BANCO DE DADOS EM BIG DATA: Uma abordagem sobre o NoSQL

Eder dos Santos Ferreira*

Resumo

Com a evolução da tecnologia, novos dispositivos de sensoriamento e acesso remoto em conjunto com gigantesca crescente no acesso à informação quase que em tempo real, surge o grande aumento na geração e consulta de dados. Com isso, a procura por um sistema altamente performático e seguro para o armazenamento e manutenção desses dados também cresceu. Cada acesso na internet, cada página visitada, cada clique efetuado e até o tempo em que o usuário permanece na página, pode gerar milhões de dados, que de nada valem se não forem processados e armazenados. Além disso, hoje em dia praticamente tudo é monitorado através de sensores incluindo carros autônomos ou não, aviões que geram bilhões de dados em um único vôo, semáforos de trânsito, equipamentos hospitalares, dispositivos residenciais como smart tvs, geladeiras, dispositivos controladores de casa inteligente, entre muitos outros, das mais variadas aplicações. Com toda essa evolução surgiu o Big Data que norteia as conformidades de manutenção e acesso a toda essa informação. Todos esses dados, dependendo da aplicação, precisam estar devidamente armazenados conforme sua necessidade. O objetivo desse artigo é apresentar, em conjunto com uma fundamentação literária, as características que envolvem os sistemas de armazenamento desses dados e como eles são classificados de acordo com a sua aplicação. Também é apresentada a relação entre o NoSQL e os bancos de dados disponíveis no mercado que são utilizados no contexto de Big Data, além das suas conformidades com as características do *ACID*, *BASE* e Teorema CAP.

Palavras-chave: Dispositivos de sensoriamento. Acesso remoto. Big Data. NoSQL. Banco de dados. ACID. BASE. Teorema CAP.

1 Introdução

Segundo Perez (2010), cada revolução, seja ela industrial ou econômica, é seguida de novos desafios que alteram o comportamento das empresas. Há algumas décadas a humanidade tem presenciado algumas revoluções da indústria. Atualmente, estamos vivenciando a quarta revolução que é denominada de Revolução Industrial, também chamada de Indústria 4.0.

Ainda assim, para identificar a Indústria 4.0, segundo Hermann et al (2016), é preciso que a referida tenha os seguintes princípios: capacidade de operação em

* Pós-Graduação em Ciência de Dados e Big Data Analytics da Universidade Estácio de Sá, Administrador de Banco de Dados Oracle. Email eder.sferreira@gmail.com.

tempo real, virtualização, descentralização, orientação a serviços, modularidade e interoperabilidade.

Para Rüßmann (2015), à indústria 4.0 são atribuídos nove pilares tecnológicos, conforme segue: big data e análise de dados, robôs autônomos, simulação, integração horizontal e vertical de sistemas, internet das coisas industrial, segurança cibernética, nuvem, fabricação de aditivos e realidade aumentada.

Com essa nova forma de operar, as empresas modernas têm uma crescente necessidade de integração em tempo real entre máquinas, fornecedores, clientes, funcionários, etc., o que demanda comunicação padronizada e confiável com o intuito de evitar falhas nos processos, segundo Morais (2018).

Com tantos dados sendo gerados e junto à eles a necessidade de processá-los e armazená-los, é preciso que sejam utilizados bancos de dados capazes de atuar de forma satisfatória, oferecendo segurança e desempenho de acordo com a aplicação.

2 Desenvolvimento

2.1 O que é Big Data?

Segundo Zanjireh e Larijani (2015), a interconexão digital de objetos cotidianos é denominada Internet das Coisas ou apenas IoT (*Internet of Things*), ou seja, a internet das coisas é uma rede com objetos físicos que possuem tecnologia embarcada capaz de se conectar à internet e transmitir dados.

Para Mauro e Grimaldi (2016), a crescente do Big Data foi devido ao alto grau no qual os dados são compartilhados, utilizados e criados atualmente e, segundo Kampakis (2020), Big Data é apenas uma complexa e enorme massa de dados que está relacionada a bancos de dados NoSQL e computação nas nuvens e, cujos softwares tradicionais utilizados para o seu processamento não são capazes de tratá-los para uso final.

McAfee e Brynjolfsson (2012) complementam o conceito de Kampakis (2020) sobre Big Data descrevendo que o termo, além de referir-se a um grande volume complicado de dados, eles são de tipos diversos, heterogêneos e oriundos de fontes distintas. Também classificam Big Data com 3Vs (Volume, Velocidade e Variedade),

alegando que nas últimas décadas, essas características aumentaram significativamente. Walker (2015), no entanto, complementa esse conceito com mais 2 (duas) classificações que melhoram as operações dos dados, permitindo sua recuperação posterior somando um total de 5Vs. Demchenko (2014), por sua vez, aprimora o conceito dos 5Vs adicionando mais um item, a variabilidade, tornando o conjunto com 6Vs que será descrito abaixo e na Figura 1:

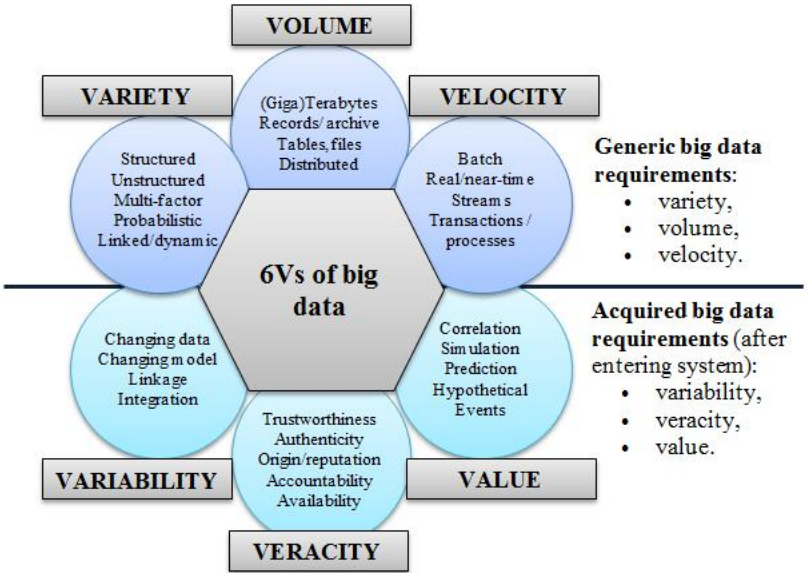

Figura 1 - 6 Vs
Fonte: Demchenko et al (2013)

