Análise de Desempenho em Banco de Dados NoSQL Orientado a Documentos: Um Índice para Comparação de Modelos de Dados

Harley Vera-Olivera[1,2] **, Edwin Alvarez-Mamani**[3] **, Maristela Holanda**[1,4]

[1]Departamento de Ciências da Computação
Universidade de Brasília (UnB) - Brasília, DF - Brazil

[2]Departamento de Ingeniería Informática
Universidad Nacional de San Antonio Abad del Cusco - Cusco, Perú

[3]Engineering Department
Pontificia Universidad Católica del Perú - Lima, Perú

[4]Department of Computer Science and Engineering
Texas A&M University College Station - TX, U.S.A.

harley.vera@unsaac.edu.pe, edwin.alvarez@pucp.edu.pe, mholanda@unb.br

Abstract. *This study compares two data modeling approaches in document-oriented NoSQL databases: referenced documents and nested documents, to establish an overall index between the access performances of these two approaches. The results, using multiple regression, show that, on average, in replica set mode, access to data in nested documents is 2.20 times faster with cache enabled and 1.74 times faster with cache disabled. In standalone mode, access to data in nested documents is 1.70 times faster with cache enabled and 1.24 times faster with cache disabled.*

Resumo. *Este estudo compara duas abordagens de modelagem de dados em bancos de dados NoSQL orientados a documentos: documentos referenciados e documentos aninhados, com a finalidade de estabelecer um índice geral entre os desempenhos de acesso dessas duas abordagens. Os resultados da análise, usando regressão múltipla, mostram que, em média, no modo replica set, o acesso aos dados em documentos aninhados é 2.20 vezes mais rápido com cache ativado e 1.74 vezes mais rápido com cache desativado. No modo standalone, o acesso aos dados em documentos aninhados é 1.70 vezes mais rápido com cache ativado e 1.24 vezes mais rápido com cache desativado.*

1. Introdução

Bancos de Dados orientados a documentos têm sido utilizados para gerenciamento de dados em diferentes sistemas computacionais [Diogo et al. 2019]. A estrutura base deste tipo de NoSQL é o documento [Corbellini et al. 2017]. Um documento pode conter atributos simples e complexos. Os atributos simples são características individuais do documento que podem ser representadas por um único valor. Enquanto os atributos complexos, também conhecidos como aninhados, englobam outros atributos ou documentos completos dentro de si, sendo geralmente representados como *arrays*. Um documento pode ser

modelado de duas formas principais: por referência, isto é, um documento referencia um outro documento; e por aninhamento, um documento está inserido em um outro documento. Assim, a escolha entre as duas abordagens pode afetar significativamente o desempenho das consultas nestes bancos de dados [Gómez et al. 2020].

Vários estudos com banco de dados baseado em documento têm sido apresentados na literatura, tais como: comparações de desempenho de acesso a documentos referenciados e aninhados [Gómez et al. 2016] [Reis et al. 2018] [Shah et al. 2022]; geração automática de esquemas de modelos em documentos [Gómez et al. 2020][Imam et al. 2020]; migração e transformação de bancos de dados relacionais para bancos de dados NoSQL orientados a documentos [Hamouda and Zainol 2017][Chen et al. 2022]. Porém, ainda não há evidências de estudos para determinar um índice geral que compare o desempenho entre os dois tipos de modelagem, referenciada e aninhada. Isso se deve, em grande parte, à complexidade dos fatores que influenciam a eficiência de consultas, como por exemplo, a estrutura do banco de dados, as diferentes consultas, as características da rede e do hardware utilizados. Portanto, a determinação deste índice depende de uma análise cuidadosa de cada um desses fatores em diferentes configurações e cenários de uso.

Neste contexto, este artigo tem como objetivo definir um "Índice de Desempenho" (InD) que permita verificar quão rápido é o acesso aos documentos aninhados, comparado com o acesso aos documentos referenciados em um banco de dados NoSQL orientado a documentos. A determinação deste *InD* exerce um papel fundamental no contexto de métricas de avaliação de modelos de dados, permitindo identificar qual modelo, seja ele referenciado ou aninhado, é mais adequado para uma determinada consulta. Para esta análise foi criado um banco de dados sintético nos modos distribuído com réplicas (*replica set*) e centralizado (*standalone*) em diferentes infraestruturas computacionais. Mediante consultas ao banco de dados sintético com diferentes quantidades de atributos e número de documentos recuperados foram implementados *datasets* junto com o tempo de resposta nas consultas, a fim de criar modelos de regressão múltipla [Weisburd et al. 2022]. Por fim, usando os modelos de regressão, foram analisados os índices de desempenho em documentos referenciados e aninhados.

A estrutura deste artigo é a seguinte. A Seção 2 mostra os trabalhos relacionados com a proposta apresentada neste artigo. Na Seção 3, é apresentado o processo metodológico que guiou o trabalho. Na Seção 4, os resultados são apresentados com as relações de desempenho entre documentos referenciados e aninhados. A Seção 5 mostra o caso de uso apresentado para nossa proposta de trabalho. Finalmente, as conclusões são apresentadas na Seção 6.

