
FERRAMENTA PARA ANÁLISE DE DESEMPENHO DE BANCOS DE DADOS NÃO RELACIONAIS

NASCIMENTO, Anderson Prado¹
OLIVEIRA, José Gabriel Alves²
JÚNIOR, Flávio Rubens Massaro³

Centro Universitário Hermínio Ometto – FHO, Araras – SP, Brasil

Resumo

Os sistemas de gerenciamento de banco de dados não relacionais, conhecidos como 'Not Only SQL'(NoSQL) surgiram para abordar a complexidade de lidar com enormes volumes de dados, mais especificamente não estruturados e semiestruturados, oferecendo uma flexibilidade maior em comparação com os modelos de banco de dados relacionais. Apesar da vantagem dos bancos de dados NoSQL conseguirem lidar com volumes de dados não estruturados e semiestruturados, os mesmos possuem uma dificuldade com a integração com outras tecnologias e também a falta de padronização e organização dos dados em relação aos bancos de dados relacionais. O projeto em questão consiste em desenvolver uma ferramenta de benchmarking que suporte modelos de bancos de dados não relacionais, realizando uma comparação de desempenho quantitativa e qualitativa. Serão utilizadas duas máquinas virtuais utilizando o software VirtualBox para implementação da ferramenta, utilizando o software Visual Studio Code e a linguagem de programação Python para seu desenvolvimento. É viável como resultado deste projeto que a ferramenta apresente uma visão ampla entre o desempenho dos bancos de dados NoSQL por meio de gráficos, facilitando a visualização e análise dos resultados obtidos.

Palavras chave: NoSQL, Ferramenta, Benchmarking, Dados, Semiestruturados, Não estruturados

1 Introdução

1.1 Contextualização

A necessidade de armazenamento e disponibilidade de informações tem se tornado uma preocupação relevante no mercado atual, devido à alta demanda e ao volume de processamento de dados, é essencial contar com um sistema capaz de gerenciar e suportar todas as informações de maneira adequada.

Os métodos de organização e estruturação dos dados em um sistema de gerenciamento de banco de dados são denominados modelos de bancos de dados. O modelo abordado neste trabalho será o modelo não relacional, que é projetado para lidar com uma grande massa de dados não estruturados e semiestruturados. Com base neste modelo, iremos desenvolver uma ferramenta de *benchmarking* com o intuito de abordar o desempenho

¹ FHO|UNIARARAS. Aluno do Curso de Sistemas de Informação, NASCIMENTO, andersonnascimento@alunos.uniararas.br

² FHO|UNIARARAS. Aluno do Curso de Sistemas de Informação, OLIVEIRA, jgararas@alunos.uniararas.br

³ FHO|UNIARARAS. Professor do Curso de Sistemas de Informação, JÚNIOR, frmassaro@uniararas.br

entre os bancos de dados *NoSQL* e apresentar os dados em formato de gráficos, facilitando a análise dos resultados obtidos.

1.2 Tema de Pesquisa

Ferramenta para comparação de desempenho entre bancos de dados *NoSQL*, projetada para auxiliar os usuários na identificação da solução mais adequada para suas necessidades.

1.3 Motivações e Justificativas

Um sistema adequado de gerenciamento de banco de dados é fundamental para o desempenho e processamento dos dados, suprimindo os requisitos de forma adequada dependendo de cada aplicação. A ferramenta desenvolvida neste trabalho oferecerá uma análise do desempenho dos dados adquiridos, com base nos sistemas de gerenciamento *NoSQL*. Esta análise abrangerá diversos aspectos, incluindo os tipos de dados utilizados e os métodos de manipulação de dados (*DML*).

1.4 Objetivos

1.4.1 Objetivo Geral

Desenvolver uma ferramenta de *benchmarking* para realizar análises de desempenho utilizando bancos de dados *NoSQL*.

1.4.2 Objetivos Específicos

- Analisar e selecionar os bancos de dados para o teste de *benchmarking*;
- Avaliar o desempenho dos bancos de dados não relacionais selecionados;
- Apresentar os resultados obtidos em cada teste realizado;
- Gerar relatórios através de gráficos.

2 Revisão Bibliográfica

2.1 Fundamentação Teórica e Técnica

Um Sistema de Gerenciamento de Banco de Dados (SGBD) é um conjunto de *software* responsável por definir, construir e manipular a estrutura de um banco de dados e gerenciar o acesso aos dados armazenados de forma eficiente e segura (CÔRTEZ, 2001, p.5).

