

## **NoSQL: Ferramentas, desafios de aplicabilidade em um case real**

**JORGE ROGÉRIO PINHEIRO LOBO FILHO**

Estácio do Pará  
R. Municipalidade, 839 - Reduto,  
Belém - PA, 66050-350  
rogerlobo2004@hotmail.com

**THIAGO AFONSO OLIVEIRA DE ALMEIDA**

Estácio do Pará  
R. Municipalidade, 839 - Reduto,  
Belém - PA, 66050-350  
thiagoalmeida78@gmail.com

**ANTONIO SOARES LOBATO**

Estácio do Pará  
R. Municipalidade, 839 - Reduto,  
Belém - PA, 66050-350  
alobato@live.estacio.br

**Resumo** *O objetivo geral deste artigo mostra como um banco relacional necessita de uma melhor escalabilidade quando se trata de banco NoSQL, pois a aplicabilidade do mesmo melhora o funcionamento de um banco tradicional. Evidencia o uso do NoSQL para lidar com grandes volume de dados trazendo uma nova estratégia de armazenamento melhorando e muito o escalonamento do banco.*

*A proposta do NoSQL é flexibilizar a grande quantidade de informações que um banco relacional possui, fazendo com que fique mais apresentável e mais fácil de ser utilizado.*

**Palavras-Chave:** *NoSQL; Aplicabilidade; Volume de dados; Armazenamento; Escalabilidade; MongoDB.*

**Abstract** *The object general of this article shows how a relational database needs better scalability when it comes to NoSQL database, because the applicability of it improves the functioning of a traditional bank. It highlights the use of NoSQL to handle large volumes of data bringing a new storage strategy improving and scaling the bank.*

*The proposed NoSQL is more flexible the large amount of information that a relational database features, making make it more presentable and easier to use.*

**Keywords:** *NoSQL; Applicability; Data volume; Storage; Scalability; MongoDB.*

## 1 INTRODUÇÃO

[1] O modelo relacional é utilizado de forma bastante abrangente em qualquer sistema de bancos de dados nos últimos tempos. E devido o crescimento de maneira exponencial dos dados armazenados, o modelo relacional de bancos de dados tem mostrados certos limites para grandes volumes dos mesmos, sendo assim obrigados a dotar outras ferramentas para o mesmo propósito.

Decorrente desses fatores em questão, a “escalabilidade” de seus sistemas se tornava uma necessidade. E de tal necessidade nascia o NoSQL. [1] Um banco de dados que poderia se tornar amplamente escalável em volume de dados.

[1] O NoSQL oferece a proposta de tornar as ferramentas de tratamentos de dados mais hábeis nesta função, mas não somente de maneira quantitativa dos dados, mas a velocidade, variedade, disponibilidade e integridade desses dados também são levadas em conta na hora do tratamento dos mesmos.

## 2 OBJETIVO

Neste artigo, teremos como objetivo geral analisar o funcionamento de um banco de dados NoSQL (banco de dados não relacional), como esse entendimento podemos analisar suas aplicabilidades no complemento de um banco de dados tradicional.

A análise tem por finalidade realizar um estudo sobre a arquitetura de alguns SGBDs NoSQL, evidenciar o motivo pelo qual eles conseguem lidar com grandes volumes de dados e mostrar algumas propriedades que o NoSQL baseia-se para fazer a gestão de dados.

Este artigo também vai apontar uma possível forma de aplicação em um cenário real, informando o bom funcionamento do sistema utilizando-se do NoSQL para armazenar e coletar de forma com que a empresa tenha menos custo e um escalonamento de dados mais consistentes.

O estudo prevê também apresentar testes comparativos de desempenho entre as duas ferramentas (SGBDs e NoSQL). Destacando seus resultados e destacar vantagens por meio de gráficos.

## 3 CONCEITOS INICIAIS SOBRE BANCO DE DADOS

Antes de começarmos a falar sobre banco de dados, vamos primeiramente aprender um pouco sobre eles. Vamos abordar a seguir o primeiro modelo de banco de dados a ser criado.

[1] Bancos de dados relacionais no modelo estrutural foi o primeiro modelo de banco de dados a ser criado, sendo estruturado unicamente no formato de tabelas. Sendo este um modelo abstrato, que por sua vez, os dados podem ser armazenar, manipular e recuperar dados. A utilização deste modelo de banco de dados é muito comum em diversos sistemas, sendo ele o mais usado.

