

COMPARATIVO ENTRE OS MODELOS DE BANCO DE DADOS RELACIONAL E NÃO-RELACIONAL

COMPARISON BETWEEN RELATIONAL AND NONRELATIONAL DATA BANK MODELS

Vinícius Salles Garcia – vinicius.s_garcia@outlook.com

Eder Carlos Salazar Sotto – eder.sotto@fatec.sp.gov.br

Faculdade de Tecnologia de Catanduva (FATEC) –SP –Brasil

DOI: 10.31510/infa.v16i2.673

RESUMO

O presente artigo consiste em uma pesquisa bibliográfica exploratória e descritiva que visa informar uma nova área da tecnologia de banco de dados, que emergiu como resposta às dificuldades pelo tradicional modelo relacional do banco de dados ao crescimento exponencial de dados gerados, armazenados e processados atualmente com o surgimento da *Web 2.0*. Neste contexto, surge um banco de dados de modelo não-relacional, denominada de NoSQL (*Not Only Structured Query Language*), proposto com o objetivo de atender aos requisitos de gerenciamento de grandes volumes de dados, semiestruturados ou não-estruturados, com alta escalabilidade e de baixo custo, comparando-se ao custo de escalabilidade de um banco de dados de modelo relacional. O objetivo principal é apresentar os sistemas de gerenciamentos de dados não-relacionais como opção para uso de banco de dados para aplicações que demandem por alta escalabilidade, sendo mais voltado para uso com aplicações *Web* e aplicativos. Espera-se com este artigo, auxiliar a compreensão sobre o modelo não-relacional, para entendimento de seu funcionamento e características, que contribuem para uma nova inovação como atendimento de necessidade.

Palavras-chave: NoSQL. Não-Relacional. Escalabilidade.

ABSTRACT

This article is an exploratory and descriptive bibliographic research that aims to inform a new area of database technology that emerged as a response to the difficulties of the traditional relational model of the database to the exponential growth of data generated, stored and processes currently with the appearance of web 2.0. In this context, a non-relational model database, known as NoSQL (Not Only Structured Query Language), is developed to meet the requirements of managing large volumes of data, semi-structured or unstructured, with high scalability, and low cost, compared to the cost of scalability of a relational model database. The main objective is to present non-relational data management systems as an option for database use for applications that demand high scalability, being more oriented towards use with WEB applications and applications. This article is intended to give a better understanding about the non-relational model, to understand its operation and characteristics, which contribute to a new innovation as a service of necessity.

Keywords: NoSQL. Non-Relational. Scalability.

1 INTRODUÇÃO

Com o desenvolvimento acelerado da internet, empresas perceberam uma nova oportunidade de negócios, e como tal, as aplicações migraram de sistemas fechados para a internet, sob forma de *websites*. De forma a organizar os *websites* que surgem, foi criado o motor de busca, como o yahoo e o google, que por sua vez guarda informações da página, por meio de palavras-chaves, referências externas e metadados. Logo após surgiram as aplicações denominadas redes sociais, que geram e mantêm um enorme número de utilizadores e cada um deles com diversas informações sobre si mesmo, por exemplo, informações básicas, fotos, vídeos e ligações, e todas essas informações geram um grande volume de dados. Para armazenar toda a informação é necessário ter um banco de dados, normalmente, os bancos de dados relacionais são usados para esse fim, no entanto o número de *sites* que aparecem por dia é elevado.

1.1 Sistemas Gerenciadores de Banco de Dados

O Sistema Gerenciador de banco de dados (SGBD) ou *Data Base Management System* (DBMS) é um software utilizado para administrar um banco de dados, permitindo criar, modificar, eliminar e inserir dados no mesmo.

Os primeiros Sistemas Gerenciadores de Banco de Dados (SGBD), segundo Lóscio *et al.* (2011, p.2)

[...] surgiram por volta de 1960 e foram desenvolvidos com base nos primeiros sistemas de arquivos. Dentre as principais características de SGBD, destacamos: controle de ocorrência, segurança, recuperação de falhas, gerenciamento do mecanismo de armazenamento de dados e controle das restrições de integridade do BD.