2. Trabalhos Relacionados

A geração automática de esquemas, a conversão de modelos relacionais para não relacionais e os processos de modelagem e migração têm impulsionado o estudo e análise da modelagem de dados em banco de dados NoSQL. Em particular, muitos estudos têm se concentrado em bancos de dados orientados a documentos como é o caso de [Imam et al. 2018] [Chen et al. 2022] e [Erraji et al. 2022].

No entanto, poucos estudos têm se dedicado à análise das relações referenciadas e aninhadas em bancos de dados orientados a documentos, embora a escolha entre esses dois tipos de modelagem possa afetar significativamente o desempenho do

banco de dados. Dentre os estudos existentes, destaca-se o trabalho apresentado por [Shah et al. 2022] que propõe esquemas alternativos usando padrões específicos de estruturação de bancos de dados orientados a documentos e analisa os critérios de armazenamento. Este artigo também compara o desempenho e os requisitos de energia do processamento de consultas com e sem indexação, além das operações de junção em nível de aplicação, e contribui para a otimização de consultas.

No trabalho de [Gómez et al. 2021] é apresentado o projeto SCORUS que utiliza automação para gerar alternativas de estruturas, avaliar métricas estruturais e auxiliar na tomada de decisões. O objetivo é oferecer alternativas de modelos de dados e fornecer métricas estruturais para avaliar a complexidade das estruturas de dados.

O trabalho de [de la Vega et al. 2020] apresenta Mortadelo, um processo de projeto de banco de dados NoSQL orientado a modelos, capaz de gerar implementações automáticas para sistemas NoSQL como Cassandra e MongoDB a partir de um modelo conceitual. Mortadelo é avaliado gerando implementações de banco de dados para vários estudos de caso típicos a partir do mesmo modelo conceitual de dados.

O artigo de [Hewasinghage et al. 2021] aborda o problema da falta de um *framework* padrão para otimização de dados e consultas em banco de dados NoSQL orientado a documentos, resultando em implementações diversas e diretrizes específicas sem considerar os custos das consultas. O objetivo do trabalho foi automatizar o desenho de modelo de dados para banco de dados NoSQL orientado a documentos com base nos custos das consultas, em vez de regras genéricas de desenho. Para isso, foi desenvolvido um modelo de custo genérico para armazenamento e consulta, estimando o uso de memória, introduzindo assim um modelo de custo para consultas de acesso aleatório.

[Kuszera et al. 2020] apresenta um conjunto de métricas baseadas em consultas para avaliar um esquema de documento NoSQL em relação a um conjunto de consultas que representam os padrões de acesso de uma aplicação. Foram usados grafos direcionados acíclicos para representar as consultas e os esquemas de documentos.

Em [Reis et al. 2018] discute-se a flexibilidade do modelo de dados e a importância da escolha do modelo de armazenamento de documentos, visto que essa escolha pode impactar significativamente o tempo de resposta às consultas. O objetivo deste estudo foi investigar o impacto de três modelos de dados - aninhado, referenciado e híbrido - no desempenho das consultas. Os resultados evidenciaram que o modelo de dados referenciado apresentou melhor desempenho nas consultas em relação aos outros modelos.

De maneira semelhante, o estudo apresentado em [Gómez et al. 2016] examina a flexibilidade dos modelos de dados referenciados e aninhados em bancos de dados orientados a documentos e seu impacto tanto no desempenho das consultas quanto na consistência dos dados. Para alcançar esse objetivo, foram conduzidos experimentos com seis modelos de dados diferentes, incluindo modelos referenciados, aninhados e híbridos. Os resultados obtidos na pesquisa indicam que coleções contendo documentos aninhados apresentam melhor desempenho em consultas.

Embora os trabalhos anteriores tenham se dedicado à análise do desempenho de modelos de dados referenciados e aninhados em bancos de dados orientados a documentos, ainda não foi examinada a relação de desempenho entre esses dois tipos de modelos. Essa análise é complexa, já que vários cenários devem ser considerados, como a infraes-

trutura (física e virtual), os modos de configuração (*standalone* ou *replica set*), o tipo de modelo de dados, a influência da memória, entre outros. Desta forma, diferentemente dos artigos apresentados anteriormente, o nosso artigo apresenta um índice de desempenho, o *InD*, para um cenário básico que é descrito nas próximas seções.

3. Metodologia

O processo metodológico aplicado nesta pesquisa é ilustrado na Figura 1 sendo composto por quatro etapas. Na primeira etapa, tem-se a criação dos bancos de dados sintéticos nos modos *standalone* e *replica set* em diferentes equipamentos computacionais. Na segunda etapa, os *datasets* foram gerados a partir de consultas feitas nos bancos de dados. Na terceira etapa, os modelos de regressão múltipla foram gerados a partir dos *datasets*. Finalmente, na quarta etapa, usando os modelos de regressão múltipla foi obtido o *InD* de acesso aos dados em documentos referenciados e aninhados.