[1] Os dados ficam armazenados em um único lugar, e nesse lugar ficam as tabelas com todos os dados armazenados. Cada tabela é separada por entidades.

Essas entidades são classificadas por setor, por exemplo: Temos uma entidade chamada “Funcionário”, essa entidade é a tabela onde ficam armazenadas todas as informações dos funcionários de uma determinada empresa, conforme a figura 1 abaixo.

id_pessoa	nome	telefone	cor_cabelo	nascimento	morada
1	Pedro Pinto	412482749	preto	1980	Guarda
2	Vitor	214748367	castanho	1977	Porto
3	Pedro Simões	213412431	castanho	1976	Lisboa
4	Marisa Pinto	232434291	loiro	1986	Viseu
5	Pedro Gaspar	101101011	castanho	1988	Coimbra
7	Rui Lopes	239456732	castanho	1984	coimbra
8	Silvia Lopes	239891267	ruivo	1981	leiria
9	Paulo Lopes	233876547	preto	1979	coimbra

Figura 1. Exemplo de tabela separada por entidades.

[1] Essas tabelas podem se relacionar com outras tabelas, sendo assim um modelo de entidades que se relacionam através das chaves primárias como mostra a figura 2 abaixo.

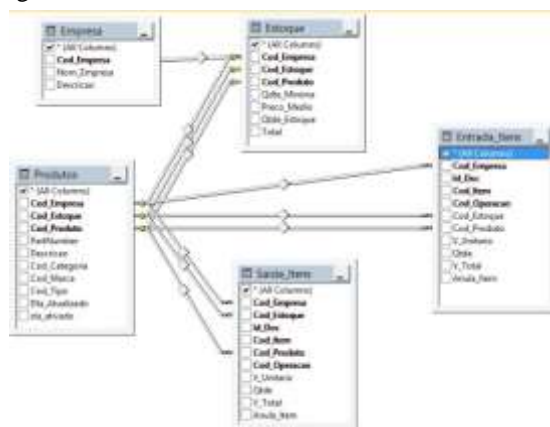


Figura 2. Exemplo de tabelas de banco de dados relacional.

A figura 2 acima demonstra claramente um modelo simples de banco de dados com tabelas se relacionando entre si. Perceba que entre elas podem existir mais de um relacionamento, facilitando assim a organização e o controle das informações armazenadas no banco de dados.

Vimos acima como é o funcionamento básico de um banco de dados modelo relacional tradicional. A partir de agora iremos ver o porquê do surgimento da ferramenta NoSQL e para qual finalidade ela foi criada.

## 4 POR QUE NoSQL ?

[2] O objetivo dos projetistas de banco de dados de organizações de grande porte passou a ser desenvolver uma nova estratégia de armazenamento na qual pudessem estar livres de certas estruturas e regras presentes no Modelo Relacional. Assim, foram surgindo soluções que pareciam voltar no tempo, retornando aos simples sistemas de gerenciamento de arquivos.

[3] O termo NoSQL surgiu em 1998, a partir de uma solução de banco de dados que não oferecia uma interface SQL, mas esse sistema ainda era baseado na arquitetura relacional.

[3] Posteriormente, o termo passou a representar soluções que promoviam uma alternativa ao Modelo Relacional, tornando-se uma abreviação de Not Only SQL (não apenas SQL), sendo utilizado principalmente em casos em que o Modelo Relacional não apresentava desempenho adequado.

[3] O propósito, portanto, das soluções NoSQL não é substituir o Modelo Relacional como um todo, mas apenas em casos nos quais seja necessária uma maior flexibilidade da estruturação do banco. Em português já existe o acrônimo MRNN para Modelo Relacional Não-Normalizado.

#### **4.1 CLASSIFICAÇÃO DE BANCO DE DADOS NoSQL**

O NoSQL possui características em comum com banco de dados relacionais, tais como serem livres de esquema, promoverem alta disponibilidade e maior escalabilidade, os sistemas de banco de dados NoSQL possuem diversas singularidades.

[4] Quando à distribuição de dados, certos sistemas promovem o particionamento e a replicação de dados, diferenciando-se dos bancos tradicionais que deixam essa tarefa para o cliente. A maioria das soluções é distribuída, exemplos como no caso do Amazon Dynamo, do CouchDB, do MongoDB, do BigTable e do Cassandra.