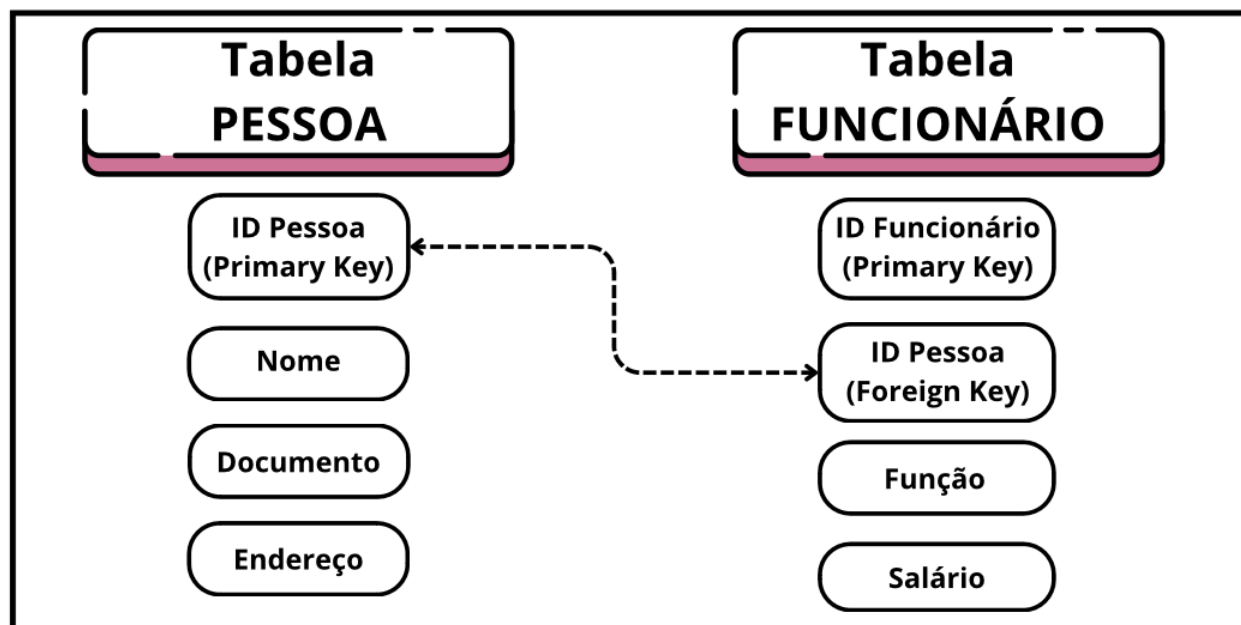
Um SGBD fornece uma série de operações ou uma linguagem chamada Linguagem de Manipulação de Dados (*DML*). Essa linguagem é utilizada para manipular dados dentro de um banco de dados, incluindo inserção, atualização, exclusão e consulta de dados (*INSERT*, *DELETE*, *UPDATE*). Também existem outras linguagens utilizadas por um SGBD, como a linguagem de manipulação e definição dos dados (*DDL*), a linguagem de consulta dos dados (*DQL*) e também a linguagem de controle de dados (*DCL*) (Elmasri; Navathe, 2005, p.24).

Referente ao processo de gerenciamento dos dados em um SGBD, o mesmo envolve definir mecanismos da linguagem *DML* para manipular essas informações. Além disso, um sistema de banco de dados deve garantir a segurança das informações armazenadas, impedindo o vazamento de informações, garantindo a proteção dos dados pessoais e assegurando que as empresas e organizações possuam uma tratativa adequada dos dados, conforme A Lei Geral de Proteção dos Dados - LGPD (Lei nº 13.709/2018).

2.1.1 Modelo Relacional

Os modelos relacionais em bancos de dados são fundamentados pela sua organização dos dados em tabelas, estruturadas em linhas e colunas. Cada tabela em um banco de dados relacional possui seus devidos atributos, sendo os mais importantes para estabelecer os relacionamentos entre as chaves primárias e as chaves estrangeiras. A chave primária é o identificador de cada tabela, não permitindo repetição na mesma tabela, geralmente referenciado como “*ID*”. A chave estrangeira, é o identificador que tem como objetivo manter o relacionamento entre as devidas tabelas. A utilização de chaves primárias e estrangeiras facilita a organização e a integridade dos dados dentro do banco de dados, permitindo consultas eficientes e relacionamentos consistentes entre diferentes tabelas. Neste exemplo, visualiza-se que a tabela funcionário, possui um relacionamento com a tabela pessoa por possuir a chave estrangeira (*Foreign Key*) “*ID Pessoa*” (NAVATHE, 2005, p.89-94).

Figura 1 - Modelo Relacional



Fonte: Elaborado pelos autores

Atualmente, existem vários Sistemas de Gerenciamento de Banco de Dados Relacionais (SGBDR) disponíveis no mercado. Na figura 2, é apresentado um *ranking* dos Sistemas de Gerenciamento de Banco de Dados (SGBD) mais utilizados atualmente, conforme evidenciado pelo site *db-engines* em março de 2024.

