

BANCO DE DADOS NOSQL: uma alternativa para grandes empresas*NOSQL DATABASE: an alternative for large companies*

Keren Antonielli Scardoelli – scardoellik@gmail.com
Faculdade de Tecnologia (Fatec) – Taquaritinga – SP – Brasil

Giuliano Scombatti Pinto – giuliano.pinto@fatectq.com
Faculdade de Tecnologia (Fatec) – Taquaritinga – SP – Brasil

DOI: 10.31510/infa.v17i2.949

Data de publicação: 18/12/2020

RESUMO

O modelo de banco de dados relacional tem sido muito utilizado no mercado pelos seus pontos fortes, mas com o surgimento da Web, empresas tiveram que começar a trabalhar com uma grande demanda de dados. Com isso, o modelo relacional acabou apresentando limitações em relação a escalabilidade, flexibilidade e disponibilidade. Este trabalho tem como objetivo apresentar algumas características dos bancos de dados relacionais e não relacionais, focando nos elementos do banco de dados não relacionais, e apresentar algumas grandes empresas que utilizam o modelo não relacional. A metodologia utilizada foi a pesquisa bibliográfica. Por meio das pesquisas realizadas, foi possível identificar que a escolha do melhor banco de dados depende da aplicação que será feita. O modelo não relacional não veio substituir o modelo relacional, mas para ser utilizado somente em ocasiões que o modelo relacional não conseguir atender determinadas necessidades.

Palavras-chave: Banco de dados. Modelo Relacional. NoSQL.

ABSTRACT

The relational database model has been widely used in the market for its strengths, but with the emergence of the Web, companies had to start working with a high demand for data. As a result, the relational model ended up with limitations regarding scalability, flexibility and availability. This work aims to present some characteristics of the relational and non-relational databases, by focusing on the elements of the non-relational database, and to present some large companies that use the non-relational model. The methodology used was bibliographic research. Through the researches carried out, it was possible to identify that the choice of the best database depends on the application that will be made. The non-relational model did not replace the relational model, but it is to be used only on occasions when the relational model is unable to meet certain needs.

Keywords: Database. Relational Model. NoSQL.

1 INTRODUÇÃO

A Informação é um conjunto organizado de dados que é vital para o sucesso de indivíduos e instituições na atualidade. O crescimento da quantidade de informações e dados na web é perceptível hoje em dia, com a popularização da internet, houve um aumento na quantidade de usuários, e esse aumento tem sido constante. De acordo com a pesquisa mais recente do IBGE (2018), mostra que em 2017 o Brasil possuiu 126,3 milhões de usuários de internet, o que levou a um aumento de 10,2 milhões de usuários em comparação com 2016. Com isso, conseqüentemente os dados e informações a serem armazenados crescem junto e por isso que os bancos de dados têm sido indispensáveis, pois com eles é possível armazenar, controlar e disponibilizar esses dados e informações.

Quando surgiram as aplicações em computadores, os dados eram armazenados em arquivos e não havia uma preocupação com desempenho, segurança e facilidade de modificação dos dados e organização deles. Com o aumento dos dados e da complexidade das aplicações, surgiu a necessidade de fazer algo para que pudesse armazenar e controlar dados de uma maneira eficaz (OLIVEIRA, M., 2014). Com isso foram criados os bancos de dados, que hoje têm sido essenciais na sociedade.

Um dos modelos de bancos de dados amplamente usados até os dias de hoje é o modelo relacional, que foi introduzido por Edgar Codd em 1970, trazendo uma nova forma de representar dados (MACÁRIO; BALDO, 2005). Ele visa garantir ao usuário, a recuperação de falhas, integridade, rapidez em consultas, segurança, entre outros benefícios (BRITO, 2010).

Porém, viu-se que os modelos de bancos de dados relacionais têm limitações ao trabalhar com grandes volumes de dados. Com isso surgiu em 1998 o banco de dados não relacional, que veio para resolver esse problema, garantindo maior disponibilidade, desempenho e escalabilidade (BRITO, 2010).