Outra importante função de um SGBD é o gerenciamento de transações. Uma transação pode ser definida, segundo Lóscio *et al.* (2011, p.2), como uma coleção de operações que desempenha uma função lógica dentro de uma aplicação do sistema de banco de dados, em outras palavras, uma transação representa um conjunto de operações de leitura ou escrita que são realizadas no banco de dados. A execução de transações em SGBD deve obedecer a algumas propriedades, a fim de garantir o correto funcionamento do sistema e a respectiva consistência dos dados.

Essas propriedades são chamadas de propriedades ACID, na qual se refere às quatro propriedades de transação de um sistema de banco de dados: Atomicidade, Consistência,

Isolamento e Durabilidade (LÓSCIO *et al.*, 2011).

- **Atomicidade:** Em uma transação envolvendo duas ou mais partes de informações discretas, ou a transação será executada totalmente ou não será executada, garantindo assim que as transações sejam atômicas.

- **Consistência:** A transação cria um estado válido dos dados ou em caso de falha retorna todos os dados ao seu estado antes que a transação foi iniciada.

- **Isolamento:** Uma transação em andamento, mas ainda não válida deve permanecer isolada de qualquer outra operação, ou seja, garantimos que a transação não será interferida por nenhuma outra transação.

- **Durabilidade:** Dados validados são registrados pelo sistema de tal forma que mesmo no caso de uma falha e/ou reinício do sistema, os dados estão disponíveis em seu estado correto.

1.2 Modelo de Dados Relacional

Apoiando a estrutura de um banco de dados está o modelo de dados: uma coleção de ferramentas conceituais para descrever dados, relações de dados, semântica de dados e restrições de consistência (SILBERSCHATZ *et al.*, 2016).

“O modelo relacional foi proposto por Edgar Codd em 1970, como uma nova maneira de representação de dados” (MACÁRIO; BALDO, 2005, p.1).

1.3 Limitações do Modelo Relacional

Com o crescimento e popularização da internet, as exigências cobradas de um banco de dados mudaram. Vários domínios de aplicações como, *Web*, redes sociais, entre diversos outros, geram algumas dezenas, ou centenas de *Terabytes* de dados diariamente (VIEIRA *et al.*, 2012).

Pode-se afirmar que os principais problemas encontrados no modelo relacional, após o crescimento exponencial de dados, estão principalmente relacionados na dificuldade de conciliar o tipo de modelo com a demanda da escalabilidade.

2 BANCOS DE DADOS NÃO-RELACIONAIS

Com a necessidade de uma nova tecnologia de banco de dados para a manipulação de

grandes volumes de dados não estruturados e semiestruturados, bem como novas necessidades de disponibilidade e escalabilidade, surgiu o NoSQL (*Not only SQL*) como uma proposta para a solução de gerenciamento de dados (LÓSCIO *et al.*, 2011).

O termo NoSQL foi primeiramente utilizado em 1998 com o nome de *open source nonrelational database*. Seu autor, Carlo Strozzi, alega que o movimento NoSQL é completamente distinto do modelo relacional. Após isso, em 2009, o termo NoSQL é reintroduzido por um funcionário do Rackspace, Eric Evans, durante a organização de um evento sobre bancos de dados *open source* distribuídos. Desde então, NoSQL passou a ser a palavra utilizada para descrever o novo modelo de banco de dados (IANNI, 2012).

Os bancos de dados NoSQL foram desenvolvidos para serem fáceis de distribuir, isso implica um diferente modelo de consistência em contrapartida ao comum ACID (CARDOSO, 2012).

2.1 Principais Características do SGBD NoSQL

De acordo com Lóscio *et al.* (2011, p.4) "Os bancos de dados NoSQL apresentam algumas características fundamentais que os diferenciam dos tradicionais sistemas de bancos de dados relacionais, tornando-os adequados para armazenamento de grandes volumes de dados não-estruturados ou semiestruturados." Algumas das características são:

- **Escalabilidade Horizontal.** A ausência de bloqueios é uma característica fundamental dos bancos de dados NoSQL, permitindo a escalabilidade horizontal e tornando esta tecnologia adequada para solucionar os problemas de gerenciamento de volumes de dados que crescem exponencialmente, como os dados da Web 2.0 (LÓSCIO *et al.*, 2011).

