

Análise Comparativa em Operações de CRUD entre Banco de Dados NoSQL: MongoDB e CouchDB em Ambiente Windows

Beatriz P. Simões, Cláudio M. Perrenoud, Kelton Augusto Pontara da Costa

Curso de Tecnologia em Banco de Dados – Faculdade de Tecnologia de Bauru
(FATEC)

Rua Manoel Bento Cruz, nº 30, Quadra 3 – Centro – 17.015-171 – Bauru, SP – Brasil

biasimoes24@gmail.com, mperrenoud@hotmail.com, kelton.costa@gmail.com

Abstract. *The objective of this work is to perform a comparative study of the performance of the database systems: MongoDB and CouchDB. The case study carried out presents the execution times of CRUD operation, that is, insertion, query, update and exclusion sets of the two database management systems, allowing to obtain these metrics in a quantitative way, and present them in the form of comparison tables for each operation, comparing the performance of both SGDBs performing the same set of operations. For the results obtained through the experiments performed on the time question in the insertion, query, update and deletion operations, within this scenario, we conclude that MongoDB is the most performative in relation to CouchDB.*

Resumo. *O objetivo deste trabalho é realizar um estudo comparativo da performance de desempenho dos sistemas de bancos de dados: MongoDB e CouchDB. O estudo de caso a ser realizado apresentará os tempos de execução do conjunto de operação básicas CRUD, ou seja, inserção, consulta, atualização e exclusão dos dois SGBDs, permitindo obter estas métricas de maneira quantitativa, e apresentá-las em forma de tabelas comparativas para cada operação, comparando a performance de ambos os SGBDs. Pelos resultados obtidos na questão tempo nas operações de inserção, consulta, atualização e exclusão, dentro desse cenário, conclui-se que o MongoDB é o mais performático em relação ao CouchDB.*

1. Introdução

Com o constante crescimento da tecnologia, torna-se importante a existência de uma maior escalabilidade, flexibilidade, segurança, eficiência e capacidade de resposta, quando se trata de Banco de Dados [Nascimento 2010]. Os bancos de dados não relacionais (NoSQL) são ideais para muitos aplicativos que tenham necessidades maiores de armazenamento e desempenho tais como, aplicativos de *Big Data*, mobilidade e *Web*, do que as oferecidas pelos bancos de dados relacionais tradicionais e tem feito os desenvolvedores reavaliarem como eles armazenam e mantêm esses dados.

Atualmente com a crescente popularização das redes sociais, a geração de conteúdo por dispositivos móveis, bem como o número cada vez maior de pessoas e

seus dispositivos conectados, faz com que o trabalho de armazenamento de dados com o objetivo de utilizá-los em ferramentas analíticas, comece a esbarrar nas questões de escalabilidade e custos de manutenção desses dados.

O NoSQL, surge, portanto, como uma alternativa de um banco de dados mais flexível no suporte de dados, respondendo geralmente com maior velocidade e podem ser escalados com maior facilidade que os bancos de dados relacionais devido a escalabilidade horizontal que consiste em adicionar mais capacidade no sistema distribuído inserindo uma nova máquina no cluster, uma das técnicas mais conhecidas é o *Sharding*, que divide os dados em múltiplas tabelas a serem armazenadas ao longo de diversos nós na rede, significa dizer que esse método rompe a cadeia de relacionamentos, assim, sistemas que apresentam rápido crescimento do volume de dados armazenados têm mais facilidade para escalarem seu projeto [Lóscio; Oliveira; Pontes 2011].

Este trabalho apresenta uma análise de desempenho entre dois modelos de banco de dados orientado a documentos (MongoDB e CouchDB) da estrutura NoSQL, utilizando como parâmetro o tempo de execução, que serão abordados pelas Seções posteriores e utilizando o modelo de operações básicas CRUD (acrônimo de *Create, Read, Update e Delete* na língua Inglesa) para análise entre as estruturas mencionadas.

Este trabalho está estruturado da seguinte maneira: Na Seção 2 será mostrado a definição do banco de dados NoSQL, como e porque surgiu, suas características principais e estrutura. Na Seção 3 apresenta-se a estrutura do banco de dados MongoDB. Na Seção 4 será abordado a descrição do banco CouchDB, elencando seus principais aspectos e modelo de dados. Na Seção 5, será exibido materiais e métodos, ou seja, qual o perfil foi definido para execução dos testes, hardware utilizado e posteriormente a exibição dos resultados obtidos. Na Seção 6 apresenta-se a conclusão e resultados da análise de desempenho entre os bancos não relacionais.