Segundo McAfee e Brynjolfsson (2012):

1. **Volume**: A quantidade de dados cresce exponencialmente todos os dias, o Big Data permite que eles sejam modelados e analisados;
2. **Variedade**: Os dados são gerados de diferentes plataformas digitais, podendo ter formato de imagens, sensores de GPS, mensagens, etc.;
3. **Velocidade**: Os dados podem ser obtidos em tempo real, de forma ágil, compatível com a demanda;

Segundo Walker (2015):

4. **Veracidade**: Os dados consultados são fiéis aos coletados e armazenados nos bancos de dados;

5. **Valor**: Diretamente ligado à capacidade da organização de tomar decisões, fazendo com que a informação tenha valor e os benefícios possam ser extraídos dela;

Segundo Demchenko (2014):

6. **Variabilidade**: No contexto de Big Data refere-se às inconsistências ou mudanças de dados entre eles uma vez que os dados podem ser oriundos de tipos de fontes diferentes.

Segundo Furlan e Laurindo (2017), a grande crescente dos dados foi a motivação para o surgimento de novas tecnologias que fossem capazes de gerenciar a enorme quantidade de dados gerados. Dessa carência é que nasce o Big Data com novas metodologias para gerar, selecionar, manipular, e armazenar a gigantesca volumetria de dados.

Para a IBM (2012), em 2012 foram gerados cerca de 2,5 quintilhões de bytes de dados e quase 90% deles foram gerados em 2010 e 2011 e, logo depois, segundo a IBM (2013), a expectativa era que até 2020 2,3 trilhões de gigabytes diários seriam gerados.

2.2 Dados Estruturados, Não estruturados e Semiestruturados

2.2.1 Estruturados

De acordo com Joyanes (2013), o que identifica dados estruturados é a facilidade de acesso devido à sua estrutura bem definida e são divididos em dois grupos: estáticos (como array, cadeia de registros) e listas dinâmicas (pilhas, caudas, arquivos e árvores).

Esses tipos de dados são gerados, formatados e transformados em um modelo bem definido e, geralmente, utilizados em bancos de dados relacionais. Como exemplo de dados estruturados podemos citar alguns de *PDV* como quantidade, códigos de barra e estatísticas de weblog, além de planilhas que tem dados organizados e de fácil manuseio ASTERA (2021).

Dados estruturados são armazenados em banco de dados relacionais e a linguagem SQL é utilizada para acessar e manipular esses dados. Um outro exemplo de dados estruturados é um cadastro de pessoas conforme o Quadro 1, Lowtomated (2019):

ID	Forename	Surname	Age
0	Arnold	Schwarzenegger	71
1	Sylvester	Stallone	72
2	Chuck	Norris	79

Quadro 1 - Exemplo de dados estruturados
Fonte: Lowtomated (2019)

2.2.2 Não estruturados

Os dados na sua forma bruta, sem tratamento, são denominados de dados não estruturados. São dados que possuem uma missão difícil de organizar, formatar e gerenciar por serem oriundos de múltiplas fontes, como mídias sociais, chats, sensores de *IoT*, emails, imagens de satélites, etc., ASTERA (2021), como mostrado na Figura 2.

Figura 2: Fontes de dados não estruturados
Fonte: Atlan (2020)

Na atualidade, os grandes dados são gerados das mídias sociais e necessitam de armazenamento e processamento específico para garantir uma melhor eficiência na sua recuperação e análise, segundo Santos e Ferreira (2013).

Dados não estruturados não podem ser armazenados em bancos de dados relacionais, apenas em bancos de dados geralmente referidos como NoSQL tais como: *MongoDB*[1], *Cassandra*[2], *HBase*[3], *Redis*[4], *BigTable*[5] e *Oracle NoSQL Database*[6], Atlan (2020).

2.2.3 Semiestruturados

Esse tipo de dado é um intermediário entre estruturado e não estruturado por conter algumas características consistentes e definidas (estruturadas), mas que também contém dados não estruturados em uma estrutura variável ASTERA (2021). Segundo Karl (2011), geralmente esses dados utilizam tags para facilitar a distinção entre eles.

Como um exemplo de dado semiestruturado, podemos usar uma fotografia digital. Mesmo que a imagem não possua uma estrutura própria definida, ela poderá ter alguns atributos estruturados, no caso de ser oriundo de um smartphone. Alguns atributos podem ser listados como geotag, ID do dispositivo e carimbo de data e hora ASTERA (2021).

2.2.4 Divisão de dados Estruturados e Não Estruturados

Como já visto, de acordo com McAfee e Brynjolfsson (2012), os principais requisitos do Big Data são os 3Vs (Volume, Velocidade e Variedade) que nos proporciona uma enorme diferença na quantidade de geração de dados quando se trata de tipos de dados. A Figura 3 descreve a divisão mostrando a origem dos dados.

---

[1] https://www.mongodb.com
[2] https://cassandra.apache.org
[3] https://hbase.apache.org
[4] https://redis.io
[5] https://cloud.google.com/bigtable/
[6] https://www.oracle.com/database/technologies/related/nosql.html

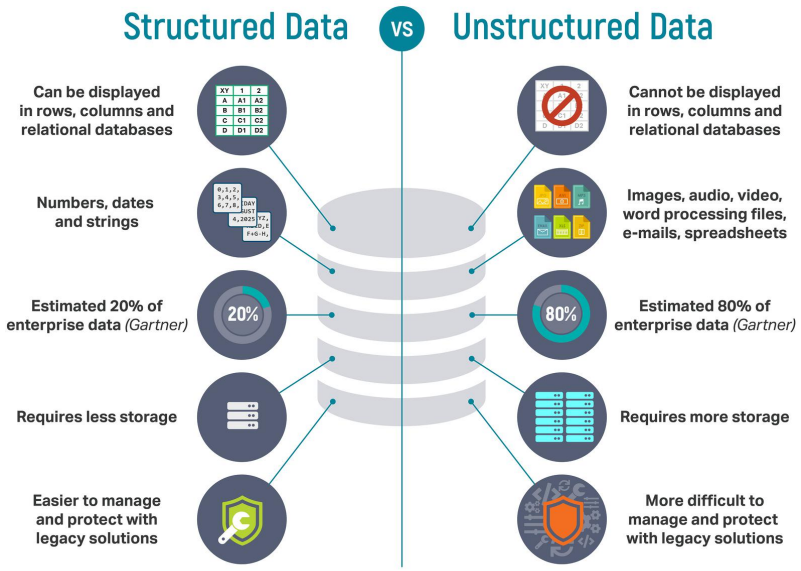

Figura 3: Proporção entre os dados estruturados e não estruturados
Fone: Lowtomated (2019)