Figura 2 - Tabela *Ranking* BD's

418 systems in ranking, March 2024

Rank			DBMS	Database Model	Score		
Mar 2024	Feb 2024	Mar 2023			Mar 2024	Feb 2024	Mar 2023
1.	1.	1.	Oracle +	Relational, Multi-model ⓘ	1221.06	-20.39	-40.23
2.	2.	2.	MySQL +	Relational, Multi-model ⓘ	1101.50	-5.17	-81.29
3.	3.	3.	Microsoft SQL Server +	Relational, Multi-model ⓘ	845.81	-7.76	-76.20
4.	4.	4.	PostgreSQL +	Relational, Multi-model ⓘ	634.91	+5.50	+21.08
5.	5.	5.	MongoDB +	Document, Multi-model ⓘ	424.53	+4.18	-34.25
6.	6.	6.	Redis +	Key-value, Multi-model ⓘ	157.00	-3.71	-15.45
7.	7.	↑ 8.	Elasticsearch	Search engine, Multi-model ⓘ	134.79	-0.95	-4.28
8.	8.	↓ 7.	IBM Db2	Relational, Multi-model ⓘ	127.75	-4.47	-15.17
9.	9.	↑ 11.	Snowflake +	Relational	125.38	-2.07	+10.98
10.	10.	↓ 9.	SQLite +	Relational	118.16	+0.88	-15.66

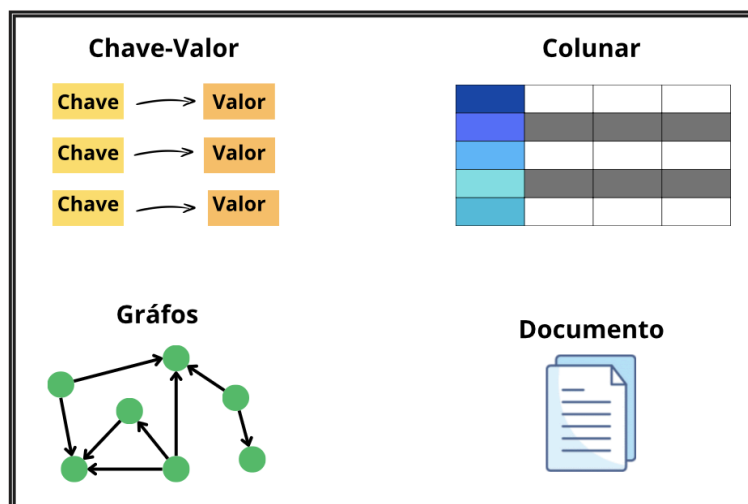
Fonte:db-engines, 2024

A partir deste *ranking*, observamos que entre os dez bancos de dados abordados, sete são relacionais, destacando sua importância e predominância no mercado atual.

2.1.2 Modelo Não Relacional

O modelo não relacional surgiu para suprir a necessidade de lidar com um grande volume de dados não estruturados e semiestruturados, destacando-se por uma alta escalabilidade e pela sua capacidade de lidar com diferentes tipos de dados. O termo *NoSQL* (*Not Only SQL*) foi introduzido primeiramente em 1998 pelo seu autor Carlo Strozzi, após um período de onze anos, o nome *NoSQL* foi reintroduzido em 2009 pelo Eric Evans em uma palestra sobre banco de dados *open source* e desde então o termo *NoSQL* predominou no mercado (GARCIA; SOTTO, 2019).

2.1.3 Tipos de bancos de dados não relacionais

Figura 3 - Bancos de dados *NoSQL*

Fonte: Elaborado pelos autores

Conforme a documentação da Microsoft, atualizada em 2022, os bancos de dados não relacionais possuem subdivisões e classificações, dependendo da forma que os dados serão armazenados e utilizados, cada um possuindo suas próprias características.

Chave-Valor - O conceito do banco de dados orientado a Chave-Valor é o mais básico, cujo o mesmo se aplica em definir um valor para uma chave, possuindo uma escalabilidade predominante perante as outras subdivisões.

Figura 4 - Modelo NoSQL Chave-Valor

Chave	Valor
Nome	Luis Oliveira
Idade	21
CPF	99999999999
Telefone	(00)000000000

Fonte: Elaborado pelos autores

Colunar - O modelo colunar consiste em armazenar os dados em colunas e linhas, permitindo uma pesquisa específica de um conjunto de dados. O destaque desse modelo é a pesquisa pela “família”, que consiste agrupar os dados em uma única coluna. É possível relacionar este modelo aos streamings, como por exemplo a Netflix, o Prime Video e a Star Plus.

Figura 5 - Modelo NoSQL Colunar

ID	Coluna_Familia: Identidade	ID	Coluna_Familia: Info_Contato
001	Primeiro_Nome: Luis Ultimo_Nome: Oliveira	001	Telefone: (00)000000000
002	Primeiro_Nome: Pedro Ultimo_Nome: Santana Suffixo: Jr.	002	Telefone: (99)999999999 Email: pedrosantana@gmail.com
003	Primeiro_Nome: Wilson Ultimo_Nome: Santos Titulo: Dr.	003	Email: wilsonsantos@gmail.com

Fonte: Elaborado pelos autores

Documentos - Este modelo tem como objetivo armazenar arquivos equivalentes a *JSON* e *XML*, podendo localizar estes documentos por um *ID* ou por outro tipo de registro que o mesmo tenha, os modelos orientados a documentos podem armazenar além de arquivos como *JSON* e *XML*, armazenar textos sem nenhum tipo de formatação, permitindo realizar um filtro por valores armazenados nos campos.