No ambiente da Internet (ou na plataforma WEB), é importante que um banco de dados ofereça desempenho, devido à própria questão de variedade de velocidade de conexões e segurança, visto que pode existir milhões de usuários nesta plataforma, e o banco de dados não relacional tem mostrado um desempenho melhor que os bancos de dados relacionais.

Diante disso, o presente trabalho tem por objetivo apresentar os principais conceitos e características do modelo não relacional, mostrar em qual situação ele apresenta melhor

desempenho, apresentando também alguns sistemas gerenciadores de banco de dados não relacional mais populares e onde eles são utilizados hoje em dia na Internet.

A metodologia utilizada no presente trabalho é a pesquisa bibliográfica, realizada em livros, artigos, monografias e teses.

2 BANCO DE DADOS

Banco de dados é uma expressão que vem do inglês *Databanks*, que logo foi substituída por *DataBases*, que significa Base de Dados. Esse nome traz um significado mais apropriado para o que um banco de dados realmente faz, pois ele funciona como um repositório de dados, onde esses dados podem ser usados em diversas aplicações, e não como um banco, emprestando dados (SETZER; SILVA, 2005).

Segundo Date (2004), um banco de dados é um depósito de dados armazenados, necessários para a manutenção das atividades de uma instituição, nas quais as informações podem ser adicionadas, alteradas, removidas ou recuperadas.

Assim, um banco de dados é um conjunto organizado de arquivos relacionados entre si, nas quais esses arquivos podem ser usados em diversas tarefas em uma determinada aplicação.

2.1 SGBD

Para que possa se manipular, organizar, ter acesso e manter segurança das informações e dados que estão em um banco de dados, existe um sistema que se chama SGBD (Sistema Gerenciador de Banco de Dados).

Segundo Date (2004, p.40) Sistema Gerenciador de Banco de Dados “é um sistema computadorizado cuja finalidade geral é armazenar informações e permitir que os usuários busquem e atualizem essas informações quando as solicitar”.

“O SGBD é um software que funciona como uma interface entre o usuário e o Banco de Dados, ou seja, todas as solicitações dos usuários, como criação de tabelas, inserção de dados, recuperação de dados, são manipuladas pelo SGBD” (CARNEIRO; MOREIRA; FREITAS, p.2).

Um SGBD faz, na verdade, exatamente o que diz seu nome, ou seja, ele é o responsável pelo gerenciamento dos bancos de dados. Diante disso, tudo que se faz no banco de dados passa pelo SGBD.

Os primeiros SGBDs surgiram em 1960, aproximadamente, e foram desenvolvidos de acordo com os primeiros sistemas de arquivos. Os primeiros a serem criados foram os modelos em rede e hierárquico. Logo depois, nos anos 60, foi criado o modelo relacional, que foi bem aceito por armazenar e manipular dados estruturados, isto é, dados que mantem uma estrutura fixa e organizada. Anos depois, com o surgimento da WEB, viu-se a necessidade de se criar um modelo de banco de dados onde pudesse ter a manipulação em grande volume de dados semi-estruturados ou não estruturados, então foi criado o modelo não relacional (LÓSCIO; OLIVEIRA; PONTES, 2011).

2.1.1 Banco de dados relacional

O modelo de banco de dados relacional foi introduzido por Edgar Codd em 1970. Com o passar do tempo, ele cresceu muito e começou a dominar o mercado de banco de dados, superando os modelos hierárquico e em rede que eram os mais utilizados na época. Algumas coisas que fizeram o modelo relacional se sobressair em relação aos outros modelos são a representação dos dados de uma forma simples e a simplificação da realização de consultas complexas (MACÁRIO; BALDO, 2005).

Atualmente, o modelo relacional é muito utilizado e, mesmo com a evolução tecnológica, ele continua dominante no mercado de banco de dados, pois oferece a recuperação de falhas, integridade, rapidez em consultas, segurança, entre outros benefícios (BRITO, 2010).