2.3 ACID e BASE

O cientista da computação Jim Gray (1981) já havia feito menção a três dos quatro atributos: Atomicidade, Consistência e Durabilidade. Posteriormente em 1983, foi inventado o acrônimo ACID Härder e Reuter (1983). O ACID é uma regra de conformidade muito importante dos bancos de dados relacionais e significa o seguinte Dummies (2020):

1. **Atomicidade**: Atributo que define a completude da transação, ou seja, ou ela finaliza com sucesso e todos os dados necessários são gravados ou, no caso de qualquer falha, nenhum dado da transação será gravado;
2. **Consistência**: Uma transação no banco de dados deve manter sua consistência, ou seja, os dados escritos no banco de dados devem ser válidos conforme todas as regras definidas nas restrições (*constraints*), operações em cascata (*cascade*), *triggers* ou qualquer outra funcionalidade que defina essas regras;
3. **Isolamento**: Esse atributo está diretamente relacionado com o controle de concorrência. O isolamento é um conjunto de técnica utilizado para evitar transações paralelas que tentam alterar o mesmo dado, assim, o banco de dados deve tratar e serializar essas transações;

4. **Durabilidade**: Após a confirmação (commit) das alterações executadas na transação, os dados alterados são persistidos mesmo se o banco de dados for desligado ou sofrer qualquer incidente após a transação.

No contexto de NoSQL as características de conformidade com ACID são descartadas e assumida uma nova regra denominada BASE que também é um acrônimo *(**B**)asically(**A**)vailabe, (**S**)oft State, (**E**)ventually Consistent,* Dummies (2020):

1. **Basically Availabe**: Indica que o banco de dados garante disponibilidade;
2. **Soft State**: Indica que o banco de dados não garante consistência todo o tempo, ou seja, os dados armazenados podem sofrer alterações;
3. **Eventually Consistent**: Esse atributo é um modelo de consistência usado em computação distribuída para alcançar a alta disponibilidade. Conforme os dados são gravados, eles são replicados nos outros nós, no caso por exemplo de um sistema Hadoop com mais de um nó. Durante a replicação, os dados são inconsistentes.

As propriedades BASE são mais flexíveis em relação às propriedades ACID e, de acordo com Robinson et al (2015) não há nenhum mapeamento direto entre eles. No entanto, alguns autores como Sudoers (2015) consideram um mapeamento que permeia a relação entre eles, conforme a Tabela 1:

ACID	BASE
Consistência	Fraca consistência
Isolamento	Foco em Disponibilidade
Concentrar-se em “commit”	Melhor esforço
Transações aninhadas	Respostas aproximadas
Disponibilidade	Mais simples e mais rápido
Conservador (pessimista)	Agressivo (otimista)
Evolução difícil	Evolução mais fácil

Tabela 1: Correlação entre ACID e BASE.
Fonte: Adaptação de Sudoers (2015)

2.4 Teorema CAP

Em 2000, o professor Brewer (2000) introduziu o Teorema CAP que é um acrônimo (**C**)*onsistency*, (**A**)*vailability* e (**P**)*artitioningTolerance*. Esse conceito consiste numa relação de compromisso entre seus atributos:

a. **Consistência**: Ao finalizar cada transação no banco de dados, o mesmo precisa garantir que a versão mais recente do dado alterado esteja disponível para todas as outras sessões;

b. **Disponibilidade**: É a propriedade que define a disponibilidade de um banco de dados, ou seja, deve garantir que esteja sempre fornecendo acesso a seus usuários;

c. **Partição tolerante à falhas:** Essa propriedade descreve que o banco de dados deve continuar disponível mesmo que parte dele esteja indisponível, seja por falha ou manutenção.

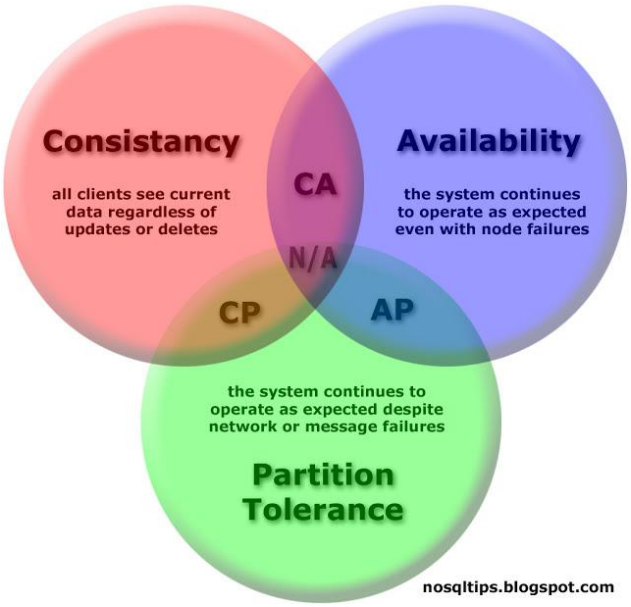

Figura 4: Teorema CAP
Fonte: NoSql Tips and Tricks (2011).

Segundo Brewer (2000), em um sistema computacional distribuído, o ideal é que todas as três propriedades estejam presentes, no entanto, conforme a Figura 4, o Teorema CAP descreve que só é possível empregar duas delas ao mesmo tempo:

1. **CA (Consistência e Disponibilidade):** Sistemas que contêm essas propriedades e que excluem o Particionamento correm o risco de, no caso de uma falha, não conseguirem concluir uma transação por falta de redundância do sistema;
2. **CP (Consistência e Particionamento):** Com essas características e ausência de Disponibilidade, os sistemas correm o risco de ficarem inteiramente sem fornecer acesso aos usuários;
3. **AP (Disponibilidade e Particionamento):** Nesse contexto os sistemas abrem mão da consistência, ou seja, em algum momento, no caso de um incidente, os dados podem ficar inconsistentes.