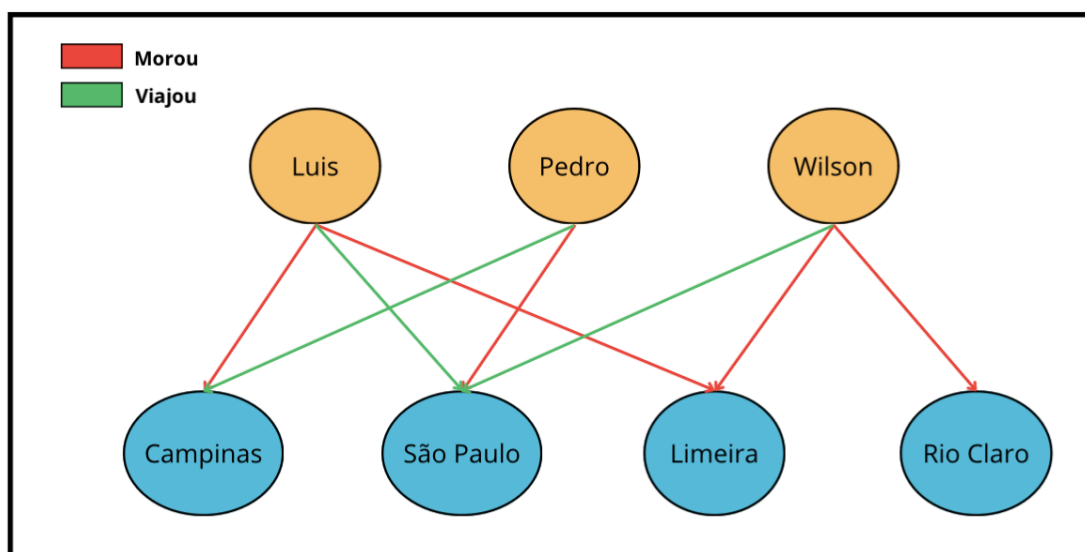
Figura 6 - Modelo NoSQL Documento

Chave	Documento
001	{ "id": 001, "Info_Contato": [{ "Telefone": 00000000000 }] }
002	{ "id": 002, "Info_Contato": [{ "Telefone": 99999999999, "Email": pedrosantana@gmail.com }] }
003	{ "id":003, "Info_Contato": [{ "Email": wilsonsantos@gmail.com }] }

Fonte: Elaborado pelos autores

Grafos - O modelo orientado a grafos consiste em nós e arestas, projetado para armazenar e representar os dados via grafos, destacando-se pela sua eficiência em pesquisas devido às relações de entidades.

Figura 7. Modelo NoSQL Grafo



Fonte: Elaborado pelos autores

2.2.1 - *Benchmarking*

Segundo Silva (2021, p.15), o *benchmarking* originou-se na década de 1970, com a empresa *Xerox Corporation* que passava por uma crise significativa devido sua diminuição na participação de mercado comparado com um de seus concorrentes que oferecia produtos similares a um preço muito mais baixo. Como estratégia de planejamento e busca de melhores resultados, a empresa buscou aplicar melhorias em seus processos para melhorar a qualidade de seus produtos reduzindo os custos de produção para competir com seus concorrentes no mesmo nível de mercado. O presidente da Xerox Charles Christ, decidiu enviar uma equipe ao Japão para estudar e entender as técnicas de *benchmarking* no intuito de aplicar em sua organização em busca de melhores resultados. Essa iniciativa levou à recuperação da Xerox e ao desenvolvimento do programa "*Leadership through Quality*" em 1983. Em 1989, a Xerox recebeu o prêmio "*Malcolm Baldrige Quality Award*", destacando suas práticas de *benchmarking* em nível mundial.

Benchmarking nada mais é que estabelecer um padrão de referência para realizar análises de comparações. Essa ferramenta aplicada de maneira correta com a necessidade, permite que as organizações identifiquem, adaptem e melhorem as práticas em seus processos e produtos, tornando assim uma organização mais competitiva no mercado, independentemente de seu tamanho ou área de atuação, provendo assim melhorias por meio do aprendizado organizacional mútuo. Essa técnica de *benchmarking* é valiosa para identificar lacunas de desempenho, realizar definições de metas de melhoria e promover mudanças necessárias para atingir um nível maior de maturidade.

2.2.2 - Aplicabilidade do processo de *benchmarking*

De acordo com a teoria de Silva (2021, p.17,18), o *benchmarking* é um processo dividido em quatro etapas, sendo elas: planejamento do projeto, coleta de dados, análise de dados e adaptação/melhoria. No planejamento do projeto, é de suma importância identificar e determinar o objeto da análise comparativa a ser realizada. Na coleta de dados, é necessário haver a consistência do ambiente e dos dados que serão inseridos na ferramenta para análise de desempenho. A análise de dados envolve avaliar as diferenças de resultados que mostrará o desempenho e identificar pontos quais os pontos fortes na geração dos resultados. A fase final é a adaptação e implementação da melhoria contínua, onde as conclusões das etapas anteriores são usadas para mudar as operações visando melhorar o desempenho a longo prazo. Esses processos devem ser cuidadosamente trabalhados para garantir resultados satisfatórios para todas as partes envolvidas.