[4] Quanto ao modelo de dados, existem quatro categorias básicas: os sistemas baseados em armazenamento chave-valor, como é o caso do Amazon Dynamo; os sistemas orientados a documentos, entre os quais temos o CouchDB e o MongoDB; os sistemas orientados a coluna, que tem como exemplos o Cassandra e o Big Table; e os sistemas baseados em grafos, como são os casos do Neo4j e do Info Grid.

#### **4.2 UM POUCO MAIS SOBRE NoSQL**

[4] Quando se é feito a análise dentro de possibilidades de se optar por uma estratégia NoSQL em detrimento de um SGBD tradicional, é preciso levar em consideração algumas questões básicas como por exemplo os critérios de escalonamento, consistência de dados e disponibilidade.

[4] A questão do escalonamento é essencial porque é justamente nesse ponto em que os bancos NoSQL apresentam as principais vantagens em relação aos SGBDs relacionais, ou seja, o NoSQL foi feito exatamente para isso, enquanto o SGBD tradicional possuem uma estruturação menos flexível e menos adaptada para cenários em que o escalonamento faz-se necessário.

[4] Seja uma determinada aplicação web, com arquitetura simples conforme ilustrado na Figura 3, para se promover opções de atendimento ao número crescente de usuários, os responsáveis pelo sistema poderiam optar pelo upgrade no servidor, solução também conhecida como escalonamento Vertical (scaleup); ou pelo aumento na quantidade de servidores, solução também conhecida como escalonamento horizontal (scale out).

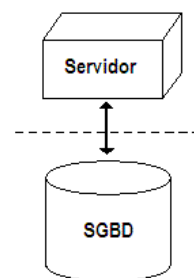


Figura 3. Arquitetura Tradicional de uma

O escalonamento vertical, que se caracteriza por sua simplicidade, tem sido tradicionalmente mais indicado para as camadas de banco de dados, enquanto o escalonamento horizontal tem sido mais utilizado nas camadas das aplicações, principalmente para a web como mostra a figura 4.

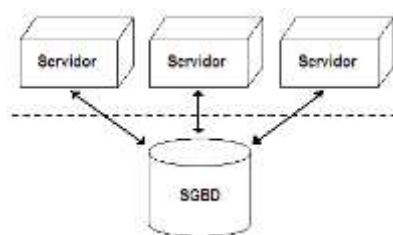


Figura 4. Utilização do Escalonamento Horizontal.

## **5 HISTÓRICO DO CASE**

Foi realizado um levantamento recente em uma empresa que usa atualmente uma aplicação com bancos de dados relacionais (PostgreSQL). Esses dados já haviam sido importados anteriormente de outra aplicação, que usava o mesmo esquema de banco de dados, só que o mesmo era diferente o atual (FireBird).

Pois bem, a conversão desses dados foi feita, só que nem todas as informações foram importadas de maneira correta e outras simplesmente não foram importadas.

Até hoje, a empresa dona de massa de dados enfrenta grandes dificuldades com a grande quantidade de dados inconsistentes. Além de perda de informações, existe um grande problema com a lentidão do banco na hora de requisitar informações. A aplicação demora tempo bastante para processar um relatório por exemplo. Tempos este que é considerado anormal para um banco de dados desse nível.

Diante de uma problemática dessas, entendemos a necessidade de uma solução para o problema. Tudo bem que o estrago já havia sido feito, mas podemos ilustrar de alguma forma que o NoSQL poderia se destacar diante desse “cenário catastrófico”.

### **5.1 PROPOSTA DE SOLUÇÃO**

Vimos a solução do problema mencionando a cima através da implantação de uma ferramenta NoSQL, ferramenta essa que tornasse fácil a transferência de um “banco de dados relacional” para um modelo “não relacional”.

[5] A proposta foi, adotar o banco de dados MongoDB para substituição do PostgreSQL, que por sua vez é o banco que se utiliza atualmente.



Resultados obtidos	Numero de vezes		
	100	1.000	10.000
Insert	0,0378	0,3750	3,587
Busca Simples	0,0145	0,1458	1,487
Busca Complexa	0,0204	0,2087	2,105

Na tabela 2 acima, apresentamos os resultados obtidos com o banco de dados MongoDB. Como podemos ver, foi executado o mesmo nível de testes que foi aplicado com o banco PostgreSQL. Na contra mão do banco de dados concorrente, o MongoDB obteve resultados muito superiores. Um destaque foi na velocidade no tratamento desses dados pesquisados, em 10.000 vezes, obtivemos resultados notavelmente baixos, cerca de 35 minutos apenas.