Tudo isso porque uma das características do banco relacional é o processamento de transações com base nas propriedades ACID (Atomicidade, Consistência, Isolamento, Durabilidade).

Suas definições segundo Lóscio, Oliveira e Pontes (2011, p. 3), são

- **Atomicidade:** todas as operações da transação são executadas, ou seja, a transação é executada por completo, ou nada é executado;
- **Consistência:** após uma transação ter sido concluída, o banco de dados deve permanecer em um estado consistente, ou seja, deve satisfazer as condições de consistência e restrições de integridade previamente assumidas;
- **Isolamento:** as duas transações estão sendo executadas concorrentemente seus efeitos devem ser isolados uma da outra. Esta propriedade está relacionada ao controle de concorrência do SGBD;

- **Durabilidade:** uma vez que uma transação ocorreu com sucesso, seu efeito não poderá mais ser desfeito, mesmo em caso de falha. Esta propriedade está relacionada a capacidade de recuperação de falhas do SGBD.

Outro aspecto que o faz ser o modelo mais utilizado é a possibilidade de usar sistemas gerenciadores de bancos de dados que executam comandos na linguagem SQL (*Structured Query Language*).

SQL é uma linguagem de programação para lidar com banco de dados relacional, que foi criada originalmente pelo IBM para consulta ao seu Sistema-R. Com sua evolução, ela se tornou a linguagem padrão para SGBDs relacionais (MACÁRIO; BALDO, 2005).

Apesar de toda a completude do banco relacional, foi visto que ele não é tão escalável como necessário, pois quando se precisa gerenciar muitos dados a sua performance acaba sendo baixa (OLIVEIRA, S., 2014).

Com isso, surgiu um novo modelo de banco de dados chamado NoSQL, que trouxe uma solução para a questão da escalabilidade, disponibilidade e manipulação de grandes volumes de dados não estruturados ou semi-estruturados (LÓSCIO; OLIVEIRA; PONTES, 2011).

2.1.2 Banco de dados não relacional

O termo NoSQL (abreviação para “*Not Only SQL*”), que em português se traduz “Não Apenas SQL”, foi introduzido por Carlo Strozzi em 1998 (BARBOSA, 2013). Esse modelo foi proposto para suprir a necessidade de lidar com a grande demanda de dados e informações das aplicações web, que crescem a cada dia. Ele surgiu com o objetivo de “gerenciar grandes volumes de dados semi-estruturados ou não estruturados, que precisam de alta disponibilidade e escalabilidade” (LÓSCIO; OLIVEIRA; PONTES, 2011, p. 1).

Eles têm esse nome “pois armazenam os dados de forma não estruturada e não exigem esquemas fixos, assim como os banco de dados relacionais” (LEAVITT, 2010 *apud* SOUZA; PRADO; SUN; FANTINATO, 2014, p. 150). Além disso, também não seguem o modelo relacional e, como o termo NoSQL já denota, não usam SQL como linguagem nativa para interagir com os bancos de dados, o que faz com que a atividade de desenvolvimento seja mais complexa. Entretanto, eles possuem uma estrutura simplificada e são mais flexíveis em relação ao modelo ACID (SOUZA; PRADO; SUN; FANTINATO, 2014).

Atualmente é fundamental que as aplicações WEB atendam uma grande demanda de dados "que precisam ser processados, analisados e entregues aos usuários que os requisitam" (FERNANDES, 2013, p. 8). Para atender demandas elevadas de dados requisitados por picos

de acessos de usuários de forma eficiente e rápida, resultando em um bom desempenho, é necessário reduzir a consistência e priorizar a disponibilidade e a escalabilidade. Isso não poderá ser atingido de acordo com as propriedades ACID. Assim sendo, as empresas preferem perder uma das propriedades para poder focar nas suas necessidades mais importantes. Então, surgiu o conceito BASE, que desconsidera as propriedades de consistência e isolamento para ganhar disponibilidade e performance (FERNANDES, 2013).