2.2.3 - *Benchmarking* no Ambiente Computacional

Em definição para Silva (2021), o *benchmarking* teve início sendo aplicado primeiramente em empresas e expandiu-se para diversas áreas da ciência, incluindo a área da tecnologia. Na computação, o *benchmarking* é utilizado para análise de desempenho em diversos segmentos, sendo informação geográfica e algoritmos de classificação, análise e desempenho de banco de dados entre outros. Enquanto a técnica de monitoração não possui padronização, o *benchmarking* utiliza métodos padronizados para garantir

comparações precisas. Para uma acuracidade nas análises, o principais objetivos é avaliar capacidade máxima de sistemas, comparar tecnologias, avaliar viabilidade em contextos específicos e mensurar relação custo-benefício. Em ciência da computação, o *benchmarking* é aplicado em diversas subáreas, como sistemas operacionais, computação paralela, mineração de dados e redes de computadores. Por exemplo, em bancos de dados, é comum definir um grupo de instruções para executar em sistemas de avaliação e extrair resultados de desempenho. A versatilidade do *benchmarking* o torna uma ferramenta valiosa para solucionar problemas em diferentes contextos computacionais.

2.2 Trabalhos Relacionados

O trabalho de Godoy, Felipe Pereira de, e Pinheiro, Gabriel Benedito, Análise de desempenho de banco de dados relacional(*MySQL*) e não relacional(*MongoDB*), teve como objetivo realizar uma análise de desempenho entre o banco de dados relacional *MySQL* e o modelo não relacional, orientado a documentos, o *MongoDB*. Os autores aplicaram diferentes volumes de transações e dados simultâneos, através dos métodos *DML* e *DQL* para obterem os resultados da pesquisa. Por meio deste trabalho foi possível analisar o tempo de execução entre os bancos de dados *MySQL* e *MongoDB*, observando como resultado, que o *MongoDB* possui uma vantagem nas operações *INSERT* e *DELETE*, já na operação *SELECT*, tiveram um resultado similar e o banco *MySQL* obteve uma vantagem na operação de *UPDATE*.

O trabalho de PILOTO LOURENÇO, Bárbara Raquel, e JANUÁRIO, Nicolas dos Santos, Análise de desempenho entre banco de dados relacional(*MySQL*) e não relacional(*Amazon DynamoDB*), teve como objetivo realizar uma análise de desempenho entre o banco de dados relacional *MySQL* e o modelo não relacional, orientado a chave-valor, o *Amazon DynamoDB*. Os autores aplicaram diferentes volumes de transações e dados simultâneos, através dos métodos *DML* e *DQL* para obterem os resultados da pesquisa. Por meio deste trabalho foi possível analisar o tempo de execução entre os bancos de dados *MySQL* e *Amazon DynamoDB*, observando como resultado, que o *Amazon DynamoDB* possui uma vantagem nas operações *INSERT* e *DELETE*, já nas operações *SELECT* e *UPDATE*, o banco *MySQL* obteve uma vantagem clara.

3 Metodologia

Dividida em 7 subseções, cada uma com seu objetivo específico de apresentar e detalhar a ferramenta de análise de desempenho entre os bancos de dados NoSQL.

3.1 - Identificação dos SGBD NoSQL

Para a identificação dos SGBD NoSQL, é demonstrado através da figura 8 alguns SGBD NoSQL, baseados no ranking do site DB-Engines.

Figura 8 - Tabelas de Modelos Não Relacionais

Modelo Chave - Valor	Modelo Grafo
Redis	Neo4j
Memcached	Memgraph
RocksDB	Nebulagraph
Erospike	Janusgraph
Modelo Documento	Modelo Colunar
Mongo DB	Apache Cassandra
Couchbase	Hbase
CouchDB	Microsoft Azure
Google Cloud Firestone	Accumulo

Fonte: Elaborado pelos autores

3.2 - Seleção dos SGBD para comparação e análises

Os critérios de seleção dos bancos de dados incluíram alguns dados técnicos, a capacidade de escalabilidade e a flexibilidade dos dados, levando em consideração sua popularidade no mercado atual através do site db-engines em março de 2024, optando por opções gratuitas pela disponibilidade de acesso aos usuários. Nesta análise de *benchmarking*, iremos desconsiderar os modelos chave-valor e o modelo grafo, pois possuem uma finalidade distinta da armazenagem de informações(dados).