Uma comparação lado a lado entre as tabelas para que não reste duvidas na obtenção dos dados de desempenos entre as duas ferramentas de estudo em questão.

**Tabela 1: Desempenho banco PostgreSQL**

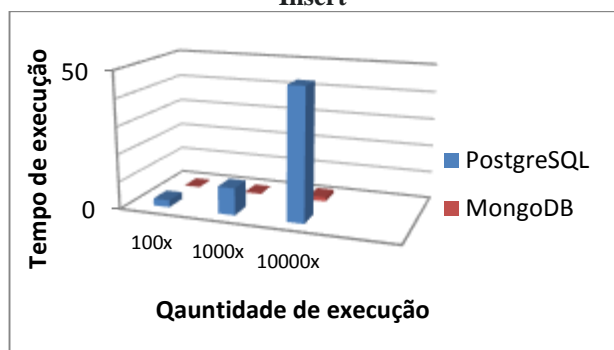
Resultados obtidos	Numero de vezes		
	100	1.000	10.000
Insert	1,356	8,843	36,254
Busca Simples	2,598	9,857	47,526
Busca Complexa	3,053	10,875	53,756

**Tabela 2: Desempenho banco MosngoDB**

Resultados obtidos	Numero de vezes		
	100	1.000	10.000
Insert	0,0378	0,3750	3,587
Busca Simples	0,0145	0,1458	1,487
Busca Complexa	0,0204	0,2087	2,105

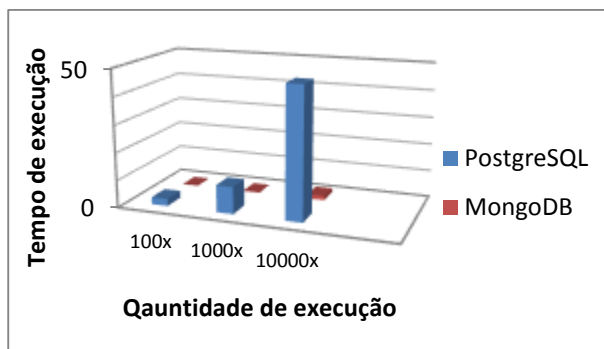
Obtivemos os mesmos resultados expondo-os em formato de gráficos. Separamos os resultados em 3 gráficos, exibindo cada modulo dos resultados obtidos.

#### **Insert**

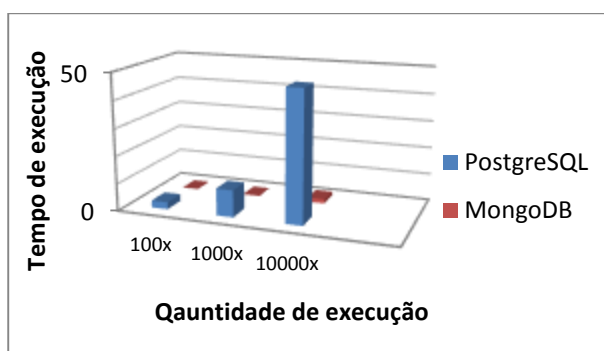


**Figura 7: Gráfico comparação de teste "Insert"**

#### **Busca Simples**



**Figura 8: Gráfico comparação de teste "Busca Simples"**



**Figura 9: Gráfico comparação de teste "Busca Complexa"**

Como podemos ver acima a ilustração, podemos perceber o quão grande a diferença de tempo de execução de ambas as ferramentas. Notamos o gral de poder de execução de grandes volumes de dados que o MongoDB pode executar em tão pouco tempo em comparação ao banco de dados PostgreSQL.

## **8. CONCLUSÃO**

Em nossos estudos, foi utilizado dois modelos de banco de dados totalmente diferentes, com propostas similares. Concluimos a superioridade do banco de dados MongoDB, sendo ele bem mais eficiente nos testes de inserção de dados e buscas. Sendo esta uma possível solução para o problema de lentidão do banco PostgreSQL da empresa que se fez este estudo de caso.

Vale ressaltar que o MongoDB (NoSQL) é uma ferramenta que foi criada exatamente para ter performance, tanto na gravação de dados quanto na consulta dos mesmos.