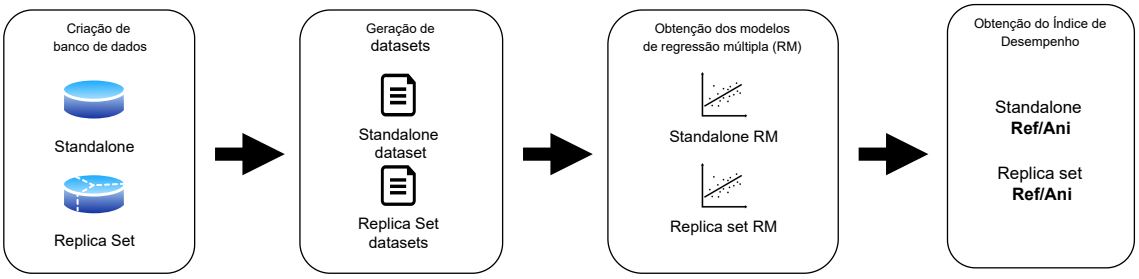


Figura 1. Processo de obtenção do índice de desempenho de acesso entre documentos referenciados e aninhados.

3.1. Ambiente Experimental

Para realizar os experimentos foram considerados três tipos de equipamentos computacionais, tanto físicos como virtuais implementados na nuvem, e dois modos de configuração no MongoDB *standalone* e *replica set*. Os equipamentos computacionais usados e suas características são apresentados na Tabela 1, o equipamento "E1" e os três equipamentos "E2" são equipamentos físicos, enquanto os três equipamentos "E3" são virtuais e implementados na plataforma *Google Cloud*. O sistema de banco de dados usado foi o MongoDB 6.0 com as configurações padrões de instalação. Em particular, as funcionalidades *read concern* e *write concern* foram executadas com suas configurações padrão, sendo "*local*" para "*read concern*" e "*acknowledge*" para "*write concern*". Adicionalmente, o modo de preferência de leitura (*read preference mode*) foi estabelecido com a configuração padrão "*primary*".

Tabela 1. Equipamentos utilizados para implementação dos bancos de dados.

Equipamento	CPU	RAM	Armazenamento	S.O.	Tipo	Quantidade
E1	Intel Xeon 32 núcleos	128 GB	HDD 1,82 TB	Ubuntu 22.04	físico	1
E2	Intel Core i7 16 núcleos	16 GB	HDD 500 GB	Ubuntu 22.04	físico	3
E3	Intel Cascade Lake 2 núcleos	4 GB	HDD 250 GB	Ubuntu 20.04	nuvem	3

A Figura 2 apresenta as configurações *standalone* e *replica set* que foram usadas para realizar os experimentos. No modo de *replica set* três computadores foram usados,

um como nó primário e dois como nós secundários. Para acessar o banco de dados, um computador externo foi utilizado para escrita e leitura de dados nos modos *standalone* e *replica set*.

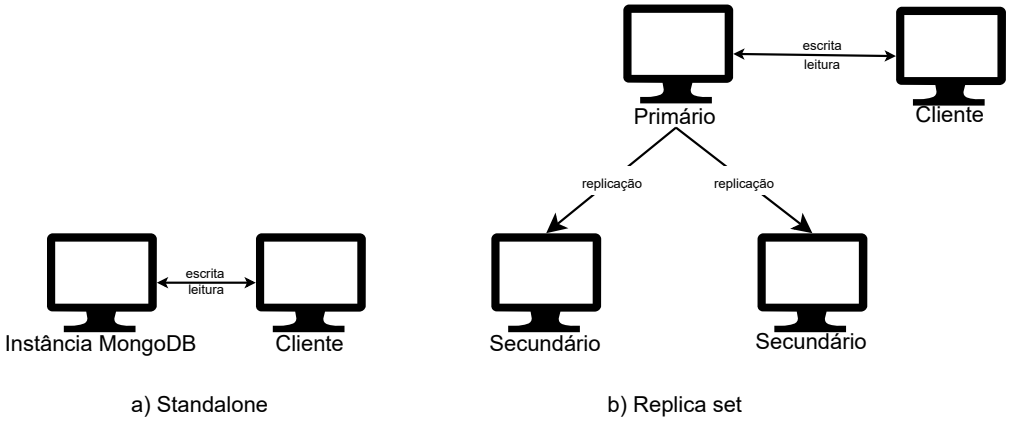


Figura 2. Configurações usadas para os testes.

Finalmente, a Tabela 2 mostra o uso dos equipamentos computacionais para a implementação dos modos *standalone* e *replica set*. Para a implementação do modo *replica set* no equipamento "E1" foi usado o *Docker* 23.0 para virtualizar os três equipamentos necessários.

Tabela 2. Implementação dos modos *standalone* e *replica set* nos equipamentos computacionais.

Equipamento	Standalone	Replica Set
E1	X	X
E2		X
E3		X

3.2. Implementação do Bancos de Dados

Com a finalidade de avaliar as consultas em documentos referenciados e aninhados, foi criado um banco de dados sintético. O banco de dados é composto de dois modelos mostrados nas Figuras 3 e 4, onde cada documento tem 50 atributos simples. Cada atributo é do tipo *string* e gerado aleatoriamente com um comprimento entre 10 e 50 caracteres. A Figura 3 mostra o primeiro modelo com o documento "A" relacionando-se com o documento "B" de forma referenciada com uma cardinalidade um a um, enquanto a Figura 4 mostra o segundo modelo com o documento "A" relacionando-se com o documento "B" de forma aninhada com uma cardinalidade também de um a um. Finalmente, ambos modelos foram implementados no MongoDB 6.0 e preenchidos com 10 milhões de documentos.