Figura 9 - Critérios de Seleção

Tipo de Banco	Nome	Escalabilidade	Disponibilidade	Flexibilidade	Confiabilidade	Popularidade
Colunar	Apache Cassandra	✓	✓			✓
Colunar	Apache HBase	✓				
Colunar	Microsoft Azure	✓	✓			
Colunar	Apache Accumulo	✓		✓		
Documentos	MongoDB		✓	✓	✓	✓
Documentos	Couchbase	✓	✓		✓	
Documentos	Firebase			✓		
Documentos	CouchDB			✓	✓	

Fonte: Elaborado pelos autores

Os melhores bancos de dados que se encaixaram nos critérios de seleção citados acima foram os bancos MongoDB, orientado a documento e o Apache Cassandra, orientado a colunas, destacando-se pela popularidade no mercado atual.

3.3 - Preparação do Ambiente

A base será configurada em ambiente de teste virtual, assim evitando interferências nos resultados apresentados de cada SGBD, obtendo mais clareza nos resultados apresentados pelos gráficos na comparação. Optou-se por configurar dois ambientes idênticos de testes virtuais, para que não haja nenhuma influência de softwares e aplicativos externos que possam prejudicar os testes, utilizando um *software* gratuito, que possua um suporte para instalação da ferramenta, bancos necessários e a configuração

da máquina virtual, utilizamos o *Oracle VM Virtualbox*. Será instalado apenas as ferramentas necessárias em cada ambiente para a execução da ferramenta, cujo terão as mesmas configurações para aplicação deste projeto.

3.3.1 - Máquina Host

3.3.2 - Ambiente Virtual de Teste

3.3.3 - Modelagem da base de dados

A modelagem da base de dados consiste em dois modelos, o primeiro modelo a esquerda na figura 10 de cada banco de dados, são os modelos simples, onde possuem apenas uma tabela de dados. A direita, possuímos o modelos complexos, que são representados por 5 tabelas.

Figura 10 - Modelagem *MongoDB* e *Apache Cassandra* (respectivamente)

<pre> Tabela_01 { _id: <ObjectId>, campo01_Tabela01: string, campo02_Tabela01: string, campo03_Tabela01: string, campo04_Tabela01: int, campo05_Tabela01: int, campo06_Tabela01: int } </pre>	<pre> Tabela_01 { _id: <ObjectId>, campo01_Tabela01: string, campo02_Tabela01: int }, Tabela_02 { _id: <ObjectId>, campo01_Tabela02: string, campo02_Tabela02: int, campo03_Tabela02: int }, Tabela_03 { _id: <ObjectId>, campo01_Tabela03: string, campo02_Tabela03: string, campo03_Tabela03: int, campo04_Tabela03: int }, Tabela_04 { _id: <ObjectId>, campo01_Tabela04: string, campo02_Tabela04: string, campo04_Tabela04: int, campo05_Tabela04: int, campo06_Tabela04: int }, Tabela_05 { _id: <ObjectId>, campo01_Tabela05: string, campo02_Tabela05: string, campo03_Tabela05: string, campo04_Tabela05: int, campo05_Tabela05: int, campo06_Tabela05: int } </pre>	<pre> CREATE TABLE 01 (id PRIMARY KEY, text, text, text, int, int, int); </pre>	<pre> CREATE TABLE 01 (id PRIMARY KEY, text, text, int); CREATE TABLE 02 (id PRIMARY KEY, text, int, int); CREATE TABLE 03 (id PRIMARY KEY, text, text, int, int); CREATE TABLE 04 (id PRIMARY KEY, text, text, int, int, int); CREATE TABLE 05 (id PRIMARY KEY, text, text, text, int, int, int); </pre>
---	---	---	---

3.4 - Plano de validação

O plano de validação deste projeto são os resultados obtidos e a representação dos testes de *benchmarking* através da ferramenta desenvolvida em formato gráfico, facilitando a visualização dos testes aplicados para cada tipo de operação.

3.5 - Desenvolvimento da ferramenta

Para o desenvolvimento, utilizamos o *Visual Studio Code* (versão 1.88, 2024) e a linguagem de programação *Python*.

3.5.1 - Visual Studio Code

O *Visual Studio Code* (VSCode) é um editor de código-fonte altamente reconhecido por sua interface intuitiva e customizável que suporta muitas extensões e múltiplas linguagens de programação que facilita a integração com sistemas de controle de versão, ferramentas de depuração eficientes e uma comunidade ativa com ampla documentação. Essas características tornam o VSCode uma poderosa ferramenta para desenvolvimento, facilitando a produtividade e a colaboração durante o projeto.

3.5.2 - Python

A linguagem Python foi escolhida pela sua simplicidade e legibilidade, o que facilita o desenvolvimento e manutenção do código. Sua versatilidade permite a boa adaptabilidade em diversos tipos de projetos. Além disso, Python possui uma vasta biblioteca de recursos e uma comunidade ativa que oferece suporte robusto com uma perfeita combinação de vastas características que torna a linguagem Python ideal para garantir a eficiência, flexibilidade e escalabilidade do projeto, garantindo a qualidade e sustentabilidade do projeto.