2.5 Bancos de Dados de Big Data

Segundo Tudorica e Bucur (2011), para suprir a necessidade de processar e armazenar o grande volume de dados com baixo custo de operação e manutenção, surgiram vários Bancos de Dados. Esses Bancos de Dados que utilizam a linguagem NoSQL que significa Not Only SQL (Não Apenas SQL) tem como principal característica a sua estrutura distribuída onde seus dados são distribuídos em vários servidores Cloudera (2020).

Esses Bancos de Dados são denominados não-relacionais e a principal diferença entre eles e os Bancos de Dados relacionais é que, este último, para que um dado seja armazenado, é preciso que seja definida uma estrutura específica capaz de comportar o tipo e o tamanho de dado que será armazenado, ou seja, os atributos do dado precisam ser conhecidos previamente para que se torne possível o seu processamento e armazenamento. Por outro lado, nos Bancos de Dados não relacionais, um determinado dado pode ser armazenado sem haver a necessidade de alterar a estrutura do banco de dados. Dessa forma, os Bancos de Dados não relacionais se tornam mais flexíveis em relação aos relacionais Sadalage e Fowler (2013).

Atualmente existe 4 (quatro) tipos de Bancos de Dados, como pode ser visto na Figura 5, onde cada um deles é designado para aplicações específicas que não podem ser resolvidos com um Banco de Dados relacional. São eles: orientado a documento, orientado a gráfico, orientado a chave-valor e orientado a coluna Medium (2020).

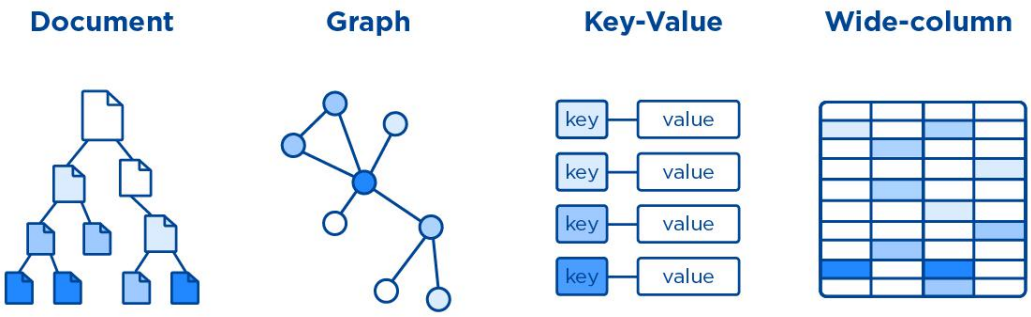

Figura 5: Tipos de bancos de dados NoSQL
Fonte: Medium (2020)

2.5.1 Orientado a Documento

Esse tipo de Banco de Dados é destinado à otimização de leitura e alto desempenho de armazenamento de grandes quantidades de registros. Por outro lado, a escrita e a leitura concorrente com alto desempenho não é o foco, Han et al (2011).

O armazenamento dos dados é feito encapsulando-os em pares de chave-valor em JSON ou em outro formato parecido, como visto na Figura 6. Cada documento tem uma única chave que é utilizada para indexar, identificar e consultá-lo e, como ele não tem uma estrutura de dados definida como há no Banco de Dados relacional, não há schema definido, Hecht e Jablonski (2011).

Key	Document
document-1	{ "id": "1", "name": "John Smith", "isactive":"true", "birthdate":"08/30/1984" }
document-2	{ "id": "2", "fullname": "Sara Walker", "isactive":"false", "birthdate":"02/15/1971" }
document-3	{ "id": "3", "fullname": { "firstname":"Max", "lastname":"Johnson", "middleinitial":"B" } "isactive":"true", "birthdate":"04/02/1964" }

Figura 6: Exemplo de dados do tipo documento
Fonte: BI-Insider (2019)

De acordo com o DB-Engines (2021), os dez principais bancos de dados não relacionais e que suportam orientação a documento, na ordem de popularidade, são os seguintes: MongoDB[1], Amazon DynamoDB[2], Microsoft Azure Cosmos DB[3], Couchbase[4], Firebase Realtime Database, CouchDB, MarkLogic, Realm, google Cloud Firestore, OrientDB.

2.5.2 Orientado a Gráfico

Esse tipo de banco de dados tem seus dados organizados como nodes enquanto que o relacionamento entre eles são representados como conexões entre si, como pode ser visto na Figura 7. Relacionamentos são peças importantes no Banco de Dados orientado a Gráfico que pode representar uma abstração que ficaria difícil fazê-la em um banco relacional, BI-Insider (2019).

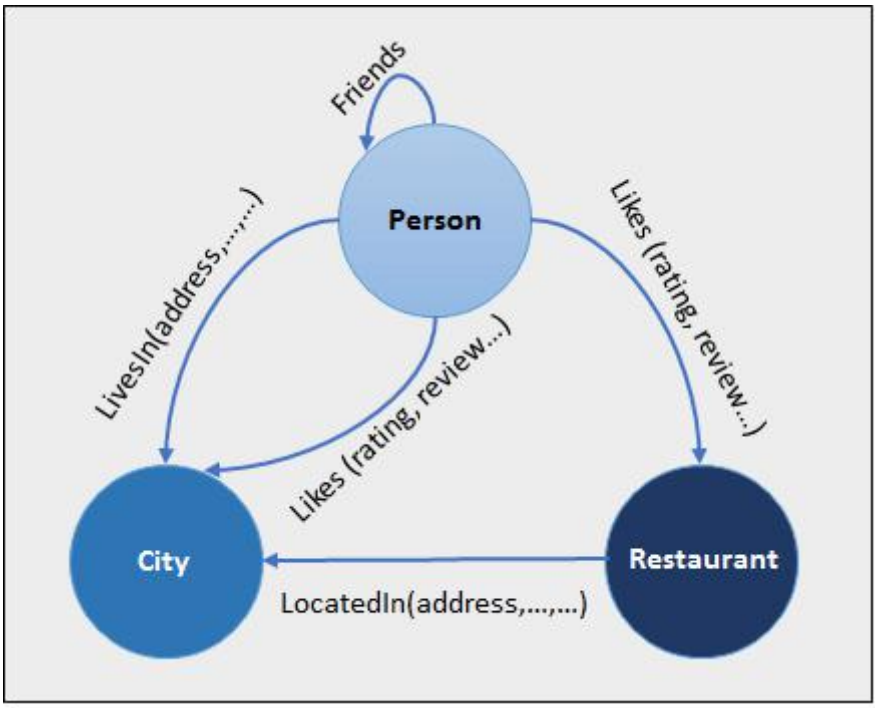

Figura 7: Exemplo de dados do tipo gráfico
Fonte: Medium (2020)

A relação entre os usuários no Twitter são armazenados em um banco de dados não relacional orientado a gráfico, denominado FLockDB. Este banco de dados possui alta performance para armazenar, ler e escrever relações desse tipo, muito profundas, Hecht e Jablonski (2011).