3.3. Datasets

Com base no banco de dados sintético e considerando a Figura 5, foram gerados 16 *datasets* para os modos *standalone* ou *replica set* implementados nos equipamentos computacionais descritos na Tabela 1. Os *datasets* foram criados para armazenar características

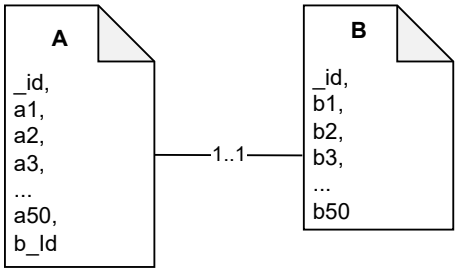


Figura 3. Documentos A e B relacionados por referência.

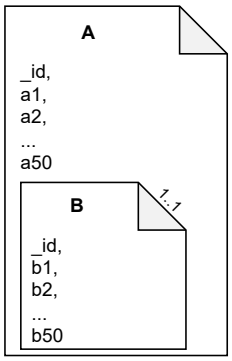


Figura 4. Documentos A e B relacionados por aninhamento.

e tempos de processamento de consultas nos 16 cenários de avaliação. Uma consulta Q_1 foi preparada para recuperar a mesma quantidade de atributos dos documentos A e B. Para o modelo referenciado, a consulta foi preparada utilizando o método *agregate* com as etapas *lookup*, *project* e *limit* do MongoDB. Em contraste, para o modelo aninhado, a consulta foi formulada empregando apenas as etapas de *project* e *limit*. Dessa forma, a consulta Q_1 foi executada nos dois modelos (*standalone* e *replica set*), em cada um dos equipamentos como se mostra na Figura 5. As consultas foram executadas com e sem a influência da memoria cache no MongoDB.

O processo de execução das consultas inicia recuperando $200\,000$ documentos com 5 atributos por documento e repetido 10 vezes, em seguida são recuperados $200\,000$ documentos com 10 atributos por documento e executada 10 vezes novamente assim até chegar aos $200\,000$ documentos com 50 atributos do documento. O processo é repetido recuperando $400\,000$ documentos até chegar aos 5 milhões de documentos recuperados. A cada execução das consultas os tempos de recuperação foram registrados.

A estrutura do *dataset* é composta de três colunas (Tabela 3). A primeira coluna representa o número de documentos requeridos numa consulta, a segunda coluna, o número de atributos por documento requerido e a terceira coluna, o tempo de resposta do banco de dados para a consulta. Assim, a interpretação dos valores da primeira linha na Tabela 3 é que foi feita uma consulta em $200\,000$ documentos, recuperando cinco atributos, com o tempo de consulta t_1. O motivo da criação do dataset com a quantidade de documentos recuperados e o número de atributos por documentos foi analisar a relação com o tempo de resposta.

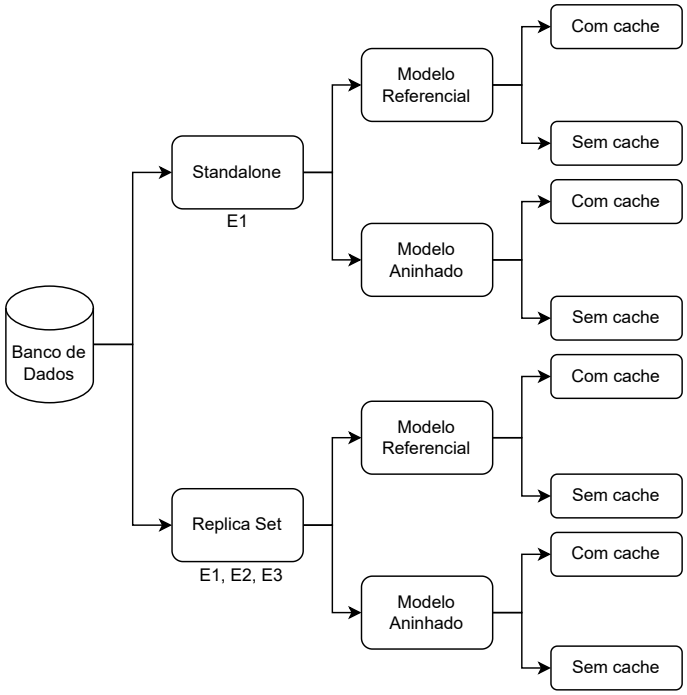


Figura 5. Processo de geração de *datasets* e modelos de regressão múltipla nos modos *standalone replica set*.

3.4. Modelos de Regressão Múltipla

Modelos de regressão múltipla demonstram utilidade na predição, especialmente quando os dados apresentam relações lineares e a quantidade disponível do *dataset* é limitada. Nesse sentido, esses modelos podem prever o comportamento entre variáveis tais como a quantidade de atributos, o número de documentos e o tempo de resposta de consultas.