2. Banco de dados NoSQL

O termo NoSQL foi utilizado pela primeira vez em 1998 por Carlo Strozzi para citar um banco de dados relacional *open-source* que omitia o uso de SQL, chamado Strozzi NoSQL. O nome surgiu pelo fato de que o banco utilizava *scripts* de *shell* de dados e não o SQL como linguagem de busca. Porém, o uso do termo como hoje é conhecido não se refere ao banco desenvolvido por Strozzi.

Segundo Fowler (2015), os bancos de dados NoSQL podem ser considerados como bancos que não dependem de um rigoroso esquema para os registros, podendo-se utilizar de um hardware comum, não utilizando o modelo algébrico comum dos bancos de dados relacionais.

Conforme Abramova; Bernardino; Furtado (2014) o desempenho dos modelos de banco de dados NoSQL na questão de armazenamento é significativamente melhor do que os modelos tradicionais, pois focam na redução do acesso a disco com o objetivo de tornar o desempenho de sistemas que os utilizem mais rápidos fazendo o uso da memória volátil para o mapeamento de registros.

Para evitar a perda de informações, o NoSQL tem uma arquitetura distribuída tolerante a falhas que se baseia na redundância de dados em vários servidores. Dessa

forma o sistema pode ser escalado facilmente agregando mais servidores e assim a falha de um deles pode ser tolerada.

As empresas de renome mundial estão envolvidas no desenvolvimento do modelo NoSQL, tais como, Facebook, Google e Amazon, uma vez que a demanda de dados que os SGBDs tradicionais não conseguiam suprir as necessidades[Lóscio; Oliveira; Pontes 2011].

Geralmente os Bancos NoSQL possuem estrutura simplificada, sem estrutura de relacionamento, são projetados para trabalharem com uma grande capacidade de armazenamento de dados não estruturados ou semiestruturados, possuem alta escalabilidade e vem sendo utilizado com frequência para aplicações web.

É possível citar como características principais: escalabilidade horizontal, na medida em que o volume de dados cresce, necessita-se de escalabilidade, ou seja, por não existir controle de bloqueios, é possível escalar dados com uma maior facilidade e eficiência.

Ausência de esquema ou esquema flexível, ou seja, é a ausência parcial ou completa do esquema que define a estrutura dos dados. Esta ausência facilita tanto a escalabilidade quanto contribui para uma alta disponibilidade, temos também a chamada consistência eventual, esta característica tem como princípio o teorema CAP (Consistência, disponibilidade e tolerância a particionamento), que afirma que, em um dado momento, apenas pode-se garantir duas das três propriedades entre consistência, disponibilidade e tolerância à partição. [Lóscio; Oliveira; Pontes 2011].

Atualmente os bancos de dados NoSQL são divididos em quatro classes principais, cada um com um tipo específico de armazenamento que são: chave-valor, orientado a documentos, orientado em colunas e baseado em grafos [Kokay 2012]. Para a fundamentação teórica do trabalho serão utilizadas as bases de dados MongoDB e CouchDB, já que são as mais representativas no mercado e consideradas como sendo as mais completas da família de bases de dados NoSQL do tipo documento.

Os bancos orientados a documentos armazenam as informações sobre uma estrutura de árvore, em formatos como Linguagem Extensível de Marcação, do inglês, *Extensible Markup Language* (XML) ou Notação de Objetos JavaScript, do inglês, JavaScript Object Notation (JSON). São documentos previamente definidos, tendo como características o armazenamento de estruturas auto descritivas e autocontidas, apresentando uma similaridade uns com os outros, mas não têm de ser exatamente a mesma formatação.

O suporte ao processamento e análise de dados é bastante variado dentre os representantes da categoria. Uma característica interessante é o método *append-only*, que é utilizado em alguns bancos de documentos para armazenar os dados e oferece operações de escrita em tempo constante. Dentro dessa estrutura, as chaves devem ser únicas. Para cada documento existe um identificador chamado ID que também deve ser único e serve para identificar o documento na coleção.

3. MongoDB

O MongoDB possui código aberto e possui alta performance, armazena dados em forma de documentos *JavaScript Object Notation* (JSON), trata-se de um formato leve para troca e armazenamento de dados computacionais, é um subconjunto da notação de objeto de JavaScript, mas seu uso não requer JavaScript exclusivamente [Plugge et al. 2010]. Este utiliza um Sistema de gerenciamento de banco de dados NoSQL do tipo *Document Store*, que permite o agrupamento de todas as informações de uma determinada coleção em um único documento, essas coleções podem ser vistas como um equivalente às tabelas nos bancos de dados relacionais.