- **Ausência de esquema ou esquema flexível.** Uma característica evidente dos bancos de dados NoSQL é a ausência completa ou quase total do esquema que define a estrutura dos dados modelados. Esta ausência de esquema facilita tanto a escalabilidade quanto contribui para um maior aumento da disponibilidade. Em contrapartida, não há garantia da integridade dos dados, o que ocorre nos bancos relacionais, devido à sua estrutura rígida (LÓSCIO *et al.*, 2011).

- **Suporte nativo a replicação.** Permite a replicação de forma nativa e diminui o tempo gasto para recuperar informações. Existem duas abordagens principais para replicação: *Master-Slave* (Mestre-Escravo) e *Multi-Master* (LÓSCIO *et al.*, 2011).

- **API simples para acesso aos dados.** O objetivo da solução NoSQL é prover uma forma

eficiente de acesso aos dados, oferecendo alta disponibilidade e escalabilidade, ou seja, o foco não está em como os dados são armazenados e sim como poderemos recuperá-los de forma eficiente. Para isto, é necessário que APIs sejam desenvolvidas para facilitar o acesso a estas informações, permitindo que qualquer aplicação possa utilizar os dados do banco de forma rápida e eficiente (LÓSCIO *et al.*, 2011).

- **Consistência eventual.** É uma característica de bancos NoSQL relacionada ao fato da consistência nem sempre ser mantida entre os diversos pontos de distribuição de dados. Esta característica tem como princípio o teorema CAP (*Consistency, Availability e Partition tolerance*), na qual diz, que em um dado momento, só é possível garantir duas de três propriedades entre consistência, disponibilidade e tolerância à partição (LÓSCIO *et al.*, 2011).

- **O *Map/Reduce*** dá suporte ao manuseio de grandes volumes de dados distribuídos ao longo dos nós de uma rede. Na fase de *map*, os problemas são quebrados em subproblemas que são distribuídos em outros nós na rede. E na fase *reduce*, os subproblemas são resolvidos em cada nó filho e o resultado é repassado ao pai, que, sendo ele também filho, repassaria ao seu pai, e assim por diante até chegar ao nó raiz do problema (LÓSCIO *et al.*, 2011).

- **O *consistent hashing*** suporta mecanismos de armazenamento e recuperação em banco de dados distribuído, onde a quantidade de *sites* está em constante modificação. O uso desta técnica é interessante, pois evita muita migração de dados entre esses sites, os quais podem ser alocados e desalocados para a distribuição dos dados (LÓSCIO *et al.*, 2011).

- **O *multiversion concurrency control (MVCC)*** é um mecanismo que dá suporte a transações paralelas em um banco de dados. Ao contrário do esquema clássico de gerenciamento de transações, como não faz uso de *locks*, permite que operações de leitura e escrita sejam feitas simultaneamente (LÓSCIO *et al.*, 2011).

- ***Vector clocks*** são usados para gerar uma ordenação dos eventos acontecidos em um sistema. Pela possibilidade de várias operações estarem acontecendo ao mesmo tempo sobre o mesmo item de dado distribuído, o uso de um *log* de operações com as suas respectivas datas é importante para determinar qual a versão de um determinado dado é a mais atual (LÓSCIO *et al.*, 2011).

3 PROCEDIMENTOS METODOLÓGICOS

A metodologia de pesquisa deste artigo consiste em, como base, levantamentos bibliográficos exploratórios e descritivos, análise de artigos relacionados ao tema abordado e notícias atuais sobre TI.

De acordo com Marconi e Lakatos (2007), a pesquisa bibliográfica, abrange toda bibliografia já tornada publica em relação ao tema de estudo, desde publicação avulsas, como, boletins, jornais, revistas, livros, pesquisas, monografias, tese, material cartográfico etc., até meios orais, como, rádio, gravações e audiovisuais, como, filmes e televisão.

E complementam, “sua finalidade é colocar o pesquisador em contato direto com tudo o que foi escrito, dito ou filmado sobre determinado assunto.” (MARCONI; LAKATOS, 2007).