Seguindo o mesmo processo mostrado na Figura 5, foram gerados 16 modelos de regressão múltipla para os modos *standalone* e *replica set* nos distintos equipamentos computacionais definidos na Tabela 1, levando em conta a influência da memória do MongoDB. Os modelos de regressão múltipla foram treinados com os *datasets* descritos na seção anterior. O objetivo foi realizar uma análise do *InD* entre a quantidade de documentos e atributos retornados a partir de uma consulta, com o tempo de resposta a esta consulta. As variáveis independentes para esta análise foram: o número de documentos a recuperar e o número de atributos que compõem o documento recuperado. A variável dependente considerada foi o tempo de resposta do banco de dados para uma determinada consulta.

A Tabela 4 mostra os coeficientes e intercepto dos modelos gerados para o modelo referenciado e aninhado com e sem a influencia do cache no ambiente "E1" para o modo *standalone* junto com os R^2score do treinamento e teste. Da mesma forma, as Tabelas 5, 6, 7 mostram os modelos de regressão múltipla gerados para o modo *replica set* nos ambientes "E1", "E3", e "E2". A ferramenta utilizada para a obtenção dos modelos foi *Scikit-learn*, com os parâmetros *fit_intercept*, *copy_X*, *n_jobs*, e *positive* todos com as configurações padrão.

Tabela 3. Estrutura do *dataset* criado nos modos *standalone* e *replica set*.

Número Documentos	Número atributos	Tempo resposta
200 000	5	t_1
200 000	10	t_{11}
...	...	...
200 000	50	t_{91}
400 000	5	t_{101}
400 000	10	t_{111}
...	...	...
400 000	50	t_{191}
...	...	...
5 000 000	5	t_{2401}
5 000 000	10	t_{2411}
...	...	...
5 000 000	50	t_{2500}

Tabela 4. Modelos de regressão múltipla obtidos no E1 para o modo *Standalone*.

		Coeficiente 1	Coeficiente 2	Intercepto	R^2score Treinamento	R^2score Teste
Modelo Referenciado	Cache	1.02472334	0.21763937	- 3.017720709513224	0.99	0.99
	Sem cache	0.99698879	0.14871953	- 2.551136584402331	0.99	0.99
Modelo Aninhado	Cache	1.04980047	- 0.00863899	- 3.1112693442386834	0.97	0.97
	Sem cache	1.0035651	- 0.01104212	- 2.477332503137232	0.99	0.99

3.5. Índice de Desempenho *InD* de Acesso

Considerando que os modelos de regressão obtidos na seção anterior generalizam o comportamento entre as variáveis, a saber: quantidade de atributos, número de documentos recuperados e tempo de resposta da consulta, torna-se possível utilizar esses modelos para prever os tempos de resposta para diferentes configurações de quantidade de atributos e número de documentos requeridos em uma consulta. Dessa forma, empregamos os modelos de regressão múltipla (mRM) para determinar os InD entre documentos referenciados e aninhados nos diferentes cenários estudados.

Por exemplo, para o modo *replica set* implementado no ambiente $E1$ foram usados o mRM gerado para o modelo referenciado (R) e aninhado (A) com a influência da cache. Para calcular mRM_R, o modelo foi alimentado com uma quantidade aleatória de atributos (entre 1 a 50 atributos) e quantidade aleatória de documentos (entre 1 a 10 milhões), as mesmas quantidades aleatórias foram usadas para o mRM_A. O processo foi executado 1 000, 10 000, 100 000 e 1 000 000 vezes para gerar a média de mRM_R e mRM_A. Finalmente, foi usada a Equação (1) para determinar o InD de tempos de acesso entre os modelos referenciados e aninhados.

Tabela 5. Modelos de regressão múltipla obtidos no E1 para *replica set*.

		Coeficiente 1	Coeficiente 2	Intercepto	R^2score Treinamento	R^2score Teste
Modelo Referenciado	Cache	1.02265971	0.19742946	-2.9585879035633367	0.97	0.97
	Sem cache	1.029068	0.20535015	-3.017058594187399	0.99	0.99
Modelo Aninhado	Cache	1.05163906	-0.01738922	-3.130097427002466	0.97	0.97
	Sem cache	1.01645005	-0.01997622	-2.9198769723630726	0.98	0.98

Tabela 6. Modelos de regressão múltipla obtidos no E3 para *replica set*.

		Coeficiente 1	Coeficiente 2	Intercepto	R^2score Treinamento	R^2score Teste
Modelo Referenciado	Cache	2.51495792	0.07972045	-11.301507836259296	0.84	0.84
	Sem cache	1.89948341	0.0667366	-7.50971984432025	0.74	0.73
Modelo Aninhado	Cache	2.45338698	-0.03287923	-11.112852953364289	0.82	0.83
	Sem cache	1.72818948	-0.01729218	-6.564202672819475	0.67	0.67

$$InD = \frac{mRM_R}{mRM_A} \quad \text{, onde:} \tag{1}$$

mRM_R e mRM_A média do processo de execução dos modelos regressionais.