Outra característica da estrutura de documentos é a utilização de redundância (cópias de segurança) com a finalidade de atenuar o problema da inconsistência, proveniente da distribuição dos registros em clusters de computadores, ou seja, quanto maior o número de computadores maior é a probabilidade de falha em algum ponto, mas caso haja alguma falha o impacto é menor, sendo mais passível a recuperação.

Para o aumento da performance, o modelo orientado a documento utiliza a ferramenta *MapReduce* que a partir de métodos avançados possibilita uma maior eficiência no agrupamento e filtragem dos dados possibilitando a leitura /escrita dos dados de forma paralela, uma vez que o banco de dados não possui esquemas. Se um documento precisar incluir um novo campo pode simplesmente incluir esse campo, sem afetar de forma adversa outros documentos do banco de dados.

Além disso, os documentos não precisam armazenar valores de dados vazios para campos para os quais não possuem um valor. Os documentos são identificados pelo campo *_id*, que pode ser gerado automaticamente pelo banco ou definido pelo usuário [Silva 2007].

Como ilustrado na Figura 1, mostra-se uma comparação da estrutura de bancos de dados relacionais e não relacionais, ambos possuem base de dados, as tabelas são chamadas de coleções, as linhas e colunas são respectivamente documentos e campos.

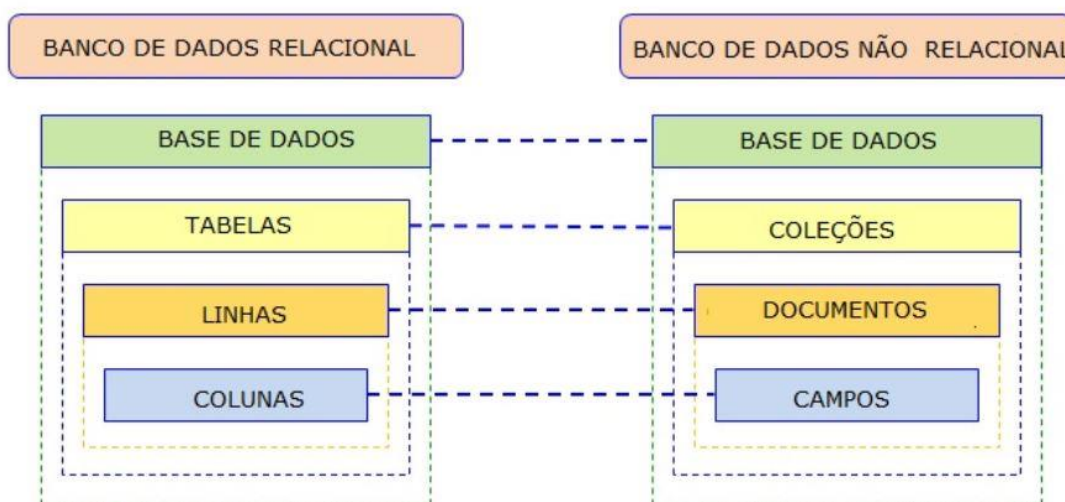


Figura 1. Comparativo entre estrutura SQL e MongoDB. Fonte: Os Autores.

3.1 Características

Segundo Brito (2010), o MongoDB possui algumas características como Banco de dados orientado a documentos (todos os dados para o documento em questão estão armazenados no próprio documento), tais como: código aberto, escrito em C++,suporte a replicação, armazena vídeos e fotos com a ferramenta GridFs, suporte à computação em nuvem, suporta consultas dinâmicas (ad hoc),suporta expressões regulares, comparadores, operadores condicionais e operadores lógicos.

Além das características, cita as seguintes vantagens: abstração de objetos do mundo real como realmente são: complexos e únicos, caso seja necessário adicionar um novo campo, pode adicionar a um único documento e não a todos os documentos como no caso de uma tabela no modelo relacional.

4. CouchDB

Para Lennon (2009) o CouchDB é um sistema de software livre de gerenciamento de banco de dados orientado a documentos que pode ser acessado através da API de JavaScript Object Notation (JSON) do tipo RESTful.

A base de dados Apache CouchDB é também uma Document-based database open-source. O termo "Couch" é um acrônimo para "Cluster Of Unreliable Commodity Hardware", que reflete a meta do CouchDB de ser.