4 PRINCIPAIS DIFERENÇAS ENTRE SGBDs RELACIONAIS E NoSQL

No momento em que se pensa na possibilidade de utilizar um banco de dados NoSQL ao invés de um modelo relacional, é preciso levar em consideração alguns critérios de escalonamento, consistência e disponibilidade de dados.

A escalabilidade é o principal aspecto dos bancos de dados NoSQL, na qual possuem uma grande vantagem em relação ao SGBD tradicional, basicamente por terem sido criados por essa finalidade.

Os bancos de dados relacionais possuem uma estruturação que não a permite tanta flexibilidade, por ter seus dados armazenados em tabelas e ligados entre si por relacionamentos, além disso, o modelo relacional é menos adaptada para situações em que o escalonamento se faz necessário, e quando escalonado o custo é bem alto, por necessitar de várias máquinas para ser distribuído o banco de dados relacional e como consequência aumentando a complexidade devido à dificuldade de adaptar toda estrutura lógica do modelo relacional (IANNI, 2012).

A disponibilidade do modelo NoSQL em relação ao modelo relacional, pelo fato de se houver queda do sistema em uma máquina não irá causar interrupção dele, IANNI (2012) complementa, “podemos citar um evento que ocorreu em 2008, em que a rede social Twitter ficou fora do ar durante 84 horas, neste evento o Twitter ainda utilizava o PostgreSQL. A partir de 2009, quando começou a utilizar o Cassandra, outro evento similar ocorreu, porém, o site ficou fora dor ar durante 23 horas e 45 minutos”.

Ao pensar em substituir um banco de dados relacional pelo NoSQL, deve-se levar em consideração algumas observações, como a falta das propriedades ACID (RABELO; CÂNDIDO, 2017, p.44).

O Quadro 1 exemplifica de forma concisa a comparação do banco de dados relacional e o NoSQL em relação a escalonamento, consistência e disponibilidade.

Quadro 1 – Análise comparativa entre Banco de Dados Relacional e NoSQL

	Banco de dados relacional	Banco de dados NoSQL
Escalonamento	É importante lembrar que é possível ser feito o escalonamento em um Modelo Relacional, no entanto, é muito complexo. Possui uma natureza estruturada, portanto, a inserção dinâmica e transparente de novos nós a tabela não é realizada naturalmente.	Não possui um esquema pré-definido fazendo com que este tipo de modelo seja flexível o que favorece a inserção transparente de outros elementos.
Consistência	Neste quesito, o Modelo Relacional se mostra forte. As suas regras de consistência são bastante rigorosas no que diz respeito à consistência das informações.	É realizada eventualmente no modelo: tem apenas a garantia que se não houver nenhuma atualização nos dados, todos os acessos aos itens devolverão o último valor que foi atualizado.
Disponibilidade	Por não conseguir trabalhar de forma eficiente com a distribuição de dados, o Modelo Relacional acaba não suportando uma demanda muito grande de informações.	Outro ponto forte neste modelo é o que diz respeito à disponibilidade, pois possui um alto nível de distribuição de dados, permitindo assim que seja possível fazer com que um enorme fluxo de solicitações aos dados seja atendido com a vantagem de o sistema ficar indisponível o menor tempo possível.

Fonte: Kokay, M. C (2012)

O Quadro 2 apresenta outras diferenças entre banco de dados relacional e o NoSQL em relação a cargas de trabalho, modelo de dados, performance e APIs.

Quadro 2 – Comparação entre Banco de Dados Relacional e NoSQL

	Banco de dados relacional	Banco de dados NoSQL
Cargas de trabalho ideais	Bancos de dados relacionais são projetados para aplicativos transacionais e fortemente consistentes de processamento de transações online (OLTP) e são bons para processamento analítico online (OLAP).	Os bancos de dados de chave-valor, documento, gráfico e em memória do NoSQL são projetados para OLTP para vários padrões de acesso aos dados que incluem aplicativos de baixa latência. Os bancos de dados de pesquisa do NoSQL são projetados para análise de dados semiestruturados.
Modelo de dados	O modelo relacional normaliza dados em tabelas, compostas por linhas e colunas. Um esquema define estritamente tabelas, colunas, índices, relações entre tabelas e outros elementos do banco de dados. O banco de dados impõe a integridade referencial nos relacionamentos entre as	Os bancos de dados NoSQL fornecem diversos modelos de dados que incluem documento, gráfico, chave-valor, em memória e pesquisa.

