

LIGA DE ENSINO DO RIO GRANDE DO NORTE
CENTRO UNIVERSITÁRIO DO RIO GRANDE DO NORTE
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

EMMANUEL RICARDO CALIXTO

**TESTE DE DESEMPENHO EM BANCO DE DADOS NOSQL E A RELAÇÃO COM
BIG DATA**

Natal/RN

2016

EMMANUEL RICARDO CALIXTO

**TESTE DE DESEMPENHO EM BANCO DE DADOS NOSQL E A RELAÇÃO COM
BIG DATA**

Trabalho de Conclusão de Curso
(Bacharelado em Sistemas de
Informação) apresentado ao Centro
Universitário do Rio Grande do Norte
(UNI-RN) como requisito final para
obtenção do título de Bacharel em
Sistemas de Informação.

Orientador: Prof. M.Sc. Romulo
Fagundes Cantanhede.

Natal/RN

2016

Catálogo na Publicação – Biblioteca do UNI-RN
Setor de Processos Técnicos

Calixto, Emmanuel Ricardo.

Teste de desempenho em banco de dados NoSQL e a relação com Big Data / Emmanuel Ricardo Calixto. – Natal, 2016.
47 f.

Orientador: Prof. M.Sc. Romulo Fagundes Cantanhede.

Monografia (Bacharelado em Sistemas de Informação) – Centro
Universitário do Rio Grande do Norte.

1. NoSQL – Monografia. 2. Volume – Monografia. 3. Dados –
Monografia. 4. Big Data – Monografia. I. Cantanhede, Romulo Fagundes.
II. Título.

RN/UNI-RN/BC

CDU 004

EMMANUEL RICARDO CALIXTO

**TESTE DE DESEMPENHO EM BANCO DE DADOS NOSQL E A RELAÇÃO COM
BIG DATA**

Trabalho de Conclusão de Curso
(Bacharelado em Sistemas de
Informação) apresentado ao Centro
Universitário do Rio Grande do Norte
(UNI-RN) como requisito final para
obtenção do título de Bacharel em
Sistemas de Informação.

Aprovado em ____/____/____

BANCA EXAMINADORA

Prof. M.Sc. Romulo Fagundes Cantanhede
Orientador

Prof. M.Sc. Alexandre Luiz Galvão Damasceno
Membro

Prof. Esp. Gilles Veleneuve Trindade Silvano
Membro

RESUMO

O mundo encontra-se cada vez mais conectado. Tal cenário, inédito até então, aproxima pessoas e reduz distâncias, mas, em contrapartida, traz um aumento exponencial na geração de dados. Redes sociais, câmeras de segurança, sites de vídeos e dispositivos de monitoramento são alguns dos responsáveis pelo aumento vertiginoso dos dados gerados na web. Em virtude da variedade de fontes de onde estes dados são gerados, grande parte deles possuem natureza não estruturada e, por isso, não podem ser adequadamente tratados por bancos de dados convencionais. A análise desse grande volume de dados, que funciona como uma espécie termômetro social, pode trazer para uma empresa um diferencial competitivo, ajudando assim no processo de tomada de decisão. Os bancos de dados NoSQL surgiram com o objetivo de preencher essa lacuna deixada pelos bancos de dados relacionais, fornecendo capacidade de tratamento de dados não estruturados, velocidade de manipulação de grandes volumes de dados e suporte nativo a replicação de dados. O objetivo desse trabalho é mostrar o cenário de crescimento exponencial de dados do Big Data bem como mostrar como os bancos de dados NoSQL estão inseridos nesse contexto. Este estudo foi feito, metodologicamente falando, no tocante à natureza, através de uma pesquisa básica, utilizando um objetivo exploratório, por meio de uma abordagem qualitativa e quanto aos procedimentos, por meio de uma pesquisa bibliográfica. De forma complementar, foi realizado nesse estudo um teste comparativo entre dois bancos de dados NoSQL orientados a documentos, o CouchDB e o MongoDB.

Palavras-chave: NoSQL. Volume. Dados. Big Data.

ABSTRACT

The world is more and more connected. Such a scenario, previously unheard of, people and reduce distances, but, on the other hand, an exponential increase in data generation. Social networking, security cameras, video sites and monitoring devices are some of those responsible for the dizzying rise of data generated on the web. Due to the variety of sources from which this data is generated, most of them are not structured and therefore can not be adequately handled by conventional databases. The analysis of large data, which functions as a kind of social thermometer, can bring a company a competitive differential, thus helping the decision-making process. The NoSQL databases aim to fill the gap left by relational databases, the ability to handle unstructured data, the speed of handling large volumes of data and support native data replication. The objective of this work is to show the scenario of exponential data growth. NoSQL are inserted in the context. This study was made, methodologically speaking, not about nature, through a basic research, using an exploratory objective, through a qualitative and procedural approach, through a bibliographical research. In a complementary way, this study was compared between two NoSQL document-oriented databases, CouchDB and MongoDB.

Keywords: NoSQL. Volume. Data. Big Data.

LISTA DE ILUSTRAÇÕES

Figura 1 – Teorema CAP	26
Figura 2 – Estrutura de shards	28
Figura 3 – Banco de dados chave-valor	29
Figura 4 – Banco de dados de colunas	30
Figura 5 – Banco de dados de grafos.....	30
Figura 6 – Número de usuários ativos.....	33
Figura 7 – Tempos de resposta COUCHDB.....	33
Figura 8 – Tempos de resposta MONGODB.....	34

LISTA DE ABREVIATURAS E SIGLAS

TI	Tecnologia da Informação
OLAP	Online Analytical Processing
PHP	Hypertext Preprocessor
RFID	Radio Frequency Identification
API	Application Programming Interface
SGBD	Sistema Gerenciador de Banco de Dados
HTTP	Hypertext Transfer Protocol
XML	Extensible Markup Language.

SUMÁRIO

1 INTRODUÇÃO	9
2 REFERENCIAL TEÓRICO	10
2.1 SQL (STRUCTURED QUERY LANGUAGE)	10
2.2 DATA WAREHOUSE	10
2.3 DATA MART	11
2.4 DATA MINING (MINERAÇÃO DE DADOS)	11
2.5 BI (BUSINESS INTELIGENCE)	12
2.6 IOT (INTERNET OF THINGS – INTERNET DAS COISAS)	13
2.7 CLOUD COMPUTING (COMPUTAÇÃO EM NUVEM)	13
2.7.1 SaaS (Software as a Service)	14
2.7.2 PaaS (Platform as a Service)	15
2.7.3 IaaS (Infrastructure As A Service)	15
2.8 JAVASCRIPT	16
2.8.1 JSON (Javascript Object Notation)	16
2.9 MONGODB	17
2.10 COUCHDB	18
2.11 POSTGRESQL	18
3 METODOLOGIA	20
4 PROBLEMA	21
5 NOSQL	22
5.1 ESCALABILIDADE HORIZONTAL	22
5.2 AUSÊNCIA DE ESQUEMA OU ESQUEMA FLEXÍVEL	23
5.3 PERMITE REPLICAÇÃO DE FORMA NATIVA	23
5.4 CONSISTÊNCIA EVENTUAL	23
5.5 ACID E BASE	23
5.5.1 ACID	24
5.5.2 BASE	24
5.6 QUAL A RELAÇÃO DE NOSQL COM BIGDATA?	25
5.7 TEOREMA CAP	25
5.7.1 Consistência	25
5.7.2 Disponibilidade	26
5.7.3 Tolerância à partição	26

5.8 TIPOS DE BANCOS NOSQL	27
5.8.1 Bancos de dados orientados a documentos	27
5.8.2 Bancos de dados chave-valor	28
5.8.3 Bancos de dados orientados a colunas	29
5.8.4 Bancos de dados orientados a grafos.....	30
6 TESTE DE DESEMPENHO	31
6.1 COMPARATIVO ENTRE DUAS SOLUÇÕES NOSQL (MONGODB E COUCHDB)	31
6.1.1 Máquina usada nos testes	31
6.1.2 Softwares usados nos testes	31
6.1.3 Configurações do JMeter	31
6.1.4 Estrutura da nota fiscal.....	32
6.1.5 Gráficos dos testes	32
6.1.5.1 COUCHDB	33
6.1.5.2 MONGODB	34
6.2 RESULTADO DOS TESTES.....	34
7 O QUE É BIG DATA?	35
7.1 OS VS DO BIG DATA	35
7.1.1 Volume	35
7.1.2 Velocidade	35
7.1.3 Variedade	35
7.2 QUE DECISÕES PODEM SER TOMADAS?	36
7.3 PRIVACIDADE	36
8 QUAIS SOLUÇÕES EXPLORAM ESSES CONCEITOS?	37
9 CONCLUSÕES	38
REFERÊNCIAS.....	39
ANEXO A – CÓDIGO PHP PARA INSERÇÃO DE DADOS NO COUCHDB.....	44
ANEXO B – CÓDIGO PHP PARA INSERÇÃO DE DADOS NO MONGODB	46

1 INTRODUÇÃO

Segundo informações do Google, eram criados a cada dois dias, no segundo semestre de 2013, 5 exabytes de dados. Para efeito comparação, essa quantidade, 5 exabytes, correspondem à toda informação criada desde o início da civilização até o ano de 2003 (BRETERNITZ; SILVA, 2013).

