# Estudo comparativo entre sistemas de gerenciamento de bancos de dados relacionais e não relacionais para o armazenamento e busca de metadadados MARC

**Jader Osvino Fiegenbaum[1], Evandro Franzen[1]**
[1]Centro Universitário Univates
Rua Avelino Tallini, 171, Bairro Universitário – Lajeado – RS – Brasil

***Abstract.*** *The use of archive management system reduces the labor and cost of internal libraries. These systems mostly need to store the data in a metadata standard for dealing with flexible and dynamic information such as MARC. Traditionally relational databases are used, most of which have rigid structures for data storage. The purpose of this study was to compare the performance of DBMS's PostgreSQL and MongoDB for storage and search MARC metadata.*

***Resumo.*** *A utilização de sistemas de gestão de acervo reduz o trabalho e o custo interno das bibliotecas. Estes sistemas, em sua maioria necessitam armazenar os dados em algum padrão de metadados para lidar com informações flexíveis e dinâmicas como o MARC. Tradicionalmente são usados bancos de dados relacionais, que em sua maioria apresentam estruturas rígidas para o armazenamento de dados. O propósito deste estudo foi comparar a performance dos SGDB PostgreSQL e MongoDB para o armazenamento e busca de metadados MARC.*

## 1. Introdução

O processo de automação de bibliotecas é indispensável para o atendimento de demandas informacionais, principalmente em bibliotecas universitárias, pois os usuários necessitam ter acesso à informação de diversas áreas. O uso de softwares para o gerenciamento de acervos bibliográficos é um fator decisivo no desempenho das funções da biblioteca.

Para suprir as necessidades destes sistemas, em 1960 foi criado o padrão de metadados MARC (*Machine Readable Cataloging)* para armazenar registros bibliográficos de diferentes tipos de objetos, com o objetivo de automatizar o processo de catalogação (FURRIE ,2003). Gil-Leiva (2007) e Furrie (2003), definem o padrão MARC como um conjunto de números, letras e símbolos combinados e adicionados aos registros catalográficos.

Um registro bibliográfico que segue o formato MARC possui dados encapsulados em campos e subcampos, que podem ser entendidos como *tags*. Cada *tag* permite organizar e gerir a informação, além de permitir a sua recuperação. Portanto, diferentes tipos de objetos não serão catalogados da mesma forma, não contendo os mesmos conjuntos de campos e subcampos, uma vez que possuem estrutura diferente.

Em função da estrutura dinâmica, o armazenamento de tais informações não se adapta tão facilmente ao modelo relacional, que apresenta uma estrutura rígida, baseada em tabelas e campos. A busca por sistemas com estruturas mais flexíveis levou o

desenvolvimento de bancos NoSQL (Not Only SQL), que possuem como principal característica armazenar dados com estrutura flexível e tratamento de grandes volumes de dados (Lóscio, Oliveira e Pontes ,2011).

Os estudos sobre modelos NoSQL tem se tornado cada vez mais frequentes, principalmente com o foco na comparação (POLITOWSKI, MARAN, 2014) e interoperabilidade com modelos relacionais (SCHREINER, 2015) ou ainda como alternativa para o armazenamento e busca de dados não convencionais (SCHULZ, 2016).

O fato de registros bibliográficos em MARC serem dinâmicos e semiestruturados possibilita que os mesmos possam ser armazenados e processados em bancos NoSQL. A principal questão a ser investigada é se o armazenamento em um modelo flexível, como NoSQL apresenta um desempenho compatível ou superior aos modelos tradicionais e, desta forma, pode ser uma alternativa para armazenar estes tipos de dados.

## 2. Procedimentos metodológicos

A metodologia usada neste trabalho foi de natureza quantitativa, pois envolveu a coleta de dados relacionados ao desempenho de sistemas que implementam os modelos relacional e NoSQL em consultas que utilizam dados no formato MARC. Os sistemas de gerência de bancos de dados escolhidos foram o PostGreSQL (relacional) e o MongoDB (orientado a documentos). Ambos possuem código aberto e estão entre os sistemas mais utilizados no cenário atual.

A modelagem de dados relacional inclui duas tabelas para controlar as *tags* e os valores para cada material. Neste caso para recuperar ou atualizar os dados é necessário acessar a tabela de cadastro do material e as tabelas de definem as informações no formato MARC. O armazenamento no MongoDB envolveu somente uma coleção, com estrutura flexível, para manter as informações relacionadas ao padrão. Para execução dos testes, foi desenvolvido um protótipo que contempla testes de busca, tolerância a falhas, carga de dados, stress e volume do banco de dados (Figura 1).


Figura 1 – Interface do protótipo para execução dos testes

O teste de busca teve como objetivo determinar o desempenho de cada SGBD (Sistema de Gerenciamento de banco de dados) ao lidar com a busca de materiais. O teste consistiu em medir o tempo ao buscar materiais cadastrados, combinando campos MARC nos filtros. Uma das estratégias foi analisar o desempenho de consultas que

retornam dados e a segunda é a execução de consultas que apenas contam registros.

O teste de volume buscou verificar a desempenho do sistema ao lidar com grande volume de dados em relação ao tempo. Foram executadas consultas que retornaram todos os materiais armazenados, e foram considerados os tempos de execução das consultas e não da obtenção dos dados, a partir dos recursos ou cursores retornados em cada caso.

## 3. Resultados e discussão

Nesta seção apresenta os resultados obtidos nos testes de desempenho de busca e volume. A execução dos testes de busca foi realizada de duas maneiras, a primeira exibe os registros retornados e a segunda exibe somente a quantidade de registros e tempo de execução da consulta em ambos SGDB's. Em ambos os casos, foi possível combinar filtros através de seleção de campos e operadores booleanos (e/ou). A Tabela 1 apresenta os resultados.