BASE (**B**asically Available, **S**oft state, **E**ventual consistency) que se caracteriza por ser basicamente disponível, ou seja, o sistema parece estar funcionando o tempo todo; em estado leve, o sistema não precisa ser consistente o tempo todo; e eventualmente consistente, o sistema torna-se consistente no momento devido (BRITO, 2010, p. 5).

- Essa ideia se baseia no Teorema CAP, que foi apresentado em 2000 por Eric Brewer e depois de dois anos foi comprovado por Seth Gilbert e Nancy Lynch. Ele afirma que em um sistema computacional distribuído, é impossível garantir consistência, disponibilidade e tolerância ao particionamento, de forma simultânea. Segundo esse teorema, só é possível garantir duas dessas propriedades ao mesmo tempo (LEITE, 2010). Em relação ao teorema, há algumas considerações importantes. São elas: Para um sistema ser consistente é preciso garantir que após ter feito uma operação, ele mantenha a integridade dos dados (WEI, 2009 *apud* LEITE, 2010). Segundo Fernandes (2013, p. 10) “isto, normalmente significa que uma vez escrito um registro, este fica disponível para ser utilizado imediatamente.”
- A disponibilidade “deve garantir que um sistema esteja sempre fornecendo acesso e funcionalidades a seus usuários. É uma propriedade indispensável para sistemas de empresas que precisam disponibilizar seus serviços continuamente” (BROWNE, 2009 *apud* LEITE, 2010, p. 11). Segundo Fernandes (2013, p. 10), “neste contexto, significa que um sistema [...] normalmente também permanece disponível durante a atualização de software e hardware.”
- E “a tolerância ao particionamento está relacionada à habilidade de um sistema continuar operando após ocorrerem particionamentos na rede” (GILBERT, 2002 *apud* LEITE, 2010, p. 11). “É a capacidade de um sistema se manter operante mesmo em casos onde ocorra uma interrupção parcial de alguns componentes” (FERNANDES, 2013, p. 10).

2.2 Tipos de Banco de Dados NoSQL

Existem quatro tipos principais de banco de dados NoSQL: chave-valor, orientado a documentos, orientado a colunas e orientado a grafos. Cada tipo de modelo tem características diferentes para atender aplicações diferentes, isso mostra que um não deve ser considerado melhor que outro (LÓSCIO; OLIVEIRA; PONTES, 2011).

2.2.3 Banco de dados do tipo chave-valor

É considerado um modelo bem simples, pois cada item possui chaves e valores, e esses valores podem ser acessados fazendo referência a sua chave, sem a necessidade de fazer consultas complexas. Os bancos mais populares que utilizam esse tipo são Dynamo, Redis, Riak e GenieDB (LÓSCIO; OLIVEIRA; PONTES, 2011).

2.2.4 Banco de dados orientados a documento

Armazena os dados como documentos. Esses documentos podem conter informações de diversos tipos, tornando esse modelo de banco uma boa opção para armazenamento de dados não estruturados (VICTÓRIA, 2019). Cada documento é um objeto que recebe um id único.

Alguns dos bancos de dados que utilizam esse modelo são o CouchDB e o MongoDB (LÓSCIO; OLIVEIRA; PONTES, 2011).

2.2.5 Banco de dados orientados a grafos

“O modelo de dados orientado a grafos é um modelo de dados em que os dados são armazenados em nós de um grafo e as arestas representam o tipo de associação entre os nós.” (BARBOSA, 2013, p. 23).

Nele, são utilizados três componentes: "os nós (são os vértices do grafo), os relacionamentos (são as arestas) e as propriedades (ou atributos) dos nós e relacionamentos." (LÓSCIO; OLIVEIRA; PONTES, 2011, p. 8). Este modelo acaba se sobressaindo aos outros quando é preciso fazer consultas complexas, pelo seu ganho de performance. Alguns bancos que utilizam esse modelo são: Neo4J, AllegroGraph e Virtuoso (LÓSCIO; OLIVEIRA; PONTES, 2011).