Considerando que o mundo possui cerca de sete bilhões de habitantes e que desse total, seis bilhões de pessoas possuem celulares, é possível mensurar a gigantesca quantidade de informações que pode ser gerada (RIBEIRO, 2014). Em meados dos anos 2000, apenas 25% de todos os dados existentes no planeta estavam sob forma digital, mais tarde, em 2013, o percentual de dados digitalizados havia aumentado para 98% (VOLPATO; RUFINO; DIAS, 2014).

Contribuindo para o aumento exponencial dos dados gerados estão tecnologias como RFID, tecnologias embarcadas, sensores e sites de vídeos (BRETERNITZ; SILVA, 2013). Além disso, câmeras de monitoramento em prédios, avenidas, lojas e ruas (DAVENPORT apud RIBEIRO, 2014). A IOT (Internet das Coisas) se mostra como uma promessa de integração de tecnologias e comunicação distribuindo inteligência e promovendo a cooperação e integração entre as tecnologias (ATZORI; IERA; MORABITO apud BRETERNITZ; SILVA; LOPES, 2013). Tal variedade de fontes de dados reflete as suas diferentes características, como o fato de grande parte não estar estruturada, não sendo possíveis de serem tratadas pelos sistemas convencionais (LOHR apud BRETERNITZ; SILVA, 2013). As mídias sociais também têm um grande papel nessa expansão de dados. De acordo com [TAURION apud VOLPATO; RUFINO; DIAS, 2014), o Facebook, por exemplo, realiza o upload de, no mínimo, 10 milhões de novas fotos por hora.

As informações acima demonstram o crescimento vertiginoso da informação no cenário mundial nos últimos anos. Com tanta informação disponível, torna-se necessário para as empresas realizar, em tais informações, uma análise para tomar decisões com base em indicadores e não em suposições.

2 REFERENCIAL TEÓRICO

2.1 SQL (STRUCTURED QUERY LANGUAGE)

No entendimento de DATE apud Farias (2009) a Structured Query Language, também conhecida pela sigla SQL, pode ser definida como uma linguagem que tornou-se padrão para interagir com bancos de dados relacionais. Tal linguagem foi criada no Centro de Pesquisa Almaden, localizado em San Jose (Califórnia, Estados Unidos).

Para Taylor (2010, p. 7):

[...] SQL é uma linguagem que foi projetada pensando em bases de dados. Através do SQL as pessoas podem criar bancos de dados, adicionar dados a ele, manter dados no banco de dados e requisitar partes dos dados armazenados. Introduzido em 1970, o SQL tem crescido e avançado ao longo dos anos para se tornar um padrão da indústria [...].

Com base nas definições apresentadas é possível concluir que o SQL é uma linguagem criada com o propósito exclusivo de gerenciar bases de dados relacionais, ou seja, através do SQL, que é um padrão da indústria para bancos de dados relacionais, podemos fazer a interação com o banco de dados e assim realizar todas as operações desejadas.

2.2 DATA WAREHOUSE

O Data Warehouse, ou armazém de dados, é um de banco de dados criado a partir de diversas fontes de dados da empresa, como bancos de dados, documentos de texto, planilhas, etc. Essas informações são todas consolidadas, de maneira otimizada, em um único local com o objetivo de realizar consultas usando o máximo de desempenho. O Data Warehouse tem grande importância no processo de tomada de decisão da empresa, pois, através dos relatórios que podem ser extraídos dele, é possível realizar estimativas bem como comparações de desempenho.

Shams (2001 apud ALMEIDA, 2016) define o Data Warehouse como uma estrutura onde ficam, devidamente organizados, todos os dados da organização. Tal estrutura é criada para subsidiar, através de muitos tipos de relatórios analíticos

complexos, o processo de apoio à tomada de decisão.

Um Data Warehouse pode ser definido como um banco de dados que possui dados, selecionados e otimizados para a realização de consultas e não de transações. Tais dados, que são extraídos do ambiente de produção de uma empresa, podem ser consolidados com dados vindos de planilhas, documentos de texto, etc. (INMON apud HOKAMA, 2004).

2.3 DATA MART

O Data Mart é um banco de dados que, assim como o Data Warehouse, possui dados da organização com o objetivo de realizar consultas. O que diferencia o Data Mart do Data Warehouse é que o primeiro possui um caráter mais departamental, ou seja, o Data Mart é restrito a um determinado setor ou departamento. A vantagem desse modelo mais restrito que um Data Mart possui é o seu baixo custo de implementação se comparado a um Data Warehouse.

No entendimento de Sislaghi (2006, p. 5):

[...] Data Mart é um conjunto de dados customizados para atender a necessidades de análises setoriais, específicas de uma área ou processo do negócio da empresa. Algumas organizações optam pelo desenvolvimento de Data Marts não apenas por causa do custo mais baixo ou do menor tempo para implementação, mas também em função dos correntes avanços tecnológicos [...].

Vas, Exe e Inm apud Moraes (2000) falam que os Data Marts funcionam como depósitos de dados com maior especialização, sendo restritos a um tipo de assunto ou departamento. Os Data Marts possibilitam o cruzamento de diversas possibilidades de dados permitindo, dessa forma, uma análise mais aprimorada.

2.4 DATA MINING (MINERAÇÃO DE DADOS)

Data Mining, também conhecido mineração de dados, é uma técnica usada para o descobrimento de novas informações e correlações entre as informações presentes, geralmente, em um Data Warehouse. A técnica de mineração de dados geralmente é aplicada em grandes quantidades de dados e, através de algoritmos específicos, pode encontrar correlações que antes não podiam ser observadas,

ajudando também no processo de tomada de decisões.

Para Cardoso e Machado (2008), Data Mining constitui-se como um procedimento usado como uma etapa da descoberta de conhecimento em bancos de dados. Esta técnica possibilita a revelação de conhecimento incluído em grandes bancos de dados corporativos. Através do uso do Data Mining é possível realizar a análise antecipada de eventos, prevendo tendências e ajudando gestores em suas decisões.

No entendimento de Nimer e Spandri apud Lemos, Steiner e Nievola (2005), a Mineração de dados se vale de técnicas de reconhecimento de padrões, estatísticas e matemáticas para revelar, em grandes quantidades de dados contidas em um Data Warehouse, novas correlações, padrões e tendências.

2.5 BI (BUSINESS INTELLIGENCE)

Os conceitos de BI e de Data Mining, em um primeiro momento, parecem muito similares. Porém, após uma observação mais aprimorada é possível perceber que o BI possui um foco mais voltado ao monitoramento de indicadores, como temperatura e custo, por exemplo. Já o Data Mining busca a identificação de padrões de comportamento de dados, elaborando novos indicadores de análise para o BI.

Soluções de BI (Business Intelligence) podem ser definidas como aquelas que possibilitam extrair informações fundamentais do negócio em uma grande massa de dados. As informações geradas por uma ferramenta de BI oferecem um diferencial competitivo, pois, podem levar à antecipação de tendências. No rol de possibilidades de análise e verificação, pode ser destacado as estatísticas de vendas sobre cada produto da empresa ou relatórios com informações detalhadas sobre o comportamento e preferências dos clientes (CARVALHO, 2000).

De acordo com Jamil (2000, p. 50):

[...] Por Business intelligence compreendemos técnicas, métodos e ferramentas que possibilitam ao usuário analisar dados e, com base nessas análises, emitir respostas que possam subsidiar objetiva e confiavelmente os processos de decisão numa empresa. [...].

2.6 IOT (INTERNET OF THINGS – INTERNET DAS COISAS)

A Internet das coisas é caracterizada pela interatuação de dispositivos comuns do dia a dia, como carros e eletrodomésticos conectados à Internet, mandando e recebendo informações e reagindo de acordo com o ambiente em que se encontram. Os dispositivos interatuantes que fazem parte da Internet das Coisas possuem uma certa autonomia, podendo estes trabalhar praticamente sozinhos, gerando informações de acordo com o que acontece no ambiente onde estão. Essa autonomia, bem como a variedade de dispositivos de IOT existentes, como câmeras e sensores, por exemplo, são também responsáveis pelo grande volume de dados gerados a cada dia. A variedade de tipos de dispositivos da IOT é responsável pela geração de dados, em grande parte, não estruturados.

No entendimento de Bilinski (2014, p. 1): “[...] A Internet das coisas ou IoT, objetiva ligar todos os equipamentos eletrônicos que usamos no dia a dia a internet ou bases de dados e com o uso de redes de sensores, processar essas informações e retornar benefícios aos usuários [...]”.

De acordo com Cervantes (2014), a Internet das coisas (IoT) é caracterizada por ser um tipo de rede híbrida e aberta, além disso, pode ser considerada heterogênea, pois possui os mais diversos tipos de dispositivos nela conectados, tais como eletrodomésticos, sensores e canetas, por exemplo, atuando com computadores, celulares, entre outros. Há uma troca de informações e recursos, sendo esses dispositivos capazes de agir e reagir conforme mudanças ocorridas no ambiente em que estão.

2.7 CLOUD COMPUTING (COMPUTAÇÃO EM NUVEM)

Cloud Computing, ou computação em nuvem, pode ser definido como um conjunto de serviços acessíveis, geralmente baseados em virtualização e operando com base na infraestrutura da Internet, que podem ser contratados seguindo o modelo de pagamento somente pelo que se usa. Além da vantagem do baixo custo, a computação em nuvem possui escalabilidade de recursos, seja de processamento ou armazenamento.