	tabelas.	
Performance	A performance normalmente depende do subsistema do disco. A otimização de consultas, índices e estrutura de tabela é necessária para alcançar máxima performance.	A performance geralmente é uma função do tamanho do cluster do hardware subjacente, da latência de rede e do aplicativo que faz a chamada.
APIs	As solicitações para armazenar e recuperar dados são comunicadas usando consultas compatíveis com uma Structured Query Language (SQL – Linguagem de consultas estruturadas). Essas consultas são analisadas e executadas pelo banco de dados relacional.	APIs baseadas em objetos permitem que desenvolvedores de aplicações armazenem e restaurem facilmente estruturas de dados na memória. As chaves de partição permitem que os aplicativos procurem pares de chave-valor, conjuntos de colunas ou documentos semiestruturados que contenham objetos e atributos de aplicativos serializados.

Fonte: Amazon (2019)

4.1 Análise de Desempenho do Banco de Dados Relacional e NoSQL

A crescente necessidade de armazenar grande volumes de dados verifica-se a necessidade de utilização de SGBDs melhores e mais eficientes. De acordo com Carneiro *et al.* (2010), “realizar operações, de forma eficiente, sobre estas grandes coleções de dados é uma questão fundamental, já que um SGBD é medido a partir de sua eficiência diante de consultas e alterações”.

Para a análise de desempenho do banco de dados relacional e NoSQL desde artigo, foi utilizado como base para comparação o Trabalho de Conclusão de Curso de Rabelo e Cândido (2017), na qual eles abordam e testam a implementação e o desempenho de um banco de dados NoSQL, na qual utilizam o MongoDB, comparando-o com um banco de dados relacional, na qual utilizam o PostgreSQL, em uma empresa de contábeis:

A fim de averiguar o tempo de resposta de cada um dos bancos de dados, foi executado um teste de carga nas funcionalidades de importar/visualizar plano de contas, importar balancete e geração de relatórios (balanço patrimonial, demonstração de fluxos de caixa e demonstrações das mutações do patrimônio líquido). Foram feitas três rodadas de testes, em ambos os bancos de dados com os mesmos arquivos, onde o primeiro leu um arquivo de 1674 registros, o segundo 10.000 registros e o último 100.000 registros (RABELO; CÂNDIDO, 2017, p.51).

Segundo os mesmos autores, os testes executados tiveram como ambiente de execução um notebook Dell Inspiron 15R com processador core i7 2,9GHz com 8GB de memória RAM, 1TB de HD, 8GB de SSD e placa de vídeo dedicada GTX force 2GB, com sistema operacional Windows 10

Pró. Todos os testes realizados foram efetuados em banco de dados local.

Após os testes foi levantado um *checklist* de avaliação de desempenho de um banco de dados NoSQL, para que posteriormente, segundo eles, este mesmo *checklist* fosse utilizado na aplicação desenvolvida para que o banco de dados pudesse ser analisado.

O Quadro 2 mostra o Resultado do *checklist* após os testes feito pelos autores Rabelo e Cândido (2017).