4. Resultados

A Tabela 8 mostra os resultados de relações dos InD em documentos aninhados e referenciados nos equipamentos E1, E2 e E3 nos modos *standalone* e *replica set*. Os resultados mostram quão rápido é o acesso aos documentos aninhados comparado com o acesso aos documentos referenciados. Por exemplo, no modo *replica set* no equipamento E2 com a memória cache ativada o acesso aos documentos aninhados é 2.18 vezes mais rápido que o acesso aos documentos referenciados e quando a memoria cache é desativada o acesso

Tabela 7. Modelos de regressão múltipla obtidos no E2 para *replica set*.

		Coeficiente 1	Coeficiente 2	Intercepto	R^2score Treinamento	R^2score Teste
Modelo Referenciado	Cache	1.04713869	0.20608861	-3.1017870609811835	0.97	0.97
	Sem cache	1.02776791	0.19461247	-2.978403725711111	0.98	0.98
Modelo Aninhado	Cache	1.01856333	-0.03465397	-2.9172136708137346	0.97	0.97
	Sem cache	1.0129133	-0.00998741	-2.903627613427707	0.97	0.96

aos documentos aninhados é de 1.30 vezes mais rápido. Da mesma forma, nos equipamentos E3 e E1 no modo *replica set* o acesso a documentos aninhado é 2.33 e 2.09 vezes mais rápido respectivamente comparado ao acesso a documentos referenciados e quando a memoria cache é desativada o InD cai a 1.97 e 1.96. No caso do modo *standalone*, com o cache ativado, o acesso aos documentos aninhados é de 1.70 vezes mais rápido em relação ao acesso aos documentos referenciados e 1.24 com a memoria cache desativada.

Tabela 8. Valores dos InD em documentos referenciados e aninhados.

	Replica set			Standalone
	E2	E3	E1	E1
Com cache	2.18	2.33	2.09	1.70
Sem cache	1.30	1.97	1.96	1.24

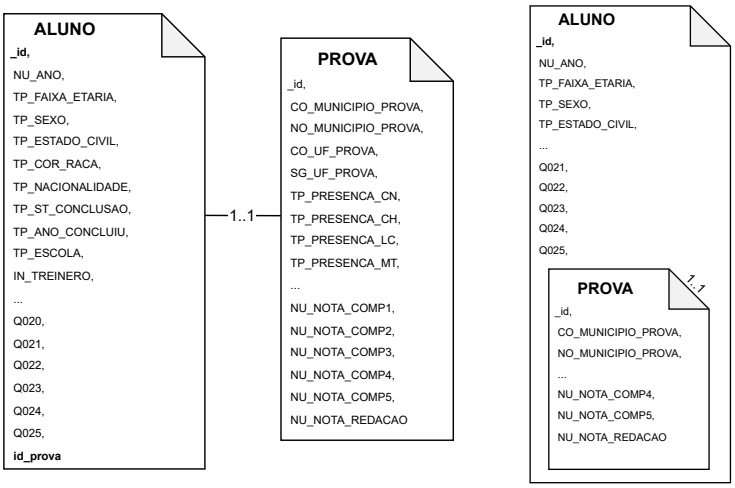
Embora a infraestrutura dos equipamentos seja diferente nos três equipamentos (físicos e virtuais) os valores do *InD* no caso do modo *replica set* com cache sempre foram superiores aos valores sem cache. Porém, a diferença de velocidade de acesso com e sem cache no modo *replica set* no equipamento E1 é mínima. Assim, no modo *replica set*, em média o acesso aos dados em documentos aninhados é de 2.20 vezes mais rápido com o cache ativado e 1.74 com o cache desativado. É importante esclarecer que estes valores foram obtidos para os tipos de modelos e suas características definidas na Seção 3.2.

5. Caso de Uso

Para validar os resultados obtidos com os dados sintéticos, utilizamos o *dataset* do ENEM (Exame Nacional do Ensino Médio) do ano 2021 como caso de uso. O ENEM é uma avaliação educacional aplicada anualmente pelo INEP, vinculado ao MEC, que tem como objetivo avaliar o desempenho dos estudantes do ensino médio em diferentes áreas do conhecimento e verificar a proficiência em redação. O *dataset* possui $3\,389\,832$ linhas e 76 colunas de dados em formato numérico (40 colunas) e alfanumérico (36 colunas). Os dados contidos no *dataset* são relativos a: dados da escola, dados do questionário socioeconômico e dados da prova.

Para avaliar os InD entre documentos referenciados e aninhados no caso de uso do ENEM foram gerados dois modelos de dados como mostra a Figura 6. No primeiro modelo (Figura 6-a) o documento "ALUNO" referencia o documento "PROVA" com uma cardinalidade de um a um. No segundo modelo (Figura 6-b) o documento "ALUNO" contém aninhado o documento "PROVA" com uma cardinalidade de um a um. A modelagem do caso de uso foi realizada considerando os modelos de dados do banco de dados sintético apresentados na Seção 3.2, com os quais foram treinados os modelos de regressão múltipla. Os dois modelos foram implementados no MongoDB e populados usando o *dataset* do ENEM.