Segundo Vaquero apud Verdi (2010), Cloud Computing é uma coleção de

recursos como hardware, plataformas de desenvolvimento e serviços, acessíveis e de fácil uso, com a característica de que esses recursos podem se ajustar de acordo com a demanda, permitindo assim, uma otimização desses recursos e um modelo de pagamento com base nos recursos que foram efetivamente usados.

No entendimento de Buyya apud Sousa, Moreira e Machado (2009), Computação em nuvem tem o propósito de oferecer serviços de TI sob demanda, sendo o pagamento feito de acordo com o uso.

2.7.1 SaaS (Software as a Service)

No entendimento de Rushell, Zanotto e Mota (2010, p. 7):

[...] Podemos dizer que o SaaS, representa os serviços de mais alto nível disponibilizados em uma nuvem. Esses serviços representam as aplicações completas que são oferecidas aos usuários. Os prestadores de serviços disponibilizam o SaaS na camada de aplicação, o que leva a rodar inteiramente na nuvem e pode ser considerado uma alternativa a rodar um programa em uma máquina local, assim o SaaS traz a redução de custos, dispensando a aquisição de licença de softwares. Colocamos como exemplo de SaaS, sistemas de banco de dados e processadores de textos [...].

Para Sousa, Moreira e Machado (2009, p. 7):

[...] O modelo de SaaS proporciona sistemas de software com propósitos específicos que estão disponíveis para os usuários através da Internet. Os sistemas de software são acessíveis a partir de vários dispositivos do usuário por meio de uma interface thin client como um navegador Web. No SaaS, o usuário não administra ou controla a infraestrutura subjacente, incluindo rede, servidores, sistemas operacionais, armazenamento ou mesmo as características individuais da aplicação, exceto configurações específicas. Com isso, os desenvolvedores se concentram em inovação e não na infraestrutura, levando ao desenvolvimento rápido de sistemas de software [...].

O SaaS, ou software como serviço, pode ser entendido como um serviço que utiliza a infraestrutura da nuvem, através da Internet, para disponibilizar aplicações completas que rodam, não no computador do usuário, mas, na nuvem, gerando assim, economia de processamento e memória na máquina do usuário. Baseado na computação em nuvem, e seguindo seu modelo de pagamento pelo uso, o SaaS também proporciona a economia com licenças de software, pois, o usuário paga somente pelo que usar, ficando as licenças geralmente por conta da

empresa que oferece a estrutura de SaaS.

2.7.2 PaaS (Platform as a Service)

O PaaS, ou plataforma como serviço, é uma plataforma de desenvolvimento e teste de softwares que utiliza a nuvem como infraestrutura. Nesta plataforma, que disponibiliza sistema operacional, ambiente de desenvolvimento e infraestrutura de rede e servidores, o desenvolvedor não precisa se preocupar com administração da infraestrutura, ficando focado no desenvolvimento de suas aplicações.

De acordo com Ruschell, Zanotto e Mota (2010, p. 8):

[...] O PaaS tem por objetivo facilitar o desenvolvimento de aplicações destinadas aos usuários de uma nuvem, criando uma plataforma que agiliza esse processo. O PaaS oferece uma infra-estrutura de alto nível de integração para implementar e testar aplicações na nuvem. Também fornece um sistema operacional, linguagens de programação e ambientes de desenvolvimento para as aplicações, auxiliando a implementação de softwares, já que contém ferramentas de desenvolvimento e colaboração entre desenvolvedores [...].

Para Sousa, Moreira e Machado (2009, p. 9) definem que:

[...] A PaaS oferece uma infraestrutura de alto nível de integração para implementar e testar aplicações na nuvem. O usuário não administra ou controla a infraestrutura subjacente, incluindo rede, servidores, sistemas operacionais ou armazenamento, mas tem controle sobre as aplicações implantadas e, possivelmente, as configurações das aplicações hospedadas nesta infraestrutura [...].

2.7.3 IaaS (Infrastructure as a Service)

A IaaS, ou infraestrutura como serviço, é a base para o funcionamento do PaaS e SaaS, pois fornece a infraestrutura de computação necessária. Tal infraestrutura se traduz, por exemplo, em servidores, armazenamento, roteadores e podem ser gerenciados através de uma interface de administração.

Ruschell, Zanotto e Mota (2010, p. 8) afirmam que:

[...] O IaaS traz os serviços oferecidos na camada de infra-estrutura, nestes serviços podemos incluir servidores, roteadores, sistemas de armazenamento e outros recursos de computação. Também é responsável por prover toda a infra-estrutura necessária para a SaaS e o PaaS. O IaaS

traz algumas características, como uma interface única para administração da infra-estrutura, a aplicação API (Application Programming Interface) para interação com hosts, switches, roteadores e o suporte para a adicionar novos equipamentos de forma simples e transparente. [...]

Sousa, Moreira e Machado (2009) dizem que IaaS é a infraestrutura onde funcionam serviços como PaaS e SaaS. O IaaS possui uma interface de administração e uma API para comunicação com switches e roteadores. Dessa forma, nesta infraestrutura, o usuário realiza o controle sobre os sistemas operacionais usados, armazenamento e, esporadicamente, alguns componentes de rede como, por exemplo, um firewall.

2.8 JAVASCRIPT

A linguagem de programação JavaScript roda, na grande maioria dos casos, no navegador do usuário. Seu uso mais comum é para criar diversas ações na página web e assim melhorar a experiência de usuário. Além disso, através do JavaScript, é possível realizar requisições assíncronas no servidor. No contexto de Big Data, o JavaScript é fundamental, pois, grande parte dos bancos de dados NoSQL existentes usam, para o armazenamento de dados, o formato JSON, que é uma notação de objetos do JavaScript.

Segundo Niederauer (2007, p. 41) “JavaScript é uma linguagem de programação que roda no lado cliente (navegador do usuário) e nos permite realizar determinadas ações dentro de uma página web, melhorando a interatividade com o usuário”.

De acordo com Silva (2010, p. 23):

[...] JavaScript é uma linguagem desenvolvida para rodar no lado do cliente, isto é, a interpretação e o funcionamento da linguagem dependem de funcionalidades hospedadas no navegador do usuário. Isso é possível porque existe um interpretador JavaScript hospedado no navegador [...].

2.8.1 JSON (Javascript Object Notation)

O JSON oferece uma alternativa simples ao formato XML, para intercâmbio de informações. Ambos os formatos são utilizados para o armazenamento e troca de informações entre sistemas diferentes, sendo esses formatos independentes da

plataforma ou mesmo linguagem de programação. O JSON possui algumas vantagens como a simplicidade de uso e, por ser uma notação de objetos JavaScript, a total integração com sistemas que usem tal linguagem. O formato JSON também é importante quanto à persistência de dados em bancos de dados NoSQL, já que sua estrutura é adequada à natureza heterogênea dos dados presentes no Big Data.

No entendimento de Silva (2010, p. 28):

[...] JSON é a sigla para JavaScript Object Notation, ou Notação para Objetos em JavaScript. Trata-se de uma técnica JavaScript que usa sinaxe leve e simples para armazenar dados. Tal como XML, JSON está estruturado de modo a ser facilmente lido, interpretado e processado por máquinas [...].

Para Frozza e Dalassen (2016, p. 6-7):

[...] O JSON é formato de texto completamente independente de linguagem, de simples formatação para troca de dados, fazendo com que a leitura e escrita por humanos seja de fácil compressão e reprodução, e de fácil interpretação para 7 máquinas. Estas propriedades estas que fazem do JSON um formato ideal para a troca de dados [...]

2.9 MONGODB

Para Walter e Duarte (2016), o MongoDB é um SGBD do tipo NoSQL, sendo orientado a documentos. O MongoDB é capaz de gerenciar uma grande massa de dados pois possui, dentre vários recursos, a escalabilidade horizontal.

Boaglio (2015, p. 11) define MongoDB da seguinte forma:

[...] O MongoDB é um document database (banco de dados de documentos), mas não são os documentos da família Microsoft, mas documentos com informações no formato JSON. A ideia é o documento representar toda a informação necessária, sem a restrição dos bancos relacionais [...].

No entendimento de Schroeder e Santos (2016, p. 3):

[...] O MongoDB é 'um banco de dados orientado a documentos que provê alta performance, alta disponibilidade e fácil escalabilidade' (MongoDB Manual, 2013). Seus dados são representados através de JSON (JavaScript Object Notation). Em comparação aos SGBDR (Sistema Gerenciador de Banco de Dados Relacional) algumas analogias podem ser feitas: uma

tabela pode ser comparada com uma coleção e um documento pode ser comparado a uma linha de dados. A escolha do banco de dados MongoDB como solução NoSQL se deve ao grande grau de maturidade que o mesmo já possui, bem como por uma maior adoção pela comunidade [...].

2.10 COUCHDB

O CouchDB é um SGDB, caracterizado por ser livre de esquema e baseado em documentos que é adequado para o trabalho em servidores que possuem hardware de baixo desempenho pois possui foco em tolerância a falhas e, além disso, mesmo executando em hardware sujeito à falhas, oferece escalabilidade, disponibilidade e confiabilidade. Esse SGBD, que utiliza documentos para armazenar os dados, faz uso da estrutura JSON nesses documentos. Para persistência de dados, é usada uma estrutura que tem como base a estrutura de árvores B+ (MACÊDO et al., 2010).