De acordo com o DB-Engines (2021), os dez principais bancos de dados não relacionais e que suportam orientação a gráfico, na ordem de popularidade, são os seguintes: Neo4j, Microsoft Azure Cosmos DB, OrientDB, ArangoDB, JanusGraph, Virtuoso, GraphDB, Amazon Neptune, FaunaDB e Stardog.

2.5.3 Orientado a Chave-Valor

Neste tipo de Banco de Dados, a sua estrutura é organizada com Chave-Valor, ou seja, os dados são armazenados como valor e indexados por um outro valor chave, conforme demonstrado na Figura 8. Uma vez armazenado, a consulta do valor é feita através de sua chave, Hecht e Jablonski (2011).

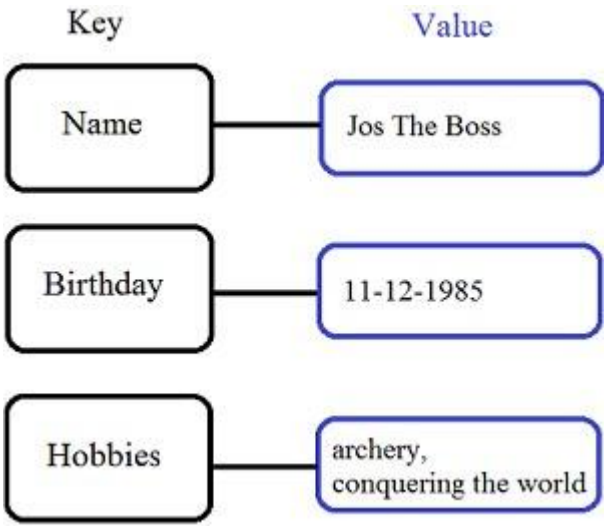

Figura 8: Exemplo de dado do tipo chave-valor
Fonte: Medium (2020)

Num contexto de armazenamento chave-valor, não existe schema e o valor do registro pode ser de tipos variados como: números, strings, binário, imagens, vídeos, JSON, XML, HTML, etc. Esse é o tipo de banco de dados mais flexível entre os NoSQL por permitir que a aplicação tenha um controle completo sobre o que é armazenado, BI-Insider (2019).

De acordo com o DB-Engines (2021), os dez principais bancos de dados não relacionais e que suportam orientação a chave-valor, na ordem de popularidade, são os seguintes: Redis, Amazon DynamoDB, Microsoft Azure Cosmos DB, Memcached, etcd, Hazelcast, Ehcache, Aerospike, OrientDB e ArangoDB.

2.5.4 Orientado a Coluna

Esse tipo de banco de dados armazena registros que contêm uma grande quantidade de colunas dinâmicas. Os dados são armazenados em células agrupadas em colunas de dados ao invés de linhas e, por sua vez, as colunas são logicamente agrupadas em famílias de colunas, BI-Insider (2019). Um exemplo é demonstrado na Figura 9.

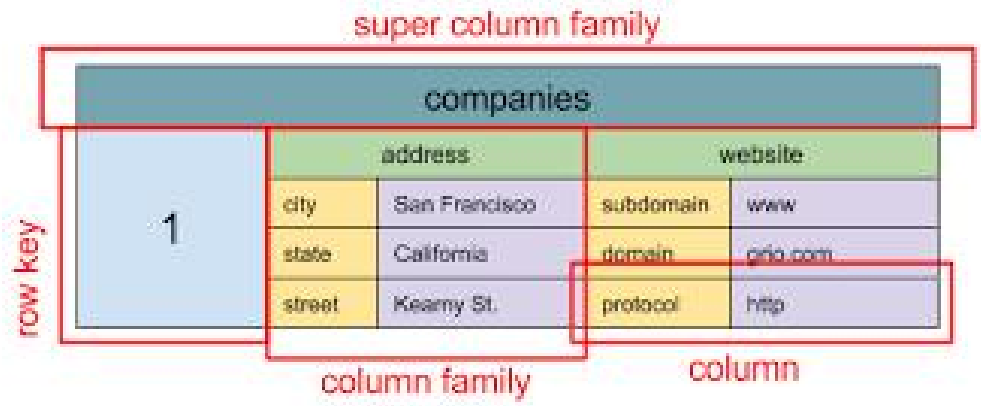

Figura 9: Exemplo de dados orientado a coluna
Fonte: Grio(2015)

Diferente dos bancos de dados relacionais, os bancos de dados orientados a coluna não suportam relacionamento de tabelas, HAN et al (2011), Han et al (2011) descreve algumas caraterísticas desse tipo de banco de dados:

1. Cada coluna contém um índice;
2. Os dados são armazenados em colunas;
3. Todas as colunas têm o mesmo tipo de dados;
4. Têm uma excelente taxa de compressão.

De acordo com o DB-Engines (2021), os dez principais bancos de dados não relacionais e que suportam orientação a coluna, na ordem de popularidade, são os seguintes: Cassandra, HBase, Microsoft Azure Cosmos DB, Datastax Enterprise, Mmicrosoft Azure Table Storage, Accumulo, Google Cloud Bigtable, ScyllaDB, HPE Ezmeral Data Fabric e Elassandra.