Tabela 1 – Resultado do teste de busca

| Condição | Quantidade de registros | Postgres (exibe registros) | Postgres (conta registros) | Mongo (exibe registros) | Mongo (conta registros) |
|---|---|---|---|---|---|
| CONTEM 'BANCO' E 650.a CONTEM 'INFORMATICA' E 100.a CONTEM 'el' | 10000 | 0,489892s | 0,446063s | 1,541534s | 0,232550s |
| CONTEM 'BANCO' E 650.a CONTEM 'INFORMATICA' OU 100.a CONTEM 'el' | 40000 | 1,001012s | 0,831537s | 5,965069s | 0,458339s |
| CONTEM 'SISTEMAS' E 260.a CONTEM 'SAO PAULO' E 650.a CONTEM 'banco' | 20000 | 2,870094s | 0,613667s | 2,984691s | 0,381753s |
| CONTEM 'SISTEMAS' E 260.a CONTEM 'RIO' E 650.a CONTEM 'banco' | 10000 | 0,490941s | 0,451199s | 1,734503s | 0,386137s |
| CONTEM 'SISTEMAS' E 260.a CONTEM 'RIO' OU 650.a CONTEM 'banco' E 100.a CONTEM 'Garcia' | 10000 | 0,505619s | 0,472283s | 1,571384s | 0,267236s |

Os resultados apontam uma performance superior do MongoDB em relação ao PostgreSQL quando utilizada a interface que não exibe os registros, o que demonstra a performance do banco de dados. Porém, a exibição dos registros indica que o PostgreSQL possui performance superior no contexto da aplicação. Isso ocorreu porque no caso da exibição de registros são retornados todos os resultados do *resource* e do cursor. Iterar sobre estes dados no PostgreSQL é mais rápido do que iterar sobre todo o cursor do MongoDB. A diferença entre eles aumenta proporcionalmente de acordo com a quantidade de registros.

Tabela 2 – Resultado do teste de volume

| Registros | Tempo Postgres (segundos) | Tempo MongoDB (segundos) |
|---|---|---|
| 21 | 0,00066363 | 0,00004863 |
| 2100 | 0,00277275 | 0,00004775 |
| 21000 | 0,02068025 | 0,00004663 |
| 210000 | 0,19547763 | 0,00005400 |
| 525000 | 0,45658263 | 0,00004738 |
| 1050000 | 0,93997550 | 0,00005313 |

Observa-se que o sistema relacional obtém dados armazenados na memória RAM, e o segundo mantém uma conexão aberta com o banco de dados. O MongoDB foi concebido para grande volume de dados, na ordem de terabytes de informação. Nesse cenário seria impossível o sistema retornar dados sem utilizar um cursor, pois não haveria memória disponível para isso.

No teste de volume foram executadas consultas que retornam todos os materiais armazenados. Os resultados são apresentados na Tabela 2.

Os resultados apontam uma performance superior do padrão NoSQL. É possível perceber que o tempo do banco relacional aumenta proporcionalmente em relação a quantidade de registros. O mesmo não ocorre para o MongoDB, devido ao fato deste utilizar memória virtual e também ter sido executado logo após o teste de carga. Portanto, é provável que todos os dados ainda estivessem em memória RAM.

## 4. Conclusão e trabalhos futuros

Este trabalho apresentou uma análise comparativa de SGBDs, com ênfase no desempenho demonstrado em consultas. Embora durante as pesquisas tenham sido realizados também testes de carga, stress e tolerância a falhas, optou-se neste artigo por apresentar somente os resultados das consultas.

Observou-se que o modelo NoSQL é uma alternativa promissora, pois exibe um bom desempenho na execução das consultas. Entretanto, é necessário criar alternativas que melhorem os resultados na aplicação, na exibição dos dados pela interface. O teste de volume demonstra que o MongoDB possui performance superior ao PostgreSQL, isso se deve principalmente ao fato do registro bibliográfico ser armazenado em apenas um documento, enquanto para armazenar um registro completo com todas as etiquetas MARC no PostgreSQL são necessários múltiplos registros, em tabelas diferentes.

Trabalhos futuros apontam para a necessidade de avaliar outras estruturas de armazenamento, como grafos, colunas e o desenvolvimento de aplicações que permitam testes reais, com usuários e dados existentes em sistemas de gestão de bibliotecas.

## Referências

FURRIE, B. **Understanding MARC bibliographic: machine-readable cataloging**. 7 ed. rev. Washington, D. C.: Library of Congress; Follet Software, 2003.

GIL-LEIVA, Isidoro. **A indexação na internet**. Brazilian Journal of Information Science. v.1, n.2, p.47-68. 2007.

LÓSCIO, Bernadette Farias; OLIVEIRA, Hélio Rodrigues de; PONTES, César de Sousa. **NoSQL no desenvolvimento de aplicações Web colaborativas**. Simpósio Brasileiro de Sistemas Colaborativos, 2011. Paraty. 2011.

POLITOWSKI, Cristino; MARAN, Vinícius. **Comparação de Performance entre PostgreSQL e MongoDB**. ERBD 2014, São Francisco do Sul, 2014.

SCHREINER, Geomar A.; DUARTE, Denio; DOS SANTOS MELLO, Ronaldo. **Análise de Abordagens para Interoperabilidade entre Bancos de Dados Relacionais e Bancos de Dados NoSQL**. ERBD 2015. Caxias do Sul, RS. 2015.

SCHULZ, Wade L. et al**. Evaluation of relational and NoSQL database architectures to manage genomic annotations**. Journal of Biomedical Informatics, v. 64, p. 288-295, 2016.