Os bancos de dados NoSQL não é nenhuma filosofia que prega contra os bancos de dados relacionais, mito pelo contrário, NoSQL pode ser usado lado a lado com os bancos de dados convencionais que existem no mercado, desde que sejam devidamente configurados.

Os testes realizados foram exclusivamente para fins acadêmicos, de maneira que pudesse abordar e exemplificar da melhor forma possível o tema em questão.

Para trabalhos futuros, se grande janela de possibilidades de aplicação da solução em questão se



abrem: Implementar a resolução em questão no estudo de caso abordado. Analisar e estudar outros modelos de bancos como: Tokutek® e suas aplicabilidades em um case real Relacionando o mesmo com outros bancos como: MySQL, FireBird, etc.

## REFERENCIAS

- [1] BRITO, Ricardo W. **Faculdade Farias Brito e Universidade de Fortaleza. Bancos de Dados NoSQL x SGBDs Relacionais: Análise Comparativa.** <Disponível em: <http://www.infobrasil.inf.br/userfiles/27-05-S4-1-68840-Bancos%20de%20Dados%20NoSQL.pdf>>, Consultado em: 23 de Novembro de 2015.
- [2] COSTA, Elisangela R. **Faculdade de tecnologia de São Paulo. BANCOS DE DADOS RELACIONAIS.** <Disponível em: <http://www.fatecsp.br/dti/tcc/tcc0025.pdf>>, Consultado em: 18 de Setembro de 2015.
- [3] GONÇALVES, Otávio. **Conceitos NoSQL** <Disponível em: <http://www.devmedia.com.br/conceito-nosql-cassandra-em-java/22863>>, Consultado em: 20 de Abril 2016.
- [4] **NoSQL ou SQL? Quando uso um, outro ou ambos** <Disponível em: <http://www.itexto.net/devkico/?p=1199>>, Consultado em: 20 de Abril 2016.
- [5] POLITOWSKI, Cristiano; MARAN, Vinícius. **Universidade Regional ^ do Noroeste do Estado do Rio Grande do Sul. Comparação de Performance entre PostgreSQL e MongoDB** <Disponível em: <http://www.lbd.dcc.ufmg.br/colecoes/erbd/2014/003.pdf>>, Consultado em: 20 de Abril 2016.
- [6] GEROSA, Marco. **NOSQL na Web 2.0: Um Estudo Comparativo de Bancos Não-Relacionais para Armazenamento de Dados na Web 2.0.** <Disponível em: [http://www.lbd.dcc.ufmg.br/colecoes/wtdbd/2010/sbbd\\_wtd\\_12.pdf](http://www.lbd.dcc.ufmg.br/colecoes/wtdbd/2010/sbbd_wtd_12.pdf)>, Consultado em: 11 de Setembro de 2015.
- [7] MEDEIROS, Higor. **Introdução aos bancos de dados MongoDB.** <Disponível em: <http://www.devmedia.com.br/introducao-ao-mongodb/30792>>, Consultado em: 11 de Setembro de 2015.
- [8] PEREIRA, Felipe S; BORGES, Hermes; RUBENS, Helio. **Unitri – Centro Universitário do Triângulo. Utilização de Banco de Dados NoSql em Ambientes Corporativos.** <Disponível em: <http://www.computacao.unitri.edu.br/erac/index.php/erac/article/view/163/230>>, Consultado em: 17 de Setembro de 2015.
- [8] SADALAGE, Pramod J; FOWLER, Martin. **NoSQL Um Guia Conciso para o Mundo Emergente da Persistencia Poliglota.** Ed: Novatec LTDA, 1ª edição, São Paulo: 2013.
- [9] SOROJET. **Introdução ao NoSQL.** <Disponível em: [http://www.sorojet.com.br/termos\\_cartao.pdf](http://www.sorojet.com.br/termos_cartao.pdf)>, Consultado em 15 de Setembro de 2015.
- [10] STEPPAT, Nico. **Bancos de dados não relacionais e o movimento NoSQL.** <Disponível em: <http://blog.caelum.com.br/bancos-de-dados-nao-relacionais-e-o-movimento-nosql/>>, Consultado em: 08 de Setembro de 2015.
- [11] “NoSQL Relational Database Management System: Home Page”. Strozzi.it. [http://www.strozzi.it/cgi-bin/CSA/tw7/I/en\\_US/nosql/Home](http://www.strozzi.it/cgi-bin/CSA/tw7/I/en_US/nosql/Home).