No entendimento de Matteussi e Minetto (2016, p. 4):

[...] O couchDB combina um modelo intuitivo de armazenagem em documentos com um poderoso sistema de pesquisa acessível via RESTful e (HTTP/JSON Api) , dentre suas funcionalidades está a replicação bidirecional com detecção de falhas e resolução de conflitos; sua base de dados é baseada em princípios MapReduce e em vez de armazenar dados em linhas e colunas, o CouchDB gerencia uma coleção de documentos, os chamados 'JSON documents' [...].

Segundo Couch (2013 apud FERRARI et al., 2016, p. 2): “[...] é um banco de dados não relacional, utilizado principalmente para web. Basicamente ele utiliza documentos para definir uma estrutura no banco, armazenando uma chave associada ao documento [...]”.

2.11 POSTGRESQL

O PostgreSQL é um sistema gerenciador de banco de dados relacional de código aberto e multiplataforma. Possui mais de uma década de desenvolvimento e, apesar de ser livre, suporta muitos recursos presentes nos sistemas gerenciadores de bancos de dados comerciais.

Tonini e Frank definem (2010, p. 2):

[...] PostgreSQL é um sistema gerenciador de banco de dados de código-fonte aberto, multiplataforma, com mais de 15 anos de desenvolvimento. Segue o padrão SQL99 e seu gerenciador de transações garante as propriedades atomicidade, consistência, isolamento e durabilidade (ACID).

Bogorny (2002, p. 2) explica que:

[...] O banco de dados PostgreSQL, por exemplo, é um Sistema Gerenciador de Banco de Dados (SGBD) objeto-relacional que roda em diversas plataformas e disponibiliza praticamente todos os recursos dos SGBD proprietários. [...].

3 METODOLOGIA

Levando em conta o aspecto metodológico, quanto à natureza, foi feita uma pesquisa básica, com objetivo exploratório e utilizando a abordagem qualitativa. Quanto aos procedimentos, foi feita uma pesquisa bibliográfica. Além disso, foi utilizada uma ferramenta de teste de estresse, chamada Jmeter, para observar o desempenho do MongoDB e CouchDB, frente à grande quantidade de requisições.

4 PROBLEMA

Bernardo e Borlone (2015, p. 20) dizem que:

[...] O mundo está cada vez mais interligado. Novas tecnologias vêm sendo criadas para conectar e diminuir a distância entre as pessoas. Com a ascensão das tecnologias há um crescimento exponencial na geração de dados variados, estruturados ou não [...].

No Facebook, 350 milhões de imagens são adicionadas pelos seus 1,15 bilhão de usuários, diariamente. O acesso de 751 milhões desses usuários é feito através de dispositivos móveis. No Twitter, um total diário de 400 milhões de publicações são geradas pelos seus 500 milhões de usuários cadastrados. O Google Plus, rede social do Google, já alcançou 500 milhões de usuários enquanto que o LinkedIn, rede social voltada para contatos profissionais, está perto da marca de 250 milhões de cadastrados. Já o Instagram, que possui 130 milhões de usuários, atingiu aproximadamente 16 bilhões de fotografias sendo estas fotografias responsáveis por 1000 comentários por segundo. Cabe lembrar que a rede social Instagram registra o envio diário de 5 milhões de fotografias (SOCIAL MEDIA TODAY apud SILVA, 2015).

De acordo com o site e Observatório da Imprensa apud Silva (2015), passaram pelas antenas de telefonia móvel em todo o mundo, mensalmente, aproximadamente 1,5 trilhão de exabytes por mês.

Há uma clara tendência de aumento na geração de dados a cada ano. Isso faz com que haja um aumento na demanda por ferramentas e recursos de ordem tecnológica capazes de lidar com essa nova realidade. Tais ferramentas e recursos devem ser capazes de analisar toda essa massa variada de dados e, em tempo hábil, apresentar uma resposta (SILVA, 2015).

5 NOSQL

Segundo Tiwari (2011), o NoSQL não é um produto ou tecnologia, mas, um termo que serve pra categorizar os bancos de dados que possuem outro paradigma, pois não seguem as regras presentes em bancos de dados relacionais e que lidam, de forma muito rápida, com grandes quantidades de dados.

Existem muitos bancos de dados NoSQL no mercado e a escolha de cada um dependerá de qual problema deverá ser resolvido. Os bancos NoSQL mais comuns são os orientados a documento, chave-valor, orientados a objetos e os de grafos (BOAGLIO, 2015).

Podemos perceber que o NoSQL é um conjunto de tecnologias e arquiteturas, da qual os bancos de dados fazem parte, que foram desenvolvidas com o objetivo de solucionar os problemas que os sistemas tradicionais têm com relação à processamento e armazenamento de grandes quantidades de dados. Além disso, a necessidade de escalabilidade horizontal e suporte nativo à replicação, cujo suporte em sistemas tradicionais também é problemático, levaram à criação do NoSQL.

Os bancos de dados NoSQL possuem características fundamentais que os diferenciam dos bancos de dados relacionais, como o PostgreSQL, por exemplo.

Tais características permitem o armazenamento de dados não estruturados ou semiestruturados e também os tornam apropriados para o armazenamento de grandes massas de dados. Segundo Lóscio et. al. apud Prado et al. (2014), tais características são:

5.1 ESCALABILIDADE HORIZONTAL

A escalabilidade horizontal é possível graças ao fato de não existir controle de bloqueios (PRADO et al., 2014). Para Vieira et. al. (2012), a escalabilidade horizontal é definida como a capacidade de distribuição de dados e carga por vários servidores sem que haja o compartilhamento de disco ou memória. Tal abordagem permite o uso de máquinas comuns e, portanto, mais baratas.

Tal escalabilidade é possibilitada com o uso de uma técnica chamada Tabela Hash Distribuída (Distributed Hash Table (DHT)), onde as entidades

possuem uma representação $\langle key, value \rangle$. A entidade, que é representada por *value*, possui uma identificação única, sendo esse o papel do elemento *key* (CLARKE apud VIEIRA et al. (2012)).

5.2 AUSÊNCIA DE ESQUEMA OU ESQUEMA FLEXÍVEL

No entendimento de Leavitt apud Prado et al. (2014), a capacidade de bancos de dados NoSQL armazenarem dados estruturados ou semiestruturados se dá pelo fato de não exigirem um esquema, como acontece nos bancos de dados relacionais.

Essa ausência ou quase inexistência de esquema é benéfica para a escalabilidade e disponibilidade. Entretanto, por não possuir a estrutura rígida dos bancos de dados relacionais, não existem garantias quanto à integridade dos dados (PRADO et al., 2014).

5.3 PERMITE REPLICAÇÃO DE FORMA NATIVA

A capacidade de replicação é um recurso nativo dos bancos de dados NoSQL. Tal recurso é responsável por uma diminuição no tempo que se gasta no momento de recuperar dados (PRADO et al., 2014).

5.4 CONSISTÊNCIA EVENTUAL

Para Ye e Li apud Prado et. al. (2014), tal característica, que é baseada nos princípios do teorema CAP, diz que a consistência dos dados, nos diversos nós não pode ser sempre garantida.

Garante que todas as réplicas, ocasionalmente, após a finalização do tempo necessário para o sincronismo e o restabelecimento da conexão, confluem-se a um estado comum (VIEIRA et al. 2012).

5.5 ACID E BASE

O ACID e BASE são dois padrões utilizados por SGBDs, sendo o ACID o

padrão em sistemas relacionais e o BASE, usado em sistemas NoSQL. O padrão ACID traz dificuldades quando se precisa realizar a distribuição de banco de dados em vários nós, inviabilizando esse tipo de operação (DIANA; GEROSA, 2010). Abaixo é apresentado o padrão ACID e o BASE:

5.5.1 ACID

- Atomicidade (*atomicity*): Diz respeito a uma transação ser executada por completo ou nada ser executado (LÓSCIO; OLIVEIRA; PONTES, 2011).
- Consistência (*consistency*): Refere-se ao fato de, após uma transação ter sido efetuada, o banco de dados permanecer em um estado de consistência (LÓSCIO; OLIVEIRA; PONTES, 2011).
- Isolamento (*isolation*): Garante que haja isolamento nas transações, ou seja, uma transação não pode interferir em outra transação concorrente (LÓSCIO; OLIVEIRA; PONTES, 2011).
- Durabilidade (*durability*): Define que os dados decorrentes de uma transação, após a finalização desta, não serão perdidos (LÓSCIO; OLIVEIRA; PONTES, 2011).

5.5.2 BASE

- Basicamente Disponível (Basically Available): Define que, mesmo com subsistemas inoperantes, o sistema deve estar disponível na maior parte do tempo (SOUZA; SANTOS, 2015). Em outras palavras, isso quer dizer que uma aplicação funciona basicamente o tempo todo.
- Estado Leve (Soft State): Estabelece que o sistema não deve ser consistente o tempo todo (SOUZA; SANTOS, 2015).
- Eventualmente Consistente (Eventually Consistent): Garante que, não ocorrendo atualizações em um registro, estará este consistente e dessa forma, será retornada, em operações de busca, sua última atualização de valor (SOUZA; SANTOS, 2015). Dessa forma, define que o sistema torna-se consistente no momento devido.