2.2.6 Banco de dados orientados a colunas

Este modelo é o oposto do relacional que armazena conjuntos de dados somente em uma linha, já esse modelo armazena os dados em linhas particulares de tabela (VICTÓRIA, 2019).

Neste modelo os dados são indexados por uma tripla (linha, coluna e timestamp), onde as linhas e as colunas são identificadas por chaves e o timestamp permite diferenciar múltiplas versões de um mesmo dado. [...] Outro conceito associado ao modelo é o de família de colunas (*columnfamily*), que é usado com o intuito de agrupar colunas que armazenam o mesmo tipo de dados.

Hbase e Cassandra são bancos que utilizam o modelo de colunas (LÓSCIO; OLIVEIRA; PONTES, 2011, p. 7 e 8).

3 PROCEDIMENTOS METODOLÓGICOS

A metodologia utilizada no presente trabalho foi a pesquisa bibliográfica. Foram realizadas diversas pesquisas em livros, artigos, monografias e teses. Inicialmente, foram abordadas as características gerais dos bancos de dados. Em seguida, foi apresentado os principais elementos dos bancos de dados relacionais e não relacionais. Posteriormente, os principais tipos de banco de dados NoSQL. Por fim, foram citadas algumas grandes empresas que utilizam o banco de dados não relacional para o gerenciamento dos seus dados.

4 RESULTADOS E DISCUSSÃO

4.1 Principais diferenças entre os bancos de dados relacionais e os não relacionais

Quando é preciso criar uma aplicação para gerenciar dados, é preciso levar em consideração critérios como escalabilidade, consistência de dados e disponibilidade. (BRITO, 2010).

Nos bancos de dados não relacionais, a escalabilidade se torna a maior vantagem, pois eles foram criados para esse propósito, na qual é possível incluir uma grande quantidade de dados. Já os bancos relacionais têm uma estruturação não flexível, na qual é difícil adaptar a escalabilidade em aplicações quando é preciso (BRITO, 2010).

A consistência é o ponto mais forte dos bancos de dados relacionais, pois possuem regras com maior rigidez em relação à consistência das informações. Já nos bancos não

relacionais isso ocorre eventualmente, pois ele não garante atualizações. Se não ocorrer atualizações sobre um determinado item de dados, será devolvido o último valor atualizado (BRITO, 2010).

Por último há a disponibilidade. Os bancos relacionais apresentam uma certa dificuldade em trabalhar de uma forma eficiente com distribuição de dados por conta de sua natureza estruturada. E para o modelo não relacional, isso é outro ponto forte que ele possui. O grau elevado de distribuição dos dados garante que o sistema fique um tempo menor não disponível e também permite que um grande número de solicitações aos dados sejam atendidas (BRITO, 2010).

É possível ver que os bancos de dados não relacionais mantêm seu foco em escalabilidade e disponibilidade, enquanto os relacionais priorizam consistência. Por isso e outras vantagens que grandes empresas acabam optando por usar os bancos de dados não relacionais.

4.2 Grandes empresas que utilizam banco de dados NoSQL

O NoSQL tem sido uma alternativa para as grandes empresas, pelos seus pontos fortes, como escalabilidade, flexibilidade, boa performance, entre outros. A seguir, serão apresentadas algumas empresas que utilizam o NoSQL para o gerenciamento dos seus dados.

Em 2004 surgiu uma das primeiras implementações de sistema não relacional. Foi quando o Google desenvolveu sua própria solução NoSQL, o BigTable, na qual seu foco era promover maior escalabilidade e disponibilidade (BRITO, 2010). Alguns produtos que utilizam o BigTable são o Gmail, Google Docs, Google Analytics, entre outros (LÓSCIO; OLIVEIRA; PONTES, 2011).