Quadro 2 – Resultado do Checklist de Análise de Desempenho

Pergunta	Conclusão
O tempo de resposta do banco de dados não relacional foi superior ao relacional?	Sim, em todos os cenários de testes propostos o banco de dados não relacional obteve um desempenho superior ao relacional.
O banco de dados não relacional conseguiu manter um nível aceitável de performance à medida que os dados iam aumentando?	Sim. Foi realizado testes com três quantidades de dados no banco: 1674, 10000 e 100000, em todos os casos, o banco de dados obteve um tempo de resposta aceitável, não passando de 14 segundos.
A falta das propriedades ACID no banco de dados não relacional teve um impacto negativo no software?	Sim, a falta de propriedades ACID como por exemplo a atomicidade faz com que não garanta a integridade dos dados no banco de dados. Na aplicação, a importação do plano de contas e balancete é de vital importância para manter a confiabilidade dos relatórios emitidos. No teste onde a comunicação com o banco de dados foi perdida na importação do plano de contas fez com que apenas 5 registros do plano de contas fosse armazenado no banco de dados, logo se o usuário não percebesse que os dados estavam incompletos, os relatórios emitidos pelo sistema estariam incorretos. Porém, este problema foi facilmente tratado através de uma validação que apresentará uma mensagem ao usuário quando este problema acontecer.
O sistema se tornou indisponível durante a execução de alguma funcionalidade?	Não, o banco de dados permaneceu acessível a outros usuários durante a execução das funcionalidades deles.
O sistema suportou dois ou mais usuários concorrendo pelos mesmos dados?	Sim, foi realizado teste onde dois usuários de forma simultânea solicitaram visualizar um plano de contas. O sistema conseguiu dispor dados para ambos os usuários.

Fonte: RABELO; CÂNDIDO (2017)

5 CONSIDERAÇÕES FINAIS

O modelo relacional é um dos modelos mais usados hoje em dia. Desde que foi inventado, é a solução mais amplamente adotada em termos de persistência. Tem como vantagens ser um modelo robusto e fiável, além de ser um modelo bastante testado e com provas que fornece garantias de atomicidade, consistência, isolamento e durabilidade.

No entanto, com o crescimento de volume de dados nas organizações, impulsionadas pelo desenvolvimento da *Web 2.0* e com a chegada do conceito de *Big Data*, o modelo relacional por ser rígido em consideração a consistência dos dados, acaba se tornando complexo a escalabilidade,

além de possuir um alto custo para o mesmo, por ter que adquirir mais máquinas para o escalamento do modelo relacional. Por isso, os bancos de dados NoSQL tem se tornado uma grande alternativa quando se referimos a escalabilidade e disponibilidade.

Neste aspecto, os bancos de dados NoSQL acrescentam diversos pontos positivos, sendo o seu maior ponto sua possibilidade de escalonamento e pela simplicidade do seu modelo, onde não há esquemas pré-definidos, além de introduzir novos modelos e melhores técnicas de distribuição do banco de dados.

No entanto para realizar a migração de um SGBD relacional para um banco de dados NoSQL é preciso levar diversos fatores em consideração. Deve se mensurar as diversas vantagens e desvantagens propostas por ambos os modelos, e estes critérios de comparação são dos mais diversos tipos, desde a escalabilidade do sistema, avaliação de consistência de dados e quão importante é a disponibilidade do banco de dados para o sistema.

Este artigo teve a finalidade de explicar as principais características e funcionamento dos bancos de dados NoSQL, assim como realizar algumas análises comparativas entre o modelo não-relacional com o modelo relacional que atualmente disputam e complementam o mercado. Ressaltando que não existe um banco de dados superior ao outro, a decisão do uso de cada um se refere à necessidade e características da aplicação onde será utilizado.

REFERÊNCIAS

AMAZON WEB SERVICES. **O que é NoSQL**. Disponível em:
<<https://aws.amazon.com/pé/nosql/>>. Acesso em: 9 de nov. de 2019.

ARMANDO, L.; **Banco de Dados**. Disponível em: <<https://slideplayer.com.br/slide/1260130/>>. Acesso em: 19 de abr. 2019.

CARDOSO, R.M.F.; **Bases de Dados NoSQL**. Disponível em:
<<https://core.ac.uk/download/pdf/47141116.pdf>>. Acesso em: 19 de abr. de 2019.

CARNEIRO, A.P.; MOREIRA, J.L.; FREITAS, A.L.C.; **TUNING –Técnicas de otimização de Banco de Dados um estudo comparativo: MySQL e PostgreSQL**. Disponível em:
<<http://repositorio.furg.br/bitstream/handle/1/1692/TUNING.pdf?sequence=1>>. Acesso em: 3 de set. de 2019.