5.6 QUAL A RELAÇÃO DE NOSQL COM BIGDATA?

No contexto de Big Data, onde há geração de grandes quantidades de dados, faz-se necessário o armazenamento e processamento de grandes quantidades de dados, nem sempre estruturados, em tempo hábil. Essa grande massa de dados, que possui diversos tipos de dados, como vídeos, informações de temperatura, áudio, dentre outros, representa um desafio para os bancos de dados relacionais, que não foram projetados para essa realidade e, por isso, apresentam um baixo desempenho ao manipular dados de tamanha magnitude. O NoSQL foi criado para suprir essa demanda não atendida pelos bancos de dados relacionais, fornecendo capacidade de manipulação de grandes quantidades de dados, sejam eles estruturados ou não.

5.7 TEOREMA CAP

A sigla CAP (*consistency, availability e partition tolerance*), que segundo Brewer apud Vieira et al. (2012), também é conhecido como Brewer's theorem, significa consistência, disponibilidade e tolerância à partição. Vieira et al. (2012, p. 3) diz que “somente duas dessas três propriedades podem ser garantidas simultaneamente em um ambiente de processamento distribuído de grande porte”. Esse teorema, que fala sobre propriedades de sistemas de dados distribuídos, diz que:

5.7.1 Consistência

Diferentemente do que diz a consistência de transações dos bancos de dados relacionais, refere-se à ordenação das requisições executadas, significando que uma leitura de um item após sua escrita deve retornar seu novo valor (DIANA; GEROSA, 2010).

Para Vieira et al. (2012), a consistência tem o objetivo de permitir, em transações distribuídas, agir com uma semântica sem parciaisidades, assim como a manutenção de um estado de consistência nas réplicas, caso existam.

5.7.2 Disponibilidade

Pode ser entendida como a capacidade que um sistema possui de dar respostas para todas as requisições que chegam em um nó em funcionamento (DIANA; GEROSA, 2010).

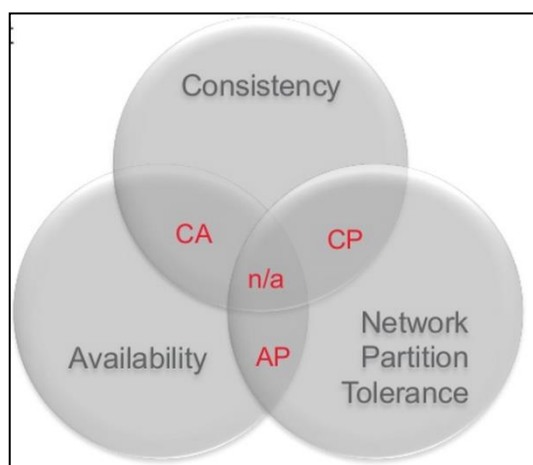
Visa manter o sistema disponível na maior parte do tempo e especifica que, mesmo quando houver algum tipo de falha, deverá haver pleno funcionamento, mesmo havendo recursos replicados indisponíveis (VIEIRA et al. 2012).

5.7.3 Tolerância à partição

Refere-se à propriedade de um sistema continuar em funcionamento mesmo quando, devido à problemas na rede, ocorre uma divisão (ou particionamento) em duas ou mais partições, impedindo assim a comunicação entre nós de partições diferentes (DIANA; GEROSA, 2010).

Através da divisão do processamento dos nós em grupos isolados, não havendo, dessa forma, comunicação entre si. Dessa forma possui o objetivo de, mesmo com uma falha da rede, manter em pleno funcionamento o sistema (VIEIRA et al. 2012). Abaixo é mostrada uma imagem que ilustra o teorema CAP:

Figura 1 – Teorema CAP



Fonte: CAP Theorem, 2017.

5.8 TIPOS DE BANCOS NOSQL

Os tipos de bancos de dados NoSQL mais comuns são os bancos de dados orientados a documentos, bancos de dados de grafos, bancos de dados de famílias de colunas e bancos de dados de chave-valor (DIANA; GEROSA, 2010). Abaixo segue uma descrição de cada tipo de banco NoSQL citado:

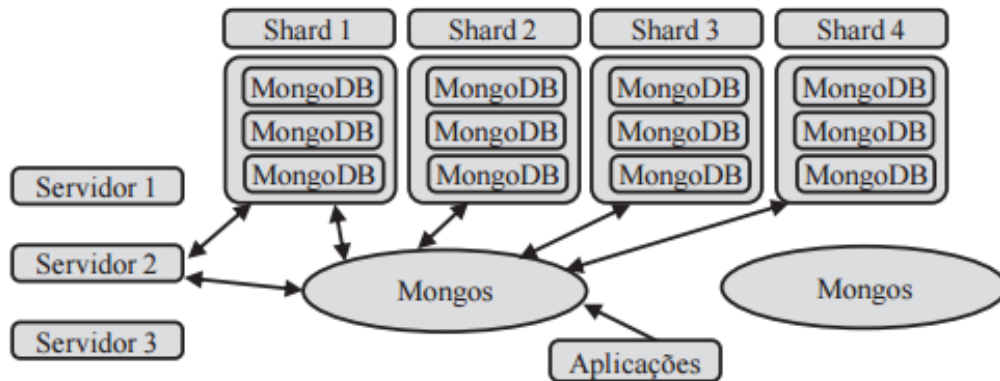
5.8.1 Bancos de dados orientados a documentos

Seus documentos funcionam como coleções de atributos e valores, podendo, em alguns casos, possuir atributos multi-valorados. O fato de não possuírem um esquema permite que os documentos armazenados não precisem ter a mesma estrutura. Essa característica torna esse tipo de banco uma excelente opção para o armazenamento de dados semiestruturados (DIANA; GEROSA, 2010).

De acordo com Pokorny apud Prado et al. (2014), bancos de dados orientados a documentos são coleções de documentos, onde cada um desses documentos possui um campo (que funciona como uma chave) e seu respectivo valor. Segundo Harrison apud Prado et al. (2014), nesse tipo de banco de dados a programação é mais fácil, pois, o documento pode fazer o mapeamento da estrutura das classes, embora isso comprometa a integridade dos dados, já que há duplicação de itens.

Diana e Gerosa (2010), fala que esse tipo de banco de dados usa, como forma de armazenamento, uma estrutura que pode embutir outra estrutura, podendo, dessa forma, um documento conter em si outro documento. Tal arranjo é feito comumente usando o formato JSON. Nesse contexto de bancos de dados, podem ser citados, como exemplo o MongoDB e o CouchDB.

Abaixo é mostrada uma imagem que ilustra uma estrutura de sharding que, segundo Prado et al. (2014), há uma divisão dos dados em diversos shards, cada um deles contando sua estrutura própria de replicação:

Figura 2 – Estrutura de shards.

Fonte: Prado (et al., 2014 p. 153).

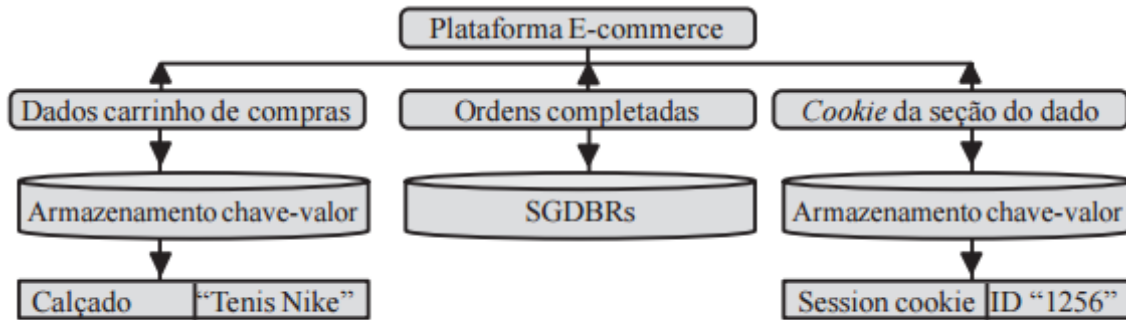
5.8.2 Bancos de dados chave-valor

Os bancos de dados dessa categoria, que também são conhecidos como tabelas de *hash* distribuídos, utilizam chaves como mecanismo de indexação para seus objetos. Essas chaves também são usadas como índices para realizar as buscas desses objetos (DIANA; GEROSA, 2010).

Esse tipo de banco de dados permite que os dados sejam visualizados na forma de uma grande tabela *hash* na qual há associação das chaves de pesquisa e valores (PRADO et al., 2014). Nessa estrutura, considerada simples e eficiente, cada par chave-valor é único e isso torna muito simples o acesso aos dados (TIWARI apud PRADO et al. 2014). Esse modelo, que é muito simples de implementar, incrementa a capacidade de acesso aos dados em sistemas de alta escalabilidade, pois permite um acesso rápido aos dados através de suas chaves (PRADO et al., 2014).

O Redis e o MemcacheDB são exemplos de bancos de dados chave-valor. Abaixo é apresentada uma imagem que mostra uma estrutura, voltada para o e-commerce, onde é possível notar o uso combinado de um banco de dados relacional e o NoSQL chave-valor:

Figura 3 – Banco de dados chave-valor.