Em 2015 foi lançada uma versão pública do BigTable, chamada CloudBigtable, que é baseado no projeto Bigtable e é compatível com a API Apache HBase (IMASTERS, 2015).

Em 2007, a Amazon desenvolveu uma solução NoSQL chamada Dynamo, com o objetivo de "garantir alta disponibilidade dos dados de seus serviços *"always-on"*" (LÓSCIO; OLIVEIRA; PONTES, 2011, p. 11). Ele foi criado para resolver alguns problemas que o site Amazon.com estava tendo em relação a confiabilidade dos dados. Por ter adotado essa tecnologia, "diversos serviços da Amazon têm se mantido disponíveis em 99,9995% das requisições realizadas" (LÓSCIO; OLIVEIRA; PONTES, 2011, p. 11).

Outras grandes empresas que estão crescendo no mundo utilizam o DynamoDB, como a Nike Digital, que migrou para o DynamoDB para melhorar a experiência de seus clientes, a Samsung Electronics, que o utiliza para o *backup* de aplicativos móveis, o Snapchat Stories, que migrou para o Dynamo para aumentar o seu desempenho e reduzir custos, e a Netflix, que o utiliza para fazer testes A/B que criam experiências de streaming personalizadas (AMAZON, 2020).

Com a popularização do Facebook, surgiu a necessidade de criar um banco de dados que pudesse ser altamente escalável, com alta disponibilidade e tolerante a falhas, para poder gerenciar seus grandes volumes de dados. Com isso, em 2008 o Facebook desenvolveu o Cassandra. No início, ele foi "criado para otimização do sistema de busca do facebook" e, "atualmente o Cassandra é utilizado para dar suporte à replicação, detecção de falhas, armazenamento em cache dentre outras funcionalidades."(LÓSCIO; OLIVEIRA; PONTES, 2011, p. 11).

Em 2010 o Twitter teve que substituir o MySQL pelo Cassandra, também por causa de problemas com disponibilidade. "A empresa utiliza o Cassandra para armazenar resultados de *data mining* realizados sobre a base dos usuários, resultados de *trendtopics*, *@toptweets* e análises em tempo real em larga escala." (LÓSCIO; OLIVEIRA; PONTES, 2011, p. 11).

Existem várias outras empresas que utilizam bancos de dados não relacionais como Globo, Yahoo, The New York Times, Ebay, entre outras (CHAGAS; ARAGÃO; SILVA, 2011).

5 CONSIDERAÇÕES FINAIS

O banco de dados não relacional vem crescendo entre as empresas por apresentar várias vantagens em relação a sua performance, escalabilidade e facilidade ao utilizá-lo. Como foi visto, a maioria das empresas optaram por usar o banco de dados NoSQL pela sua vantagem em poder gerenciar uma grande quantidade de dados e assim ter um alta disponibilidade, o que acaba não sendo possível com o modelo relacional.

É importante ressaltar que isso não quer dizer que o banco de dados NoSQL veio para substituir o modelo relacional, ele na verdade acaba sendo uma alternativa para determinadas aplicações da WEB, que necessitam de uma boa performance, alta disponibilidade,

escalabilidade e ter um esquema flexível para poder gerenciar dados semi-estruturados e não estruturados.

Portanto, para poder garantir todos esses requisitos, acaba muitas vezes não sendo possível obter consistência dos dados, controle de concorrência, recuperação de falhas, segurança, entre outros, que são limitações do modelo NoSQL e vantagens do modelo relacional. Portanto, dependendo do software também existe a possibilidade de utilizar os dois modelos para poder suprir as necessidades de determinadas aplicações.

REFERÊNCIAS

AMAZON. **Amazon DynamoDB**. Disponível em: <<https://aws.amazon.com/pt/dynamodb/>>. Acesso em: 25 jul. 2020.