DATE, C.J.; **Introdução a sistemas de bancos de dados**. Disponível em:
<<https://books.google.com.br/books?hl=pt-BR&lr=&id=xBeO9LSIK7UC&oi=fnd&pg=PP23&dq=Os+primeiros+Sistemas+Gerenciadores+de+Banco+de+Dados&ots=xbQAmYwb1P&sig=kRY0MSnfhKfRwzSOB2Z4fKgevRM#v=onepage>>

&q=Os%20primeiros%20Sistemas%20Gerenciadores%20de%20Banco%20de%20Dados&f=false>
 . Acesso em: 19 de abr. de 2019.

DIANA, M.; M.A.; **NOSQL na Web 2.0**: Um Estudo Comparativo de Bancos Não- Relacionais para Armazenamento de Dados na Web 2.0. Disponível em:
 <http://200.17.137.109:8081/novobsi/Members/josino/fundamentos-de-banco-de-dados/2012.1/sbbd_wtd_12.pdf>. Acesso em: 3 de set. de 2019.

GIL, A.C.; **Como classificar as pesquisas**. Disponível em:
 <<http://www.madani.adv.br/aula/Frederico/GIL.pdf>>. Acesso em: 19 de abr. de 2019.

KOKAY, M.C.; **Banco de dados NoSQL**: um novo paradigma. SQL Magazine, 2015. Disponível em: <<https://www.devmedia.com.br/banco-de-dados-nosql-um-novo-paradigma-revista-sql-magazine-102/25918#6>>. Acesso em: 3 de set. 2019.

LAKATOS, E.M.; MARCONI, M.A.; **Técnicas de Pesquisa**. Disponível em:
 <http://www.labev.uerj.br/textos/tecnicas-pesquisa_pesquisa-bibliografica.pdf>. Acesso em: 19 de abr. de 2019.

LEE, K.K.Y.; TANG, W.C.; CHOI, K.S.; **Alternatives to relational database**: Comparison of NoSQL and XML approaches for clinical data storage. Computer Methods and Programs in Biomedicine. ed. 1. p. 99-109. 2013.

LÓSCIO B.F.; OLIVEIRA, H.R.; PONTES, J.C.S.; **NoSql no desenvolvimento de aplicações Web colaborativas**. Disponível em:
 <https://www.researchgate.net/profile/Bernadette_Loscio/publication/268201466_NoSQL_no_desenvolvimento_de_aplicacoes_Web_colaborativas/links/576aa72008aef2a864d1ef8c/NoSQL-no-desenvolvimento-de-aplicacoes-Web-colaborativas.pdf>. Acesso em: 19 de abr. de 2019.

MACÁRIO, C.G.N.; BALDO, S.M.; **O Modelo Relacional**. Disponível em:
 <<http://www.ic.unicamp.br/~geovane/mo410-091/Ch03-RM-Resumo.pdf>>. Acesso: 19 de abr. de 2019.

MICROSOFT TECHNET. **Avaliando o desempenho**. Disponível em:
 <[https://technet.microsoft.com/pt-br/library/ms179440\(v=sql.105\).aspx#Nesta seção](https://technet.microsoft.com/pt-br/library/ms179440(v=sql.105).aspx#Nesta%20se%C3%A7%C3%A3o)>. Acesso em: 3 de set. de 2019.

MOREIRA, D.R.; DIAS, M.S.; **Web 2.0**: A Web Social. Disponível em:
 <http://www.portalcatalao.com/painel_clientes/cesuc/painel/arquivos/upload/temp/5b8d871edec20a2cea22e4a06c772a66.pdf>. Acesso em: 19 de abr. de 2019.

PANIZ, D.; **NoSQL**: Como armazenar os dados de uma aplicação moderna. Disponível em:
 <https://books.google.com.br/books/about/NoSQL.html?id=Jii5DAAAQBAJ&printsec=frontcover&source=kp_read_button&redir_esc=y#v=onepage&q=map%20reduce&f=false>. Acesso em: 19 de abr. de 2019.

PAULETO, PAINS, DANTAS E SAKURAI. **Estudo comparativo das tecnologias de banco de dados relacional vs. NoSQL (MySQL vs. MongoDB)**. São Paulo, 2013.

Disponível em:

<<https://www.metodista.br/congressos-cientificos/index.php/CM2013/facs/paper/view/5227>>.

Acesso em: 3 de set. de 2019.