3.6 - Execução da base de dados e testes

A - Definição de Parâmetros de Simulação:

- Número de Registros: Determinar quantos registros serão utilizados no teste.
- Número de Threads: Definição da quantidade de threads que simulam acessos simultâneos ao banco de dados.
- Número de Repetições: Especificar quantas vezes cada operação será repetida durante o teste.

B - Escolha do Banco de Dados e Aplicação das Tabelas:

- Selecionar o banco de dados a ser utilizado (MongoDB ou Apache Cassandra).
- Criar e aplicar as tabelas necessárias no banco de dados escolhido, de acordo com a estrutura exigida pelo teste.

C - Seleção do Tipo de Operação:

-
- Decidir qual operação será testada (Insert, Select, Update, Delete).

D - Início do Teste:

- Iniciar o teste na ferramenta de benchmarking, que passará a executar as operações configuradas.

E - Disparo das Threads:

- A ferramenta inicia as threads selecionadas, que começam a executar as operações de forma concorrente.

F - Consumo da Base de Dados:

- As threads acessam e manipulam a base de dados e tabelas conforme as operações configuradas.

G - Cronometragem do Teste:

- Um cronômetro é iniciado junto com o disparo das threads para monitorar o tempo de execução total do teste.

H - Execução das Operações:

- A operação selecionada (Insert, Select, Update, Delete) é executada repetidamente enquanto o contador de repetições (N) for menor que o número de repetições definido.

I - Finalização do Teste:

- Após a conclusão de todas as repetições, o cronômetro é parado para registrar o tempo total de execução.

J - Armazenamento dos Resultados:

- Os resultados do teste, incluindo métricas de desempenho como tempo de execução, taxa de sucesso e possíveis erros, são armazenados para análise posterior.

3.7 - Análise dos resultados

Na fase de validação dos resultados obtidos, o foco é analisar o tempo de resposta em segundos, como principal métrica de desempenho realizado para cada caso dentro de seu SGBD. O tempo de resposta é crucial para avaliar a eficiência do banco de dados aplicado neste trabalho em diferentes tipos de operações.

4 Resultados

Como resultados deste projeto, selecionamos dois bancos de dados (*MongoDB* e *Apache Cassandra*), para realizar os testes de *benchmarking*. Selecionamos a linguagem de desenvolvimento da ferramenta (*Python*), cuja a mesma proporciona recursos e bibliotecas que auxiliam no desenvolvimento do código. Os testes são evidenciados através das comparações entre os bancos de dados *NoSQL*, concentrando-se na

linguagem de manipulação de dados (*DML*) e na linguagem de consulta de dados (*DQL*), devido à sua relevância no mercado atual.

4.1 - *INSERT*

4.2 - *DELETE*

4.3 - *UPDATE*

4.4 - *SELECT*

5 Considerações Finais

Este trabalho oferece uma visão preliminar sobre análise de desempenho de banco de dados *NoSQL*, abordando conceitos dos tipos de bancos *SQL* e *Nosql*. Até agora, os resultados indicam a importância de um sistema de gerenciamento de banco de dados escolhido de maneira correta e com isso a ferramenta de *benchmarking* é capaz de auxiliar para que seja realizada uma análise e avaliado os resultados para escolha do banco de dados correto para um determinado projeto.

Os próximos passos incluem o desenvolvimento da ferramenta, execução da base de dados, coleta e análise dos resultados obtidos, processos essenciais para aprofundar o projeto e validar as conclusões.

Estamos ansiosos para concluir este trabalho, com os resultados obtidos, permitindo uma visão completa do projeto, ficando explícito que ao realizarmos a análise de *benchmarking*, o desempenho varia conforme o tipo do sistema de gerenciamento de banco de dados *NoSQL* utilizados na comparação, ressaltando a influência dos dados inseridos com base na linguagem *DML* citada anteriormente. Enfatizando a importância da escolha adequada de um sistema de gerenciamento de banco de dados conforme a necessidade do software que será desenvolvido.

Referências Bibliográficas

ABRANTES, Andressa. **Desvendando os Segredos do SQL e Dominando o SQL Server: Do Básico ao Avançado**. Disponível em:

<<https://www.dio.me/articles/desvendando-os-segredos-do-sql-e-dominando-o-sql-server-do-basico-ao-avancado>>. Acesso em: 17 mar 2024.

BEHS, GUILHERME NUNES. **Análise de desempenho de banco de dados NoSQL**. 2020. Disponível em: <https://tconline.feevale.br/tc/files/0001_5192.pdf>. Acesso em 06 abr 2024.