BARBOSA, M. J. O. de.; **Análise Comparativa de Banco de Dados Relacionais e NoSQL em um Ambiente de Computação nas Nuvens**. Disponível em: <http://fbuni.edu.br/sites/default/files/tcc_-_2013_1_-_maria_josiane_de_oliveira.pdf>. Acesso em: 25 jul. 2020.

BRITO, R. W.; **Banco de Dados NoSQL x SGBDs Relacionais: Análise Comparativa**. Disponível em: <<http://docplayer.com.br/433629-Bancos-de-dados-nosql-x-sgbds-relacionais-analise-comparativa.html>>. Acesso em: 25 jul. 2020.

CARNEIRO, A. P.; MOREIRA, J. L.; FREITAS, A. L. C. de. **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: 10 abr. 2020.

CHAGAS, A. T. S.; ARAGÃO, N.; SILVA, M. **NoSQL**. Disponível em: <<https://pt.slideshare.net/tadeuchagas/artigo-nosql-10405449>>. Acesso em: 25 jul. 2020.

DATE, C. J.; **Introdução a Sistemas de BANCO DE DADOS**. 8. ed. Rio de Janeiro: Elsevier, 2003.

FERNANDES, A. S. de.; **NOSQL BASE X ACID TEOREMA CAP**. Disponível em: <<https://document.onl/technology/sistemas-nosql-surgimento-caracteristicas-e-exemplos.html>>. Acesso em: 25 jul. 2020.

IBGE. **PNAD Contínua TIC 2017: Internet chega a três em cada quatro domicílios do país**. Disponível em: <<https://agenciadenoticias.ibge.gov.br/agencia-sala-de-imprensa/2013->

agencia-de-noticias/releases/23445-pnad-continua-tic-2017-internet-chega-a-tres-em-cada-quatro-domicilios-do-pais>. Acesso em: 04 out. 2020.

IMASTERS. **Google lança Cloud Bigtable, um banco de dados NoSQL altamente escalável.** Disponível em: <<https://imasters.com.br/noticia/google-lanca-cloud-bigtable-um-banco-de-dados-nosql-altamente-escalavel>>. Acesso em: 25 jul. 2020.

LEITE, G. S.; **Análise Comparativa do Teorema CAP Entre Banco de Dados NoSQL e Banco de Dados Relacionais.** Disponível em: <<http://fbuni.edu.br/sites/default/files/tcc-20102-gleidson-sobreira-leite.pdf>>. Acesso em: 25 jul. 2020.

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: 27 mar. 2020.

MACÁRIO, C. G. do N.; BALDO, E. M. **O Modelo Relacional.** Disponível em: <<https://www.ic.unicamp.br/~geovane/mo410-091/Ch03-RM-Resumo.pdf>>. Acesso em: 27 mar. 2020.

OLIVEIRA, M. N. **Estudo Comparativo Entre Sistemas de Banco de Dados NoSQL e Relacional.** Disponível em: <<https://www.cin.ufpe.br/~tg/2014-2/mno2-proposta.pdf>>. Acesso em: 27 mar. 2020.

OLIVEIRA, S. S. de. **Banco de Dados não-relacionais: um novo paradigma para armazenamento de dados em sistemas de ensino colaborativo.** Macapá, v.2 n. 1, p. 184-194, ago.- dez. 2014.

SETZER, V. W.; SILVA, F. S. C. da. **BANCOS DE DADOS: APRENDA O QUE SÃO, MELHORE SEU CONHECIMENTO, CONSTRUA OS SEUS.** São Paulo: Edgard Blucher, 2005.

SOUZA, A. M.; PRADO, E. P. V.; SUN, V.; FANTINATO, M. **Critérios 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: 22 mar. 2020.

VICTÓRIA, P.; **Banco de dados NoSQL: um manual prático e didático.** Disponível em: <<https://blog.geekhunter.com.br/banco-de-dados-nosql-um-manual-pratico-e-didatico/>>. Acesso em: 14 jun. 2020.