O CouchDB foi criado pela Fundação Apache e desenvolvida em Erlang, uma linguagem de programação utilizada para aplicações concorrentes e distribuídas devido à sua ênfase em tolerância a falhas. Possui como princípio arquitetural a Web. Atualmente o CouchDB conta com drivers para várias linguagens de programação, sendo elas: C, C++, C#, Java, Perl, PHP, Python, Ruby e Scala.

O modelo relacional necessita que suas estruturas estejam devidamente modeladas antes de serem usados, já no Couch não há essa necessidade, pois, o documento já está pronto para receber qualquer estrutura, porém permite redundância e consistência eventual dos dados.

As propriedades ACID são cumpridas nesta base de dados, fornecendo atualizações serializadas. É já, pela sua natureza, distribuída e suporta vários tipos de esquemas de replicação e resolução de conflitos.

Dentre as principais características que tornam o CouchDB um banco de dados vantajoso em diferentes situações, pode-se listar: distribuído, descentralizado, altamente escalável, alta disponibilidade e confiabilidade, tolerante a falhas, consistência e alta performance.

Esse sistema disponibiliza uma série de características que permitem que servidores com hardware de baixo desempenho consigam funcionar sem nenhum problema.

4.1 Modelo de dados

Em relação ao seu modelo, o CouchDB possui licença gratuita, é formado por uma coleção independente de documentos em formato JSON, que é um modelo para

armazenamento e transmissão de informações no formato texto [Queiroz et al. 2013]. Os valores podem ser de vários tipos: números, strings, booleanos, listas e dicionários. Os documentos são livres de esquema, a cada documento são atribuídos pelo menos dois atributos: um número de revisão (`_rev`), necessário para a resolução de conflitos durante atualizações, e um identificador único (`_id`). Na Figura 2 está representado um exemplo de uma descrição de livro na estrutura CouchDB.

```
1.  {
2.    "_id":"282e1890a809bcc0cb1318bed85bcb377e2700f",
3.    "_rev":"946B7D1C",
4.    "Type":"Book",
5.    "Title":"Flatland",
6.    "Author":"Edwin A. Abbot",
7.    "Date":"2009-10-09",
8.    "Language":"English",
9.    "ISBN":"1449548660",
10.   "Tags":["flatland","mathematics","geometry"],
11. }
```

Figura 2. Descrição de um livro. Fonte:Silva,2007.

Esta base de dados é um simples contentor de uma coleção de documentos e não estabelece nenhuma relação obrigatória entre eles. Embora o CouchDB não imponha nenhum tipo de relação entre os documentos, é importante possuir algum tipo de estrutura que retrate como os documentos são organizados. Sendo assim, o CouchDB fornece um *view model*. [Silva 2007].

5. Materiais e métodos

Dividiu-se os testes em 4 categorias: inserção, consulta, atualização e exclusão. Foram executados em loops de 1.000, 10.000 e 100.000 registros, os testes foram repetidos três vezes cada, criando assim uma média aritmética, a qual permitiu a comparação dos resultados obtidos de cada SGBD. O objetivo é analisar o desempenho pelo tempo de execução e verificar qual obteve o menor tempo de resposta para as ações citadas.

Os testes foram realizados em um equipamento computacional com a seguinte configuração: Notebook Dell, que possui as seguintes configurações: processador Intel Core i5-7200U CPU de 2.5 GHz com 8 GB de RAM e SSD de 128 GB com o sistema operacional Windows 10 Pro de 64 bits.

A figura 3 descreve as etapas da execução dos testes.

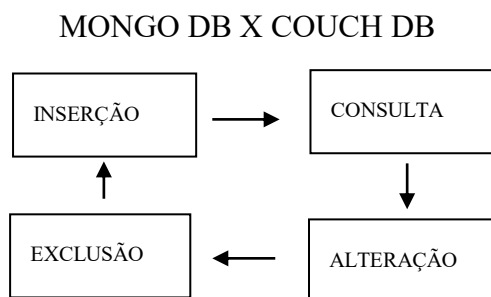


Figura 3. Descrição metodológica. Fonte: Elaborado pelos próprios autores.

5.1 Desenvolvimento

Foi utilizada a versão MongoDB 4.0.9 e a versão 2.3.1 do CouchDB instalada diretamente na plataforma Windows 10, 64 bits. Para o CouchDB, utilizou-se também a ferramenta *Curl*, trata-se de uma API tipo RESTful HTTP que interage diretamente por linha de comando.