Fonte: Prado (et al., 2014, p. 151).

5.8.3 Bancos de dados orientados a colunas:

São bancos de dados que possuem um melhor desempenho quando se deseja otimizar a leitura de dados estruturados, pois, armazenam os dados contiguamente, por coluna. A gravação de um novo registro é bem mais morosa, se comparada a um banco de dados relacional. Isso faz esse tipo de banco de dados ser mais adequado para o processamento analítico online (OLAP) (STONEBRAKER apud DIANA; GEROSA, 2010).

Esse modelo de banco de dados possui maior complexidade que o chave-valor (POKORNY apud PRADO et al. 2014). A indexação dos dados ocorre por meio de uma tripla (linha, coluna e *timestamp*), onde chaves fazem a identificação de linhas e colunas e o *timestamp* é usado para fazer a diferenciação de versões de um mesmo dado (PRADO et al., 2014).

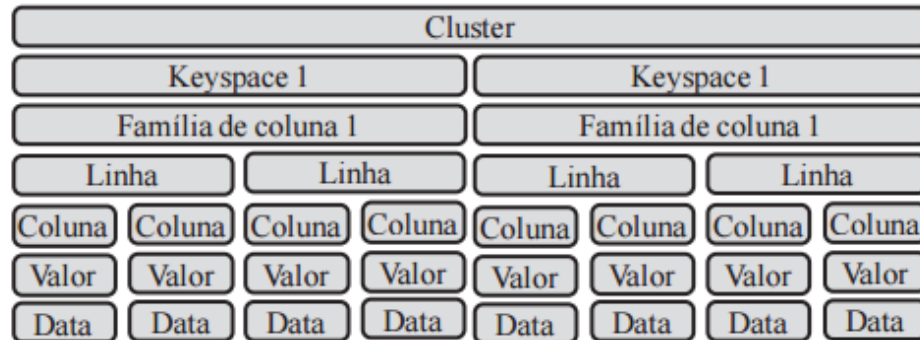
Tiwari apud Prado et al. (2014) diz que dentre todos os bancos de dados não relacionais, são os bancos de dados orientados à colunas que possuem uma maior popularidade. Segundo Prado et al. (2014), mesmo que esse tipo de banco já exista nas mais variadas formas há um bom tempo, sua popularização se deu entre 2003 e 2008, após a publicação de alguns artigos pelo Google.

No entendimento de Rahien apud Prado et al. (2014), embora exista uma semelhança entre bancos de dados relacionais e bancos de dados orientados à coluna, esses dois tipos são totalmente distintos. Dessa forma, o mesmo tipo de solução não pode ser aplicado aos dois tipos de bancos de dados citados.

Bancos de dados como o Cassandra e o Hbase são exemplos dessa

categoria. Abaixo é possível observar um modelo desse tipo de banco de dados:

Figura 4 –Banco de dados de colunas.



Fonte: Prado (et. al., 2014 p. 152).

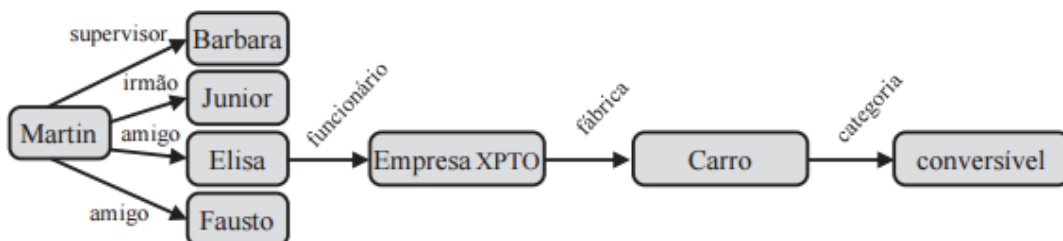
5.8.4 Bancos de dados orientados a grafos

Baseando-se no modelo de grafos, esse tipo de banco de dados representa os dados na forma de grafos dirigidos (ANGLES; GUTIERREZ apud DIANA; GEROSA, 2010).

Segundo Prado et. al. (2014), os bancos de dados orientados a grafos baseiam-se em nós, propriedades dos nós e relacionamentos. Os nós funcionam como os vértices do grafo enquanto os relacionamentos funcionam como as arestas. Tal modelo possibilita armazenar as entidades juntamente com seus relacionamentos. Dessa forma, é possível armazenar um dado e interpretá-lo de diversas maneiras, de acordo com seu relacionamento.

Neo4j e o InfoGrid são exemplos de bancos desse tipo. Abaixo é possível ver uma imagem que ilustra a estrutura de um bando de dados orientado a grafos:

Figura 5 – Banco de dados de grafos



Fonte: Prado (et. al., 2014, p. 153).

6 TESTE DE DESEMPENHO

6.1 COMPARATIVO ENTRE DUAS SOLUÇÕES NOSQL(MONGODB E COUCHDB)

Com o objetivo de observar o desempenho de dois SGBDs NoSQL orientados a documentos, foi feito um teste de carga(estresse), visando realizar um comparativo de desempenho, para o banco de dados MongoDB e CouchDB. Para realização do teste foi usada a ferramenta JMeter em conjunto com um plugin instalado, chamado *Stepping Thread Group*.

Além disso, foi desenvolvido, usando a linguagem PHP, um script de inserção de dados para cada um dos SGBDs testados. Tais scripts, responsáveis por inserir os dados (estrutura da nota fiscal) no CouchDB e MongoDB, podem ser vistos, respectivamente, no ANEXO A e ANEXO B. Dessa forma, a cada requisição, uma operação de inserção era realizada no banco de dados.

6.1.1 Máquina usada nos testes

- Processador: Core I3 M380 (1733 MHz)
- Memória RAM: 6GB DDR3
- Disco Rígido: 240 GB SSD

6.1.2 Softwares usados nos testes

- Ubuntu Linux 14.04 LTS
- PHP 5.5.9
- Apache JMeter 3.0
- Java versão 1.8.0_91
- MongoDB versão v3.0.12
- CouchDB versão 1.5.0

6.1.3 Configurações do JMeter

- Número de usuários (threads): 1000

6.1.4 Estrutura da nota fiscal

Para a execução dos testes foi criada, usando dados fictícios, uma estrutura que simula uma nota fiscal, possuindo cabeçalho, rodapé e uma seção de itens, que contém três produtos listados. Tal estrutura pode ser vista abaixo:

```

serie: 05336 - numero: 456

emissor
nomeEmissor: Brasil LTDA - cgcEmissor: 12312.35.6456, inscEstadualEmissor:
45456456556
enderecoEmissor: Rua Mar del Plata - estadoEmissor: RN telefoneEmissor:
98856-9632

comprador
nome: Joao Dos Santos Almeida - cic: 45456.4546.789798
telefone: 98854-9658 - endereco: Av. das Alagoas, 523 – Neopolis

numlinha: 1 - idproduto: 158
descproduto: Notebook Lenovo - qtd: 1 - precouni: 1264,50
precototal: 1264,50

numlinha: 2 - idproduto: 159
descproduto: Notebook LG - qtd: 2 - precouni: 2251,60
precototal: 4503,20

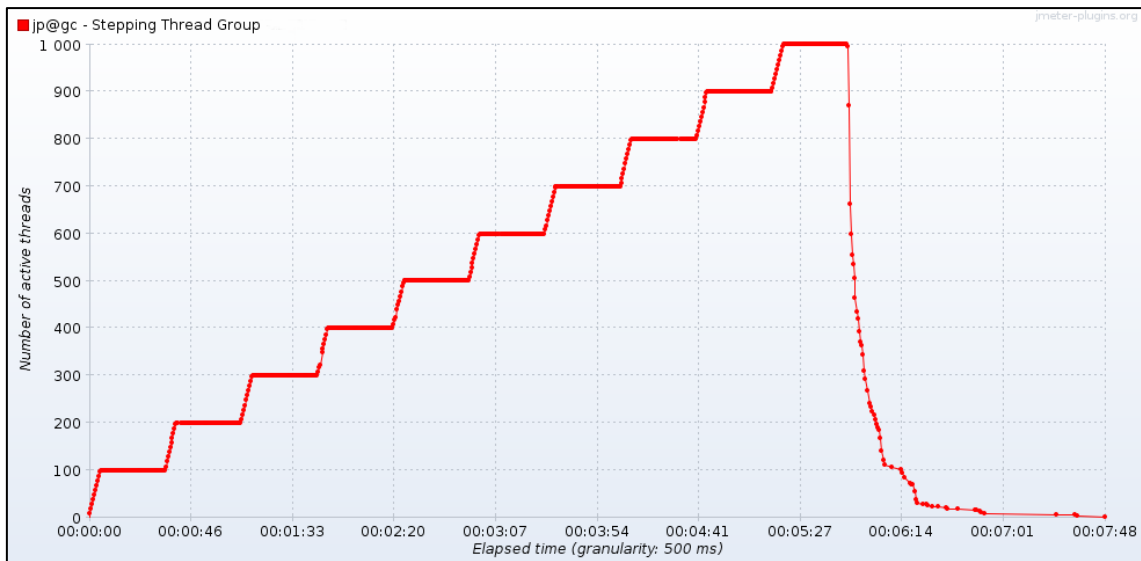
numlinha: 3 - idproduto: 160
descproduto: Impressora Laser -          qtd: 2 - precouni: 894,50
precototal: 1789,00

totalimpostos: 256,68
valortotalnota: 4650.20