RABELO, D.F; CÂNDIDO, M.V.I.; **Análise de desempenho de banco de dados NoSQL em um sistema que utiliza um banco de dados relacional e não relacional para armazenamento de dados.** Disponível em:

<http://repositorio.aee.edu.br/bitstream/aee/45/1/TCC2_2017_02_DanielFerreiraRabelo_MarcoViniciusLückeCandido.pdf>. Acesso em: 3 de set. de 2019.

RAUPP, F.M.; BAUREN, I.M.; **Metodologia de pesquisa aplicável a ciências sociais.** Disponível em:

<https://s3.amazonaws.com/academia.edu.documents/33863767/metodologia_de_pesquisa_aplicavel_as_ciencias_sociais.pdf?AWSAccessKeyId=AKIAIWOWYYGZ2Y53UL3A&Expires=1556164644&Signature=Rl1dbtnLrZWNroWkG8llz8pbcQQ%3D&response-content-disposition=inline%3B%20filename%3DMetodologia_de_pesquisa_aplicavel_as_cie.pdf>. Acesso em: 19 de abr. de 2019.

REILLY, T.O.; **Web 2.0.** Disponível em:

<https://maximiza.typepad.com/maximiza_cool_geek/files/web20_report_excerpt.pdf>. Acesso em: 19 de abr. de 2019.

REZENDE, R. **A História dos Banco de Dados.** Disponível em: <<https://www.devmedia.com.br/a-historia-dos-banco-de-dados/1678>>. Acesso em: 19 de abr. de 2019.

SADALAGE, P.J.; FOWLER, M.; **NoSQL Essencial: Um Guia Conciso para o Mundo Emergente da Persistência Poliglota.** Disponível em: <https://books.google.com.br/books?hl=pt-BR&lr=&id=m6jiDQAAQBAJ&oi=fnd&pg=PA25&dq=livro+sobre+NoSQL&ots=KWnbdlqtHz&sig=m0wsgJzagONIWqWjSy19ZLF3zQk&redir_esc=y#v=onepage&q=livro%20sobre%20NoSQL&f=false>. Acesso em: 19 de abr. de 2019.

SILBERSCHATZ, A.; SUNDARSHAN, S.; KORTH, H.F.; **Sistema de banco de dados.**

Disponível em: <<https://books.google.com.br/books?hl=pt-BR&lr=&id=1FBaDwAAQBAJ&oi=fnd&pg=PT5&dq=banco+de+dados+&ots=erJ6EeShoZ&sig=YwFbC34EvO0FEcv99H38od4CEOM#v=onepage&q=banco%20de%20dados&f=false>>.

Acesso em: 19 de abr. de 2019.

SOUZA, A.M.; PRADO, E.P.V.; SUN, V. FANTINATO, M.; **Crítérios para Seleção de SGBD NoSQL: o Ponto de Vista de Especialistas com base na Literatura.** Disponível em:

<https://www.researchgate.net/profile/Marcelo_Fantinato/publication/272158291_Criterios_para_Selecao_de_SGBD_NoSQL_o_Ponto_de_Vista_de_Especialistas_com_base_na_Literatura/links/57d221e108ae601b39a21041/Criterios-para-Selecao-de-SGBD-NoSQL-o-Ponto-de-Vista-de-Especialistas-com-base-na-Literatura.pdf>. Acesso em: 19 de abr. 2019.

SOUSA, G.; PEREIRA, J.L.; **Document-Based Databases: Estudo exploratório no âmbito das Bases de Dados NoSQL.** Disponível em:

<https://repositorium.sdum.uminho.pt/bitstream/1822/51744/3/CAPSI2015_GS_JLP.pdf>. Acesso: 19 de abr. 2019.

TOTH, R.M. **Abordagem NoSQL**: uma real alternativa. Disponível em:
<https://dcomp.sor.ufscar.br/verdi/topicosCloud/nosql_artigo.pdf>. Acesso: 19 de abr. de 2019.

IANNI, V. **Introdução aos bancos de dados NoSQL**. Disponível em:
<<https://www.devmedia.com.br/introducao-aos-bancos-de-dados-nosql/26044>>. Acesso em: 19 de abr. 2019.