Criou-se o banco de dados de nome "banco", referindo-se a dados de uma conta bancária de determinado cliente de uma agência, a coleção foi chamada de "usuário1" e contém os seguintes atributos: número da conta, data da abertura da conta, código do banco, nome do banco, código da agência, nome da agência, nome do cliente, data de nascimento, CPF, telefone, sexo, lançamento, data dos lançamentos, descrição e valor.

Por se tratar de um grande volume de registros a serem incluídos, utilizou-se a estrutura de repetição, chamada FOR.

Segue arquivo JSON utilizado.

```
1. {
2.   "conta": {
3.     "numero_conta": "3869-5",
4.     "dt_abertura": "2019-01-01",
5.     "banco": {
6.       "codigo_banco": "001",
7.       "nome_banco": "Banco do Brasil"
8.     },
9.     "agencia": {
10.      "codigo_agencia": "2030",
11.      "nome_agencia": "Centro"
12.    },
13.    "cliente": {
14.      "nome_cliente": "Claudia",
15.      "dt_nascimento": "1951-02-07",
16.      "cpf": "00000000001",
17.      "fone": "1499993322",
18.      "sexo": "Feminino"    },
19.      "lista_lancamento": {
20.        "lançamento": {
21.          "data": "2019-01-01 T08:00:00 ",
22.          "descricao": "padaria",
23.          "valor": "- 100"
24.        }
25.      }
26.    }
27.  })
28. }
```

Utilizou-se no MongoDB o comando *profile*, que gera um arquivo de log com extensão .txt permitindo assim verificar o tempo de execução de cada operação e para o CouchDB utilizou-se o comando *explain* em conjunto com a tag *execution_time_ms*. Para a realização das inserções, consultas, atualizações, exclusões, foram utilizados os comandos *insert*, *find*, *update* e *remove* no MongoDB e para o CouchDB, *post*, *get*, *put* e *delete* respectivamente.

5.2 Resultados

São descritos nesta seção os resultados obtidos através da realização dos testes, que através das operações de inserção, consulta, atualização e exclusão, retornaram o tempo de execução de cada ação.

Os testes medem a diferença entre os tempos de execução de todas essas operações nos SGBDs, com a finalidade de evidenciar a diferença de tempo para cada execução das ações citadas anteriormente. A seguir são mostrados os resultados obtidos com o SGBD MongoDB através das tabelas 1, 2, 3 e 4.

Tabela 1. Resultados da Inserção com MongoDB e CouchDB em segundos

MongoDB				CouchDB			
N. de Inserção	1.000	10.000	100.000	N.de Inserção	1.000	10.000	100.000
1ª Execução	0,88	6,32	65,95	1ª Execução	2,84	8,02	68,27
2ª Execução	0,89	7,38	66,96	2ª Execução	2,85	8,03	69,27
3ª Execução	0,85	7,09	67,36	3ª Execução	2,88	8,02	69,30
Média	0,87	6,93	66,75	Média	2,86	8,02	68,94

Fonte: Os autores (2019).

Na Tabela 1, apresenta-se os dados obtidos através da média de inserções onde os valores estão representados em segundos, o MongoDB apresenta resultado superior ao obtido com o CouchDB. Enquanto o MongoDB leva 0,88 segundos para inserir 1.000 registros no banco de dados, o CouchDB leva 2,84 segundos em média para fazer a mesma operação. O MongoDB continua superior ao CouchDB inserindo 10.000, 100.000 registros.

Tabela 2. Resultados da Consulta com MongoDB e CouchDB em segundos

MongoDB				CouchDB			
N. de Inserção	1.000	10.000	100.000	N.de Inserção	1.000	10.000	100.000
1ª Execução	0,32	0,45	0,58	1ª Execução	2,60	2,86	4,30
2ª Execução	0,25	0,49	0,38	2ª Execução	2,60	2,85	4,25
3ª Execução	0,31	0,45	0,36	3ª Execução	2,63	2,85	4,26
Média	0,29	0,46	0,44	Média	2,61	2,85	4,27

Fonte: Os autores (2019).

Na tabela 2, os resultados obtidos nas médias de consulta em segundos, o MongoDB se mantém superior ao CouchDB com os seguintes resultados: a consulta com 10.000 registros, resultou na média de 0,46 e o CouchDB levou 2,85 segundos para a efetuar a consulta nas mesmas condições.