```

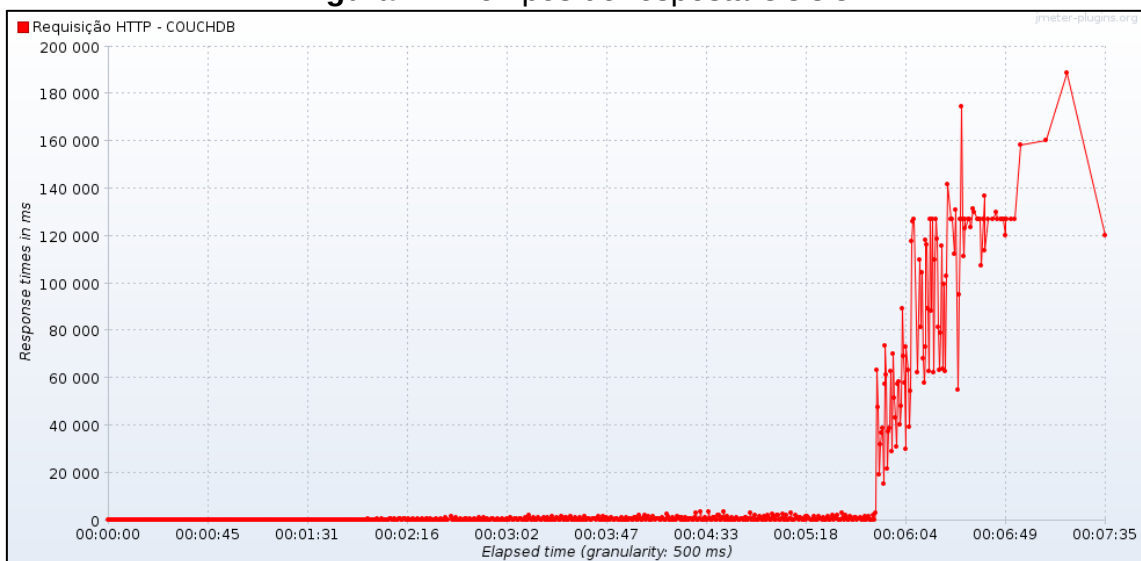
6.1.5 Gráficos dos testes

Após a realização dos testes, foram obtidos, através de gráficos, os seguintes resultados:

Figura 6 – Número de usuários ativos

Fonte: Autoria própria, 2016.

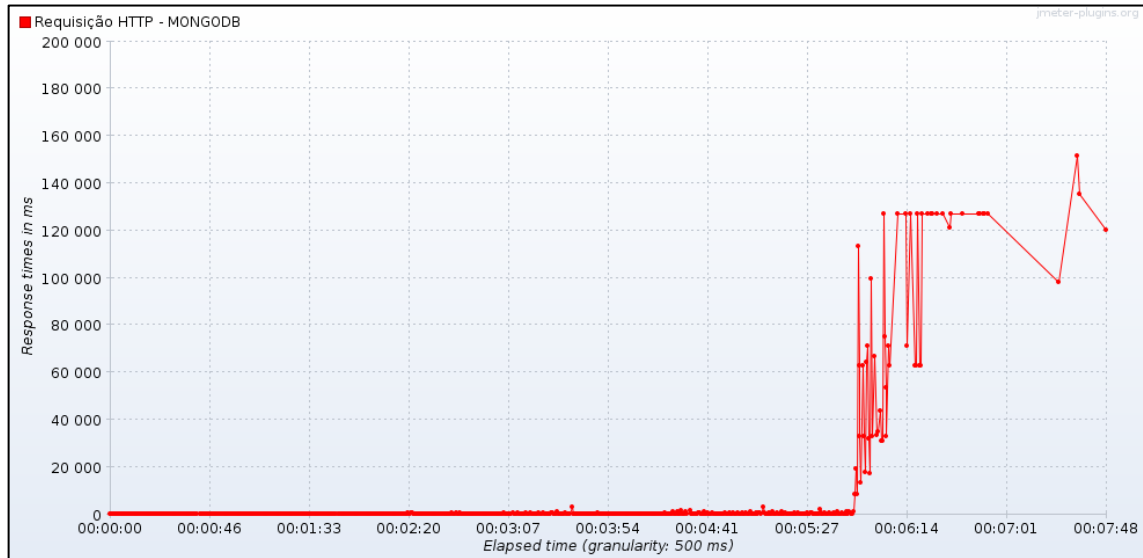
6.1.5.1 COUCHDB

Figura 7 – Tempos de resposta COUCHDB

Fonte: Autoria própria, 2016.

6.1.5.2 MONGODB

Figura 8 – Tempos de resposta MONGODB



Fonte: Autoria própria, 2016.

6.2 RESULTADO DOS TESTES

Ao final dos testes foi observado um volume final de dados no CouchDB de 255,6 MB contra 918,1 MB do MongoDB.

Conforme mencionado anteriormente, para os testes realizados com os dois bancos foram utilizados mil usuários (threads) simultâneos. Nos gráficos exibidos para cada banco de dados foram mostrados os tempos de resposta de cada SGBD diante das requisições dos usuários. De acordo com os gráficos obtidos ao final do teste, é possível perceber que, ao chegar ao máximo de usuários do teste, o CouchDB apresentou uma maior variação (oscilação) nos tempos de respostas do que o MongoDB. Tal oscilação não diz exatamente que o MongoDB é melhor que o CouchDB, mas, relata apenas os resultados obtidos, especificamente, nos testes desse estudo.

7 O QUE É BIG DATA?

Ainda não há uma definição precisa para Big Data, mas pode-se usar o termo para designar uma coleção de tendências tecnológicas que permite uma nova abordagem para o tratamento e entendimento de grandes conjuntos de dados para fins de tomada de decisões (BRETERNITZ; SILVA, 2013).

7.1 OS VS DO BIG DATA

O Big Data possui três dimensões principais: Volume, velocidade e variedade (VOLPATO; RUFINO; DIAS, 2014). Tais dimensões são descritas de forma mais detalhada abaixo:

7.1.1 Volume

O volume se caracteriza pela grande quantidade de dados que, gerados por sistemas convencionais, redes sociais, sistemas de RFID, sensores diversos e smartphones, etc. Essas fontes diversas de dados são as responsáveis diretas pelo crescimento exponencial dos dados em formato digital.

7.1.2 Velocidade

Um outro aspecto importante do Big Data, a velocidade, diz respeito à celeridade da captura e processamento, sendo quase em tempo real (MCAFEE; BRYNJOLFSSON apud BRETERNITZ; SILVA; LOPES, 2013).

7.1.3 Variedade

A variedade se refere ao fato dos dados do Big Data possuírem origens distintas, não sendo, em grande parte, estruturados, pois representam grandezas diferentes como, por exemplo, composição química do ar ou temperatura (LOHR apud BRETERNITZ; SILVA; LOPES, 2013).

7.2 QUE DECISÕES PODEM SER TOMADAS?

Volpato, Rufino e Dias (2014), relatam que o Big Data pode trazer uma grande contribuição, no que se refere ao processo de tomada de decisão, com informações de qualidade e em pouquíssimo tempo. Isso garante à empresa uma vantagem competitiva e um melhor posicionamento no mercado. Segundo McAfee e Brynjolfsson (apud BRETERNITZ; SILVA; LOPES, 2013) uma pesquisa mostrou que, nas empresas que usam Big Data, o índice de produtividade é 5% maior, enquanto a lucratividade aumenta 6%.

O Big Data também pode auxiliar na área de segurança pública por meio do armazenamento e processamento de grandes quantidades de dados, oriundos de comunicações eletrônicas, dessa forma, é possível a identificação de padrões que indiquem algum tipo de ameaça, como um grupo terrorista (SILVA, 2015).

7.3 PRIVACIDADE

No final do ano de 2014, o mundo ficou sabendo, por meio de denúncias dos jornais The Guardian e The Washington Post, sobre o programa de vigilância do governo americano, chamado de PRISM. Tal programa permite o acesso da agência nacional de segurança americana, ou NSA (National Security Agency), aos servidores das principais empresas de serviços de Internet e telecomunicações americanas. Através da análise das informações destas empresas é possível identificar padrões de comportamento que indiquem ligação com crimes ou terrorismo (BERNARDO; BORLONE, 2015).

Tal fato mostra a grande exposição a qual estamos sujeitos, mesmo usando serviços dos mais simples ligados a grande rede de computadores. Além disso, faz levantar o questionamento sobre o limite da invasão de privacidade, mesmo que justificada pela segurança do próprio cidadão.

8 QUAIS SOLUÇÕES EXPLORAM ESSES CONCEITOS?

Dentre as soluções de Big Data existentes, foi escolhida nessa seção, pelo grande número de recursos e também simplicidade, a ferramenta chamada *IBM InfoSphere BigInsights*.

Essa ferramenta, desenvolvida pela IBM, possui uma versão do Hadoop modificada, além de ferramentas como o Hive e o Pig. Tendo como principal diferencial a simplificação do uso do Hadoop, o *IBM InfoSphere BigInsights* vem sendo bastante usado no meio corporativo. Possui capacidade de fazer análises comportamentais e sentimentais, com base em funções e macros com algoritmos, além de fornecer um amplo suporte à gráficos.