2.5.5 Como escolher qual Bancos de Dados NoSQL utilizar?

Segundo Dataversity (2018), os passos para a escolha do melhor banco de dados para se utilizar em uma aplicação é resumida nos seguintes passos:

1. Determinar se precisa utilizar um banco de dados NoSQL ou um Relacional através da análise das propriedades ACID e BASE;
2. Fazendo uso do Teorema CAP, determinar as características entre CA, AP ou CP que satisfazem as necessidades da sua aplicação;
3. Ainda utilizando o Teorema CAP e após definidas as características necessárias, determinar o tipo de banco de dados. Por exemplo, se no passo anterior tiver escolhido o CP como característica da aplicação, de acordo com a figura X, o banco de dados deveria ser do tipo orientado a documento. Caso a escolha tenha sido CA, o tipo de banco de dados seria orientado a gráfico;
4. Uma vez definido o tipo de banco de dados basta escolher qual utilizar dentro do tipo determinado, conforme mostrado na Figura 10.

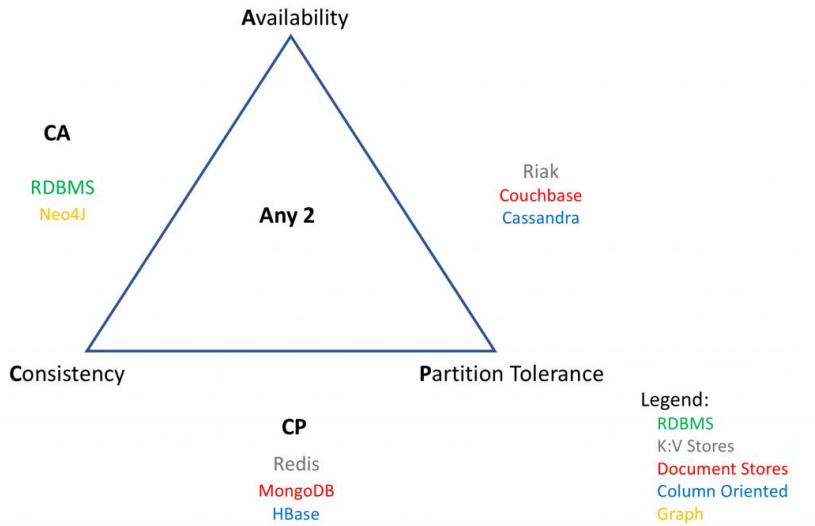

Figura 10: Relação do Teorema CAP
Fonte: Dataversity (2018)

Grandes players no mercado utilizam bancos de dados NoSQL e garantem excelentes resultados, como mostrado na Figura 11. A grande vantagem de conhecermos quais bancos de dados cada uma dessas grandes empresas utilizam, é o fato de sabermos que o banco de dados é confiável. Isso é interessante porque

essas empresas têm uma exigência global e, certamente, o banco de dados é peça fundamental no quebra-cabeças de infraestrutura. Portanto, se traz resultado para as grandes empresas, se for utilizado de forma adequada, trará bons resultados também para as pequenas e médias, Chaudhry et al (2020).

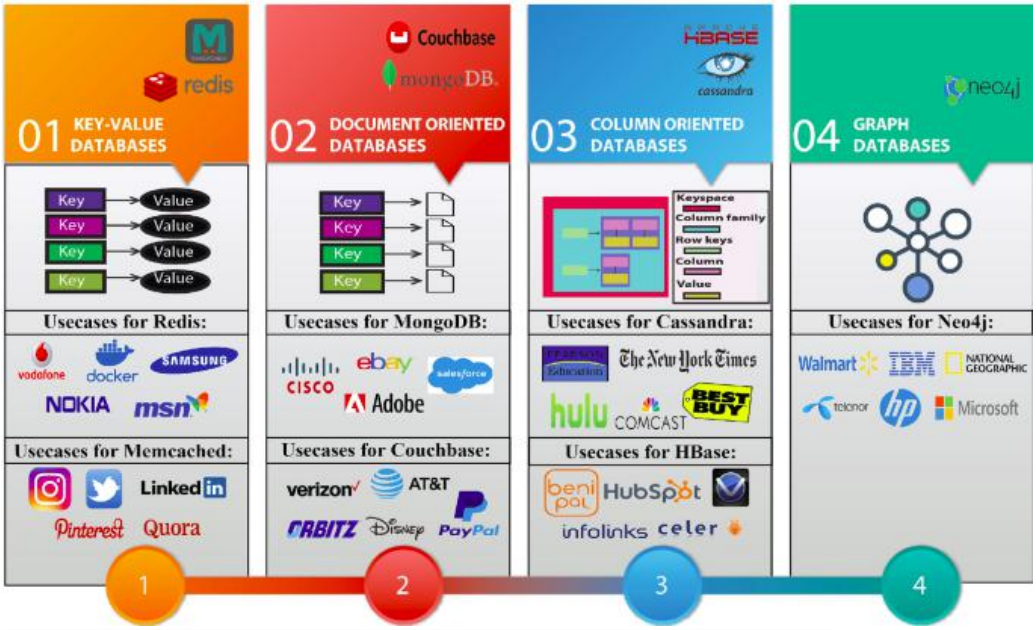

Figura 11: Relação os bancos NoSql com e empresas que os utilizam
Fonte: Chaudhry et al. (2020)

2.5.6 Bancos de Dados NoSQL

2.5.6.1 Redis

Redis é um banco de dados open source baseado na licença BSD, in-memory para armazenar estrutura de dados, usado como banco de dados, cache e message broker. Ele pode trabalhar com os seguintes tipos de dados: strings, hashes, lists, sets, sorted sets com range queries, bitmaps, hyperloglogs, índices geospaciais e streams. Para alcançar a performance, o Redis pode trabalhar com datasets in-memory ou, dependendo do caso, persistir os dados, Redis (2021).

2.5.6.2 CouchBase

Couchbase é um banco de dados NoSQL, open source, distribuído e orientado a documento e também a key-value. Ele oferece suporte a vários padrões de acesso além do modelo JSON que é altamente flexível, SQL amigável e fácil de

usar, totalmente integrável com SDK Java, Scala, .NET, Go, JavaScript e Python. O Couchbase tem se concentrado em estabelecer um alto padrão em suas características para que o resultado entregado seja um conjunto robusto e com desempenho sólido, Couchbase (2021).

2.5.6.3 HBase

O HBase é um banco de dados mantido pela Apache e em conjunto com o Hadoop se torna um sistema de gerenciamento de dados altamente escalável, distribuído e consegue armazenar uma enorme massa de dados. Ele pode ser escalável de forma modular ou linear, possui leitura e escrita consistente, fácil integração com API java. O HBase tem como objetivo suportar tabelas extremamente grandes com bilhões de linhas e colunas hospedados em hardware comum, HBase (2021).

