de metodologia científica*. 6. ed. São Paulo: Atlas, 2003.
- PATTON, M. Q. *Qualitative research and evaluation methods*. 3. ed. Thousand Oaks: Sage, 2002.
- REINSEL, D.; GANTZ, J.; RYDER, J. *Data Age 2025: The Evolution of Data to Life-Critical*. IDC White Paper, 2017.
- ROBINSON, I.; WEBBER, J.; EIFREM, E. *Graph Databases*. Sebastopol: O'Reilly Media, 2015.
- RODRIGUES, E.; TRAD JÚNIOR, S. M. Análise de desempenho de sistemas gerenciadores de banco de dados. In: *Congresso Nacional de Computação*, 2013.
- SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. *Sistemas de Banco de Dados*. 6. ed. São Paulo: Pearson Addison Wesley, 2010.
- SILVA, L. T. S. *Uma Revisão Sistemática da Literatura sobre Bancos de Dados Relacionais e Não Relacionais*. UTFPR, 2023.
- ZIMMER, N. R. *Análise Comparativa de Desempenho de um Banco de Dados Relacional em Diferentes Linguagens de Programação*. UFRGS, 2023.