Tabela 3. Resultados da Alteração com MongoDB e CouchDB em segundos

MongoDB				CouchDB			
N. de Alteração	1.000	10.000	100.000	N. de Alteração	1.000	10.000	100.000

1ª Execução	0,39	2,73	37,62	1ª Execução	2,54	3,01	43,17
2ª Execução	0,36	2,68	36,58	2ª Execução	2,79	3,13	43,09
3ª Execução	0,35	2,54	36,32	3ª Execução	2,62	3,54	43,08
Média	0,36	2,65	36,84	Média	2,65	3,22	43,11

Fonte: Os autores (2019).

Na tabela 3, observa-se que a média de alteração de registros no MongoDB com 1.000 registros é igual a média do CouchDB, porém com 10.000 registros.

Tabela 4. Resultados da Exclusão com MongoDB e CouchDB em segundos

MongoDB				CouchDB			
N. de Alteração	1.000	10.000	100.000	N. de Alteração	1.000	10.000	100.000
1ª Execução	0,25	0,60	1,92	1ª Execução	2,80	3,01	4,10
2ª Execução	0,36	0,66	2,12	2ª Execução	2,82	3,27	4,19
3ª Execução	0,26	0,43	2,13	3ª Execução	2,79	3,13	4,11
Média	0,29	0,56	2,05	Média	2,80	3,13	4,13

Fonte: Os autores (2019).

Na tabela 4, com os resultados obtidos através da exclusão de registros é possível observar que o MongoDB supera o CouchDB em todas as quantidades de registros e em tempo de execução, tendo o MongoDB tempo de execução relativamente menor se comparado ao CouchDB.

6. Conclusão

A comparação entre os dois SGBDs não-relacionais MongoDB e CouchDB permitiu verificar quanto tempo cada banco de dados leva para inserir, consultar, alterar e excluir os dados do banco. Com os resultados obtidos é possível concluir que o MongoDB apresentou desempenho consideravelmente superior se comparado ao CouchDB, destacou-se com o menor tempo necessário para realização de qualquer operação de manipulação de dados. Nota-se que em todos os testes efetuados, o tempo de execução do MongoDB teve resultados relativamente superiores se comparados ao CouchDB, é possível dizer que o SGBD MongoDB é o mais indicado para aplicações em que necessite de fácil escalabilidade e alta performance.

Para trabalhos futuros, as pesquisas incluem a realização dos mesmos testes utilizando bases de dados com tamanhos mais diferenciados (1GB, 10GB e 100GB, por exemplo). Pode-se abordar a comparação dos resultados apresentados com os de outros bancos não relacionais da estrutura de documentos, incrementar buscas utilizando outros critérios bem como novas tabelas, a realização de testes de backup e de replicação.

7.Referências

- Abramova, V., Bernardino, J., e Furtado, P. (2014). “Which nosql database? a performance overview”. Open Journal of Databases.
- Brito, R.W. (2010). Bancos de dados no sql x sgbds relacionais: análise comparativa.
- Fowler,A.(2015). NoSQLForDummies. JohnWiley&Sons,111 RiverStreet,Hoboken, New Jersey, USA.
- Kokay, M.C. (2012): “Banco de dados NoSQL: Um novo paradigma – Revista SQL Magazine 102”.
- Lennon,J.(2009)<https://www.ibm.com/developerworks/br/library/os-couchdb/index.html>,Outubro.
- Lóscio, B. F., Oliveira, H. R. e Pontes, J.C. NoSQL no desenvolvimento de aplicações Web Colaborativas.SBSC,2011.
- Nascimento, J. (2010) “NoSQL- você realmente sabe do que estamos falando?” [http:Plugge/imasters.com.br/17043/banco-de-dados-/NoSQL-voce-realmente-sabe-do-que-estamos-falando](http://Plugge/imasters.com.br/17043/banco-de-dados-/NoSQL-voce-realmente-sabe-do-que-estamos-falando), Outubro.
- Plugge et al. (2010). The Definitive Guide to MongoDB: The NoSQL Database Cloud and Desktop Computing,Apress.
- Queiroz G. R. De, Miguel, A., Monteiro, V., & Câmara, G. (2013). “Geographic Databases and NoSQL: Accinokishments and Future Directions”, Revista Brasileira de Cartografia, 479 – 492.
- Silva, N. F. F. (2007). Armazenamento de Dados XML: Técnicas de Benchmark para avaliação.151f. Monografia (Graduação em Ciência da Computação) - UFG. Catalão, GO. [www.catalao.ufg.br/cc/disc/pfc/mono/Na dia2007.pdf](http://www.catalao.ufg.br/cc/disc/pfc/mono/Na%20dia2007.pdf), Outubro.