Foi definida uma consulta (Q_2) para avaliar a relação de desempenho de acesso entre os modelos de dados referenciado e aninhado no contexto do caso de uso. Considerando o objetivo de avaliar a relação de desempenho de acesso em conformidade com o treinamento do modelo regressão múltipla, foram consideradas as seguintes premissas ao definir a consulta: (1) a equidade do número de atributos acessados nos dois documentos



(a) Modelo referenciado

(b) Modelo aninhado

Figura 6. Modelos de dados ENEM.

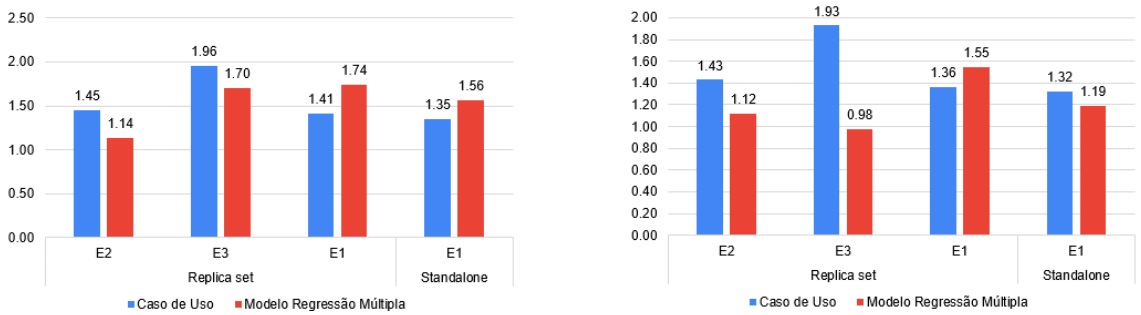
(aluno e prova); (2) a natureza alfanumérica dos atributos a serem acessados; (3) a igualdade do número total de documentos extraídos nos dois modelos de dados. Neste teste, a consulta Q_2 foi definida para recuperar 10 atributos do documento ALUNO (TP_SEXO, SG_UF_ESC e os atributos Q001 a Q009) e 10 atributos do documento PROVA (CO_UF_PROVA, SG_UF_PROVA, TX_RESPOSTAS_CN, TX_RESPOSTAS_CH, TX_RESPOSTAS_LC, TX_RESPOSTAS_MT, TX_GABARITO_CN, TX_GABARITO_CH, TX_GABARITO_LC e TX_GABARITO_MT). A escolha dos atributos nas coleções ALUNO e PROVA é fundamentada no fato de que eles são do tipo alfanumérico. Para o modelo referenciado, a consulta foi preparada utilizando o método *agregate* com as etapas *lookup*, *project* e *limit* do MongoDB. Enquanto para o modelo aninhado, foram utilizadas apenas as etapas *project* e *limit*.

A Figura 7 apresenta os valores InD obtidos após a aplicação da consulta Q_2 no modelo de regressão múltipla e no banco de dados do caso de uso, nos modos *replica set* e *standalone* nos equipamentos E1, E2 e E3, com e sem a influência do cache. Por exemplo, na Figura 7-a do caso de uso, podemos observar que no equipamento E2, no modo *replica set*, a recuperação de dados em documentos aninhados é 1.45 vezes mais rápida do que a recuperação de dados em documentos referenciados com o cache ativado. Já na Figura 7-b, também no caso de uso, podemos constatar que no equipamento E2, no modo *replica set*, a recuperação de dados em documentos aninhados é 1.43 vezes mais rápida do que a recuperação de dados em documentos referenciados com o cache desativado.

Da Figura 7 a) e b) podemos perceber que os valores InD sempre foram superiores, tanto no caso de uso quanto no modelo de regressão, com o cache ativado. Adicionalmente, os valores InD obtidos no modelo de regressão e no banco de dados do caso de uso para a consulta Q_2 foram similares.

6. Conclusões

Neste estudo, foi analisado o InD de acesso a dados entre documentos referenciados e aninhados em banco de dados orientado a documentos. Os resultados indicam o acesso



a) Com cache b) Sem cache

Figura 7. Valores InD de desempenho de acesso no banco de dados do caso de uso e no modelo de regressão múltipla

mais rápido a documentos aninhados do que a referenciados. Adicionalmente, os resultados mostraram que acessar a documentos aninhados é em média 2.20 vezes mais rápido com o cache ativado e 1.74 vezes com o cache desativado no modo *replica set*. No modo *standalone*, acessar documentos aninhados é 1.70 vezes mais rápido com o cache ativado e 1.24 com o cache desativado.

É importante ressaltar que os resultados deste artigo são iniciais, sendo válidos para o modelo de dados e o banco de dados específico que foram utilizados neste estudo. A análise foi realizada em um ambiente controlado e os resultados podem variar em diferentes modelos de dados, tipo de dados e bancos de dados. É necessário levar em consideração as características específicas de cada modelo e banco de dados ao aplicar esses resultados em outros contextos. Nesse sentido, é necessário mais casos de usos e experimentos para validar os resultados apresentados neste trabalho, uma vez que o caso de uso do ENEM não possui as mesmas características de tipo de dados e longitudes de atributos do banco de dados sintético criado.