CARDOSO, R.M.F **Bases de Dados NoSQL**. Disponível em:

<<https://core.ac.uk/download/pdf/47141116.pdf>>. Acesso em: 24 fev 2024

CÔRTEZ, S. C.; LUCENA, C.J. , P. **Um Framework para construção de Sistemas de Banco de Dados Móvel com Regras Ativas**. 2001. p. 1–6, out. Disponível em:

<http://bib-di.inf.puc-rio.br/ftp/pub/docs/techreports/01_35_cortes.pdf>. Acesso em: 27 jan 2024

DB-Engines. Ranking. Disponível em:

<<https://db-engines.com/en/ranking>>. Acesso em: 22 mar 2024

EDPRICE-MSFT. **Dados não relacionais e NoSQL** - Azure Architecture Center. Disponível em: <<https://learn.microsoft.com/pt-br/azure/architecture/data-guide/big-data/non-relational-data>>. Acesso em: 06 abr 2024.

GARCIA, V. S.; SOTTO, E. C. S. **Comparativo entre os modelos de banco de dados relacional e não relacional**. Revista Interface Tecnológica, [S. l.], v. 16, n. 2, p. 12-24, 2019. DOI: 10.31510/infa.v16i2.673. Disponível em:

<<https://revista.fatectq.edu.br/index.php/interfacetecnologica/article/view/673>>. Acesso em: 22 mar 2024.

GODOY, Felipe Pereira de; PINHEIRO, Gabriel Benedito. **Análise de desempenho de banco de dados não relacional(MySQL) e não relacional(MongoDB)**. 2021. Trabalho de Conclusão de Curso (Bacharelado em Sistemas de Informação) - FHO-Uniararas, Araras. 2021.

LAUDON, K.C.; LAUDON, J. P. **Sistemas de informação gerenciais**. 1999. Cap 5, p. 136–178. Disponível em:

<https://edisciplinas.usp.br/pluginfile.php/7552318/mod_resource/content/1/Laudon%20e%20Laudon.pdf>.

Acesso em: 18 fev 2024.

MATSUMOTO, Cristina Yoshie, **A importância do banco de dados em uma organização**. Maringá Management: Revista de Ciências Empresariais, v. 3, n.1 - p.45-55, jan./jun. 2006. Disponível em:

<<https://core.ac.uk/download/pdf/199473173.pdf>>. Acesso em: 12 mar 2024.

Mendonça Silva, Phillipe Idivaldo, **Benchmarking em classificação de dados: Redes neurais e árvores de decisão**. 2021 Disponível em:

<<https://monografias.ufma.br/jspui/bitstream/123456789/6914/1/PhilipeSilva.pdf>>. Acesso em: 27 abr 2024.

NASCIMENTO, Matheus Bellio. **MongoDB: Um Estudo Teórico Prático do Conceito de Banco de Dados NoSQL**. 2014. Disponível em:

<[https://www.researchgate.net/profile/Computacao-Unifaccamp/publication/272887058_MongoDB_Um_Estu do_Teorico-Pratico_do_Conceito_de_Banco_de_Dados_NoSQL_-_Trabalho_de_Diplomacao/links/54f24f3b0cf2f9e34f0357e6/MongoDB-Um-Estudo-Teorico-Pratico-do-Conceito-de-Banco-de-Dados-NoSQL-Trabalho-de-Diplomacao.pdf](https://www.researchgate.net/profile/Computacao-Unifaccamp/publication/272887058_MongoDB_Um_Estu_do_Teorico-Pratico_do_Conceito_de_Banco_de_Dados_NoSQL_-_Trabalho_de_Diplomacao/links/54f24f3b0cf2f9e34f0357e6/MongoDB-Um-Estudo-Teorico-Pratico-do-Conceito-de-Banco-de-Dados-NoSQL-Trabalho-de-Diplomacao.pdf)>. Acesso em: 03 fev 2024.

NAVATHE, Ramez Elmasri Shamkant B. , **Sistemas de bancos de dados** Disponível em:

<http://www.tonysoftwares.com.br/attachments/article/5297/Sistema_de_banco_de_dados_Navathe.pdf>.

Acesso em: 03 fev 2024.

OLIVEIRA, Samuel Silva. **Banco de dados não-relacionais: um novo paradigma para armazenamento de dados em sistemas de ensino colaborativo**. ISSN 2175-6147 Macapá, v.2 n. 1, p. 184–194, ago.- dez. 2014. Disponível em: < <https://www2.unifap.br/oliveira/files/2016/02/35-124-1-PB.pdf> >. Acesso em: 22 mar 2024.

PIRES, Carlos E. S, NASCIMENTO, Rilson O, SALGADO, Ana C. **Comparativo de desempenho entre banco de dados de código aberto**. 2006 Disponível em:

<https://www.researchgate.net/profile/Ana-Carolina-Salgado/publication/242213657_Comparativo_de_Dese mpenho_entre_Bancos_de_Dados_de_Codigo_Aberto/links/004635296426ba710e000000/Comparativo-de-Desempenho-entre-Bancos-de-Dados-de-Codigo-Aberto.pdf>. Acesso em: 08 abr 2024.

TOMIO, Gustavo Veloso. **Utilizando a tecnologia de banco de dados NoSQL**. 2015. Disponível em:

<http://repositorio.utfpr.edu.br/jspui/bitstream/1/15944/2/PG_COCIC_2015_1_03.pdf>. Acesso em: 28 jan 2024.