2.5.6.4 Neo4j

O Neo4J é nativamente um banco de dados orientado a gráficos criado inicialmente para aproveitar os dados e os seus relacionamentos. Nesse banco de dados, cada registro ou nó armazena ponteiros (ligações) para todos os outros registros aos quais estão relacionados. O Neo4j é líder no mercado nesse tipo de banco de dados e oferece uma performance de magnitude extraordinária em consultas a dados com relacionamentos complexos. A linguagem dele é o Cypher. Ele também suporta, oficialmente, drivers para Java, .Net, JavaScript, Go e Python no entanto, a comunidade de desenvolvedores já criaram drivers para PHP, Ruby, Erlang, Clojure e C/C++, Robinson et al (2015).

3 Análise dos resultados

Neste artigo foi feito um levantamento histórico dos dados até a atualidade mostrando sua evolução e a correlação deles com a volumetria existente. Também foi analisado os tipos de dados e listadas as classificações segundo suas características. Todo dado gerado, independente da sua origem, há um tipo relacionado a ele como como estruturado, semiestruturado ou não estruturado e,

ainda assim, para cada tipo de dado existe um tipo de banco de dados específico recomendado para armazená-lo, cada um segundo suas características. Foi proposto neste artigo fazer um estudo literário com o objetivo de apresentar a diferença entre banco de dados relacional e não relacional e estabelecer uma relação com os tipos de dados e aplicações específicas.

Os bancos de dados não relacionais também conhecidos como NoSQL são amplamente utilizados em conjunto com Big Data, fornecendo, cada um conforme suas especificações, alta performance, segurança e disponibilidade de acordo com a hipótese do professor Brewer (2000) que propôs o Teorema CAP.

Os quatro tipos de banco de dados NoSQL, são eles: orientado a chave-valor, orientado a documento, orientado a gráfico e orientado a coluna são fundamentais para a existência do Big Data pois, com tantos dados não estruturados sendo gerados constantemente, é preciso ter tecnologia que suporte, de forma eficiente, seu armazenamento.

4 Conclusões

Neste artigo, o qual foi fundamentado na literatura mundial pelo principais autores que promoveram discussões sobre o passado, presente e futuro do Big Data, foi possível identificar e mapear as características essenciais de um sistema integrante deste futuro supracitado. Conforme o tempo passa, estamos chegando cada vez mais perto do estado da arte do Big Data. É possível observar essa trajetória analisando os V's que, inicialmente eram três (Volume, Variedade e Velocidade) McAfee e Brynjolfsson (2012), logo depois foram adicionados mais dois (Veracidade e Valor) Walker (2015) e posteriormente mais um (Variabilidade) Demchenko (2014). Cada V que é acrescentado nas características do Big Data, torna sua definição mais detalhada e consequentemente sua implementação e resultados mais assertivos.

Além disso, também foi possível observar e compreender a importância dos sistemas de bancos de dados, pois, sem dúvidas, é uma peça chave extremamente importante na engrenagem que participa no armazenamento e consulta da enorme quantidade, variedade e velocidade de dados gerados a cada segundo no mundo inteiro. Por esse motivo, a definição do banco de dados adequado para cada

aplicação é fundamental para que o sistema seja seguro, confiável, escalável e não mais importante, com a melhor performance.

Também é possível encontrar neste artigo, uma metodologia para a escolha correta do banco de dados adequado para um sistema específico. Isso é possível mapeando o tipo de dado e as necessidades do sistema envolvido. Como, atualmente, há no mercado uma vasta variedade de SGBDs NoSQL que estão bem maduros, é possível degustar e testar vários deles para obter a melhor experiência que atenda às expectativas de cada sistema.

5 Referências

KAMPAKIS, S. **The Decision Maker's Handbook for Data Science: a Guide for Non Tecnical Executives, Managers and Founders**. London, Apress, 2020.

PEREZ, C. Technological revolutions and techno-economic paradigms. **Cambridge Journal of Economics**, [s.l.], v. 34, n.1, p.185-202, 2010

GRAY, J. **The Transaction Concept: Virtues and Limitations**. Proceedings of the 7th International Conference on Very Large Databases. Cupertino, CA: Tandem Computers, 1981, pp. 144–154.

HÄRDER, T.; REUTER, A. **Principles of transaction-oriented database recovery**. ACM Computing Surveys, 1983, 15 (4). doi:10.1145/289.291

ZANJIREH, M. M., LARIJANI, H. **A Survey on Centralised and Distributed Clustering Routing Algorithms for WSNs**, 2015. IEEE Vehicular Technology Conference. VTC 2015. Glasgow, Escócia.

HERMANN, M.; PENTEK, T.; OTTO, B. **Design Principles for Industrie 4.0 Scenarios**. In: 2016 49TH HAWAII INTERNATIONAL CONFERENCE ON SYSTEM SCIENCES (HICSS). IEEE, 5 jan. 2016

RÜßMANN, M.; LORENZ, M.; GERBERT, P.; WALDNER, M.; JUSTUS, J.; ENGEL, P.; HARNISCH, M. **Industry 4.0: The future of productivity and growth in manufacturing industries**. Boston Consulting Group, v. 9, 2015.

MORAIS, M. O; BREJÃO, A. S; FERIGATTO, E. A; COSTA NETO, P. L. O. **Inovação e Conhecimento como Ferramentas Estratégicas nas Organizações: Estudo de Casos Múltiplos.** Rev. FSA, Teresina, v.15, n.4, art. 9, p. 169-191, jul./ago. 2018.

MAURO, A., GRECO, M., & GRIMALDI, M. **A formal definition of Big Data based on its essential features**. Library Review, 65(3), 122–135. 2016.

McAFEE, A.; BRYNJOLFSSON, E. Big data: **The management revolution**. Harvard Business Review, v. 90, n. 10, p. 60, 2012.

WALKER, R. **From Big Data to big profits: success with data and analytics**. New York: Oxford University Press, 2015.

DEMCHENKO, Y.; GROSSO, P.; DE LAAT, C.; MEMBREY, P. Addressing **Big Data issues in scientific data infrastructure**. Colaboration Technologies ans Systems (CTS), 2013.