O presente estudo explorou o InD entre documentos referenciados e aninhados em um cenário básico. No entanto, como trabalhos futuros, é necessário investigar as cardinalidades $1..N$ e $N..M$, assim como outros tipos de dados e operações no banco de dados, como escrita, atualização e remoção. Além disso, devem ser consideradas infraestruturas adicionais, como os *shard clusters*. Essas áreas representam oportunidades para pesquisas futuras e podem fornecer uma compreensão mais completa dos InD em diferentes cenários. Adicionalmente, uma análise dos fatores de hardware e configurações de software que podem influenciar nos experimentos é necessária em todos os cenários e equipamentos computacionais utilizados.

Referências

[Chen et al. 2022] Chen, L., Davoudian, A., and Liu, M. (2022). A workload-driven method for designing aggregate-oriented nosql databases. *Data & Knowledge Engineering*, 142:102089.

[Corbellini et al. 2017] Corbellini, A., Mateos, C., Zunino, A., Godoy, D., and Schiaffino, S. (2017). Persisting big-data: The nosql landscape. *Information Systems*, 63:1–23.

[de la Vega et al. 2020] de la Vega, A., García-Saiz, D., Blanco, C., Zorrilla, M., and Sánchez, P. (2020). Mortadelo: Automatic generation of nosql stores from platform-

independent data models. *Future Generation Computer Systems*, 105:455–474.

[Diogo et al. 2019] Diogo, M., Cabral, B., and Bernardino, J. (2019). Consistency models of nosql databases. *Future Internet*, 11(2):43.

[Erraji et al. 2022] Erraji, A., Maizate, A., and Ouzzif, M. (2022). Toward a smart approach of migration from relational system database to nosql system: Transformation rules of structure. In *Innovations in Smart Cities Applications Volume 5: The Proceedings of the 6th International Conference on Smart City Applications*, pages 783–794. Springer.

[Gómez et al. 2016] Gómez, P., Casallas, R., and Roncancio, C. (2016). Data schema does matter, even in nosql systems! In *2016 IEEE Tenth International Conference on Research Challenges in Information Science (RCIS)*, pages 1–6. IEEE.

[Gómez et al. 2020] Gómez, P., Casallas, R., and Roncancio, C. (2020). Automatic schema generation for document-oriented systems. In *Database and Expert Systems Applications: 31st International Conference, DEXA 2020, Bratislava, Slovakia, September 14–17, 2020, Proceedings, Part I 31*, pages 152–163. Springer.

[Gómez et al. 2021] Gómez, P., Roncancio, C., and Casallas, R. (2021). Analysis and evaluation of document-oriented structures. *Data & Knowledge Engineering*, 134:101893.

[Hamouda and Zainol 2017] Hamouda, S. and Zainol, Z. (2017). Document-oriented data schema for relational database migration to nosql. In *2017 International conference on big data innovations and applications (innovate-data)*, pages 43–50. IEEE.

[Hewasinghage et al. 2021] Hewasinghage, M., Abelló, A., Varga, J., and Zimányi, E. (2021). A cost model for random access queries in document stores. *The VLDB Journal*, 30(4):559–578.

[Imam et al. 2020] Imam, A. A., Basri, S., Ahmad, R., Wahab, A. A., González-Aparicio, M. T., Capretz, L. F., Alazzawi, A. K., and Balogun, A. O. (2020). Dsp: Schema design for non-relational applications. *Symmetry*, 12(11):1799.

[Imam et al. 2018] Imam, A. A., Basri, S., Ahmad, R., Watada, J., and González-Aparicio, M. T. (2018). Automatic schema suggestion model for nosql document-stores databases. *Journal of Big Data*, 5(1):1–17.

[Kuszera et al. 2020] Kuszera, E. M., Peres, L. M., and Didonet Del Fabro, M. (2020). Query-based metrics for evaluating and comparing document schemas. In *Advanced Information Systems Engineering: 32nd International Conference, CAiSE 2020, Grenoble, France, June 8–12, 2020, Proceedings 32*, pages 530–545. Springer.

[Reis et al. 2018] Reis, D. G., Gasparoni, F. S., Holanda, M., Victorino, M., Ladeira, M., and Ribeiro, E. O. (2018). An evaluation of data model for nosql document-based databases. In *Trends and Advances in Information Systems and Technologies: Volume 1 6*, pages 616–625. Springer.

[Shah et al. 2022] Shah, M., Kothari, A., and Patel, S. (2022). Influence of schema design in nosql document stores. In *Mobile Computing and Sustainable Informatics: Proceedings of ICMCSI 2021*, pages 435–452. Springer.

[Weisburd et al. 2022] Weisburd, D., Wilson, D. B., Wooditch, A., Britt, C., Weisburd, D., Wilson, D. B., Wooditch, A., and Britt, C. (2022). Multiple regression. *Advanced Statistics in Criminology and Criminal Justice*, pages 15–72.