FURLAN, P; LAURINDO, F. J. B. **Agrupamentos epistemológicos de artigos publicados sobre big data analytics. Transinformação**, v. 29, n. 1, 2017, p. 91-100. Disponível em: <http://www.scielo.br/pdf/tinf/v29n1/0103-3786-tinf-29-01-00091.pdf>. Acesso em: 15 fev. 2021.

JOYANES, L., **Big Data: Análisis de grandes volúmenes de datos en organizaciones**, Editorial Alfaomega, 2013.

FERREIRA, J. A. ; SANTOS, P. L. V. A. C. . **O modelo de dados Resource Description Framework (RDF) e o seu papel na descrição de recursos. Informação & Sociedade** (UFPB. Online), v. 23, p. 13-23, 2013.

KARL, P., **Moving Media Storage Technologies: Applications &Workflows for Video and Media Server Platforms**, USA: Elsevier, Inc, 2011.

ROBINSON, I.; WEBBER, J. EIFREM, E. . **Graph Databases - New Opportunities for connect data.** O'Reilly, 2015.

BREWER, E. A. **Towards robust distributed systems**. (Invited Talk). Principles of Distributed Computing (PODC), Portland, Oregon, Julho 2000.

TUDORICA, B. G.; BUCUR, C. **A comparison between several NoSQL databases with comments and notes. Proceedings** - RoEduNet IEEE International Conference, 2011. ISSN 20681038.

SADALAGE, P. J.; FOWLER, M. **NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence,** 2013**.** Disponível em: <https://bigdata-ir.com/wp-content/uploads/2017/04/NoSQL-Distilled.pdf>. Acesso em: 16 fev. 2021.

HAN, J., HAIHONG, E., LE, G., DU, J. **Survey on NoSQL database. Proceedings** - 2011 6th International Conference on Pervasive Computing and Applications, ICPCA 2011, pages 363–366.

HECHT, R; JABLONSKI, S. **NoSQL evaluation: A use case oriented survey. In Cloud and Service Computing (CSC)**, International Conference on , 2011, pag. 336 e 341.

ASTERA. **Noções básicas sobre dados estruturados, semiestruturados e não estruturados,** 2021. Disponível em: <https://www.astera.com/pt/type/blog/structured-semi-structured-and-unstructured-data/>. Acesso em: 15 fev. 2021.

LAWTOMATED, **Structured Data vs. Unstructured Data: what are they and why care?**, 2019. Disponível em: <https://lawtomated.com/structured-data-vs-unstructured-data-what-are-they-and-why-care/>. Acesso em: 17 fev. 2021.

ATLAN. **Unustructured Data**, 2020 Disponível em: <https://wiki.atlan.com/unstructured-data/>. Acesso em: 13 fev. 2021.

IBM. **What is big data?,** 2012. Disponível em: <https://developer.ibm.com/technologies/analytics/blogs/what-is-big-data-more-than-volume-velocity-and-variety/>. Acesso em: 15 fev. 2021.

IBM. **The Four V's of Big Data,** 2013 Disponível em: <http://www.ibmbigdatahub.com/infographic/four-vs-big-data>. Acesso em: 15 fev. 2021.

DUMMIES. **ACID versus BASE Data Stores**, 2020 Disponível em: <https://www.dummies.com/programming/big-data/hadoop/acid-versus-base-data-stores/>. Acesso em: 10 fev. 2021

SUDOERS. **NoSQL,** 2015. Disponível em: <http://blog.sudoers.com.br/nosql/>. Acesso em: 11 fev. 2021.

CLOUDERA. **Operational Database NoSQL and Related Capabilities,** 2020. Disponivel em: <https://blog.cloudera.com/operational-database-nosql-and-related-capabilities/>. Acesso em: 13 fev. 2021.

NOSQL TIPS AND TRIKS. **CAP Theorem Diagram for distribution,** 2011. Disponível em: <http://blog.nosqltips.com/2011/04/cap-diagram-for-distribution.html>. Acesso em: 16 fev. 2021.

EDIUM. **4 Types of NoSQL Databases,**2020. Disponível em: <https://medium.com/swlh/4-types-of-nosql-databases-d88ad21f7d3b>. Acesso em: 15 fev. 2015

CHAUDHRY, N., YOUSAF, M.M. **Architectural assessment of NoSQL and NewSQL systems. Distrib Parallel Databases** 38, 881–926 (2020). Disponível em: <https://doi.org/10.1007/s10619-020-07310-1>. Acesso em: 15 fev. 2021.

BI-INSIDER. **Document NoSQL Database,** 2019. Disponível em: <https://bi-insider.com/posts/document-nosql-database/>. Acesso em: 16 fev. 2021.

BI-INSIDER. **Graph NoSQL Database**, 2019. Disponível em: <https://bi-insider.com/posts/graph-nosql-database/>. Acesso em: 16 fev. 2021.

BI-INSIDER. **Key-Value NoSQL Database,** 2019. Disponível em: <https://bi-insider.com/posts/key-value-nosql-database/>. Acesso em: 16 fev. 2021.

BI-INSIDER. **Wide Column / Column Family NoSQL Database,** 2019. Disponível em: <https://bi-insider.com/posts/wide-column-column-family-nosql-database/>. Acesso em: 16 fev. 2021.

DB-ENGINES. **DB-Engines Ranking**,2021. Disponível em: <https://db-engines.com/en/ranking>. Acesso em: 17 fev. 2021.

GRIO. **SQL & NOSQL: A Brief History.** 2015. Disponível em: <https://blog.grio.com/2015/11/sql-nosql-a-brief-history.html>. Acesso em: 17 fev. 2021.

DATAVERSITY. **How to Choose the Right NoSQL Database for Your Application?,**2018**.** Disponível em: <https://www.dataversity.net/choose-right-nosql-database-application/>. Acesso em: 18 fev. 2021.

REDIS. I**ntroduction to Redis,**2021. Disponível em: <https://redis.io/topics/introduction>. Acesso em: 18 fev. 2021.

COUCHBASE. **Enterprise-Class NoSQL Cloud-to-Edge database**, 2021. Disponível em: <https://www.couchbase.com/>. Acesso em: 18 fev. 2021.

HBASE. **Welcome to Apache HBase**,2021. Disponível em: <https://hbase.apache.org/index.html>. Acesso em: 19 fev. 2021.