Além do suporte à praticamente todos os gráficos considerados tradicionais, a ferramenta dispõe de gráficos Text Cloud, responsável pela geração de tags, que são etiquetas contendo links. Outro recurso dessa ferramenta é a possibilidade de geração de gráficos em formato de mapas, gerados com base em dados de áreas geográficas (BERNARDO; BORLONE, 2015).

9 CONCLUSÕES

A presente pesquisa permitiu perceber a importância das soluções NoSQL no cenário atual, caracterizado pelo crescimento exponencial dos dados, assim como a necessidade de um ágil processamento de tais informações, muitas vezes quase em tempo real.

No teste comparativo de desempenho entre o MongoDB e o CouchDB, optou-se por realizar o teste apenas uma vez para cada banco de dados. Para a obtenção de um resultado mais preciso, o ideal seria a realização de pelo menos dez testes para cada banco, realizando-se em seguida uma média com os resultados obtidos.

Este estudo trouxe um grande aprofundamento do conhecimento sobre as tecnologias envolvidas no contexto de NoSQL e Big Data, permitindo dessa forma, um amadurecimento profissional a respeito do tema. Além disso, foi possível entender como os bancos de dados NoSQL estão inseridos no contexto de Big Data. O estudo mostrou que existem diversos tipos de bancos de dados NoSQL, sendo cada um utilizado de acordo com o problema existente. A respeito do uso de bancos NoSQL, em uma pesquisa básica no Google sobre o assunto, foram encontrados cem empresas ou projetos que usam o MongoDB contra quarenta e nove que usam o CouchDB.

Através das informações obtidas por meio desse estudo é possível perceber, diante da tendência de aumento da quantidade de dados produzidos mundialmente, que há um futuro promissor para os bancos de dados NoSQL. Isso é reforçado pelo suporte nativo à escalabilidade e replicação, além de excelente desempenho na manipulação de grandes volumes de dados, em sua maioria não estruturados, correspondentes à realidade de hoje.

REFERÊNCIAS

ALMEIDA, André Luiz de et al. Data Warehouse para a Saúde Pública: Estudo de Caso SES-SP. Disponível em: <http://www.incor.usp.br/spdweb/prodcient_eng/filesTrabalhos2006/CBIS2006RS.pdf>. Acesso em: 2 set. 2016.

BERNARDO, Daniel dos Santos; BORLONE, Danilo Romualdo. **Análise comportamental por intermédio das tecnologias de Big Data**. 2015. Disponível em: <<http://www.revistarefas.com.br/index.php/RevFATECZS/article/view/45/72>>. Acesso em: 5 nov. 2016.

BILINSKI, Fernando Rodrigo. **Internet das Coisas**. 2014. Disponível em: <http://www.inf.ufpr.br/aldri/disc/artigos/2014/fernando_f1.pdf> Acesso em: 5 set. 2016.

BOAGLIO, Fernando. **MongoDB: construa novas aplicações com novas tecnologias**. São Paulo: Casa do Código, 2015.

BOGORNÝ et al. **Desenvolvimento de um SIG para saúde pública usando software livre**. 2002. Disponível em: <<http://www.inf.ufsc.br/~vania.bogorny/artigos/WSL2002.pdf>> Acesso em: 27 ago. 2016.

BRETERNITZ, Vivaldo José; SILVA, Leandro Augusto da; LOPES, Fabio Silva. **O uso de Big Data em computacional social science: tema que a sociedade precisa discutir**. 2013. Disponível em: <<http://fatecid.com.br/reverte/index.php/revista/article/view/78/79>> Acesso em: 30/09/2016.

BRETERNITZ, Vivaldo José; SILVA, Leandro Augusto. **Big Data: um novo conceito gerando oportunidades e desafios**. 2013. Disponível em: <<http://201.55.32.167/RETC/index.php/RETC/article/view/74/pdf>>. Acesso em: 06 set. 2016.

CAP Theorem. Disponível em: <<https://github.com/codedellemc/training/blob/master/accreditation/intro-to-nosql.md>>. Acesso em: 8 fev. 2017.

CARDOSO, Olinda Nogueira Paes; MACHADO, Rosa Teresa Moreira. **Gestão do conhecimento usando data mining**: estudo de caso na Universidade Federal de Lavras. 2008. Disponível em:
 <http://s3.amazonaws.com/academia.edu.documents/36686696/6643-12435-1-PB.pdf?AWSAccessKeyId=AKIAJ56TQJRTWSMTNPEA&Expires=1473467229&Signature=fZfo7MZdx8mzylftTdRYzcolH1o%3D&response-content-disposition=inline%3B%20filename%3DGestao_do_conhecimento_usando_data_minin.pdf> Acesso em: 4 set. 2016.

CARVALHO, Rodrigo Baroni de. **Aplicações de softwares de gestão do conhecimento**: tipologia e usos. 2000. Disponível em:
 <http://www.bibliotecadigital.ufmg.br/dspace/bitstream/handle/1843/LHLS-69XQRP/mestrado___rodrigo_baroni_de_carvalho.pdf?sequence=1> Acesso em: 4 set. 2016.

CERVANTES, Christian Alonso Vega. **Um sistema de detecção de ataques sinkhole sobre 6lowpan para internet das coisas**. 2014. Disponível em:
 <<http://acervodigital.ufpr.br/bitstream/handle/1884/38133/R%20-%20D%20-%20CHRISTIAN%20ALONSO%20VEGA%20CERVANTES.pdf?sequence=1&isAlloved=y>> Acesso em: 5 set. 2016.

DIANA, Mauricio de; GEROSA, Marco Aurélio. **NOSQL na Web 2.0**: um estudo comparativo de bancos não-relacionais para armazenamento de dados na Web 2.0. 2010. Disponível em:
 <http://www.lbd.dcc.ufmg.br/colecoes/wtdbd/2010/sbbd_wtd_12.pdf> Acesso em: 2 nov. 2016.

FARIAS, Marcelo Bukowski de. **Injeção de SQL em aplicações web**: causas e prevenção. 209. Disponível em:
 <<http://www.lume.ufrgs.br/bitstream/handle/10183/18573/000730977.pdf?sequence=1>>. Acesso em: 09 jun 2016.

FERRARI, Edilaine R. et al. **Um comparativo de consumo elétrico entre arquiteturas x86 e ARM em servidores de bancos de dados**. Disponível em:
 <https://www.researchgate.net/profile/Vinicius_Maran/publication/259339458_Um_Comparativo_de_Consumo_Eletrico_entre_Arquiteturas_x86_e_ARM_em_Servidores_de_Bancos_de_Dados/links/00b4952b190ba16296000000.pdf>. Acesso em: 6 set. 2016.

FROZZA, Iago Ferreira; DALLASEN, Ricardo Vanni. **LunchBOX**: Sistema online para reservas de refeições. 2016. Disponível em:
 <<http://painel.passofundo.ifsul.edu.br/uploads/arq/20160711182938423414178.pdf>> Acesso em: 19 ago. 2016.

HOKAMA, Daniele Del Bianco et al. **A modelagem de dados no ambiente Data Warehouse**. 2004. Disponível em: <<http://meusite.mackenzie.com.br/rogerio/tgi/2004ModelagemDW.pdf>>. Acesso em: 2 set. 2016.

JAMIL, George Leal. **Business Intelligence**: informações para a inteligência dos negócios. 2000. Disponível em: <<http://www.fumec.br/revistas/facesp/article/download/5/5>> Acesso em: 4 set. 2016.

LEMOS, Eliane Prezepiorski; STEINER, Maria Teresinha Arns; NIEVOLA, Julio César. **Análise de crédito bancário por meio de redes neurais e árvores de decisão**: uma aplicação simples de data mining. 2005. Disponível em: <<http://rausp.usp.br/wp-content/uploads/files/V4003225.pdf>> Acesso em: 4 set. 2016.

LÓSCIO, Bernadette Farias; OLIVEIRA, Hélio Rodrigues de; PONTES, Jonas César de Sousa. NoSQL no desenvolvimento de aplicações Web colaborativas. **VIII Simpósio Brasileiro de Sistemas Colaborativos**, Brasil, 2011. Disponível em: <http://www.addlabs.uff.br/sbsc_site/SBSC2011_NoSQL.pdf> Acesso em: 2 nov. 2016.

MACÊDO, José Antônio F. de et. al. **Gerenciamento de dados em nuvem**: conceitos, sistemas e desafios. Disponível em: <http://s3.amazonaws.com/academia.edu.documents/4660346/gerenciamento_dados_nuvm.pdf?AWSAccessKeyId=AKIAJ56TQJRTWSMTNPEA&Expires=1480273454&Signature=SYZvGd%2Fu759ywRONLWmHICBw%2Bw%3D&response-content-disposition=inline%3B%20filename%3DGerenciamento_de_Dados_em_Nuvem_Conceito.pdf>. Acesso em: 27 nov. 2016.

MATTEUSSI, Kassiano, MINETTO, Elton. **Desenvolvimento de uma interface WEB com PHP para gerenciamento de banco de dados CouchDB**. Disponível em: <http://eltonminetto.net/docs/artigo_kassiano.pdf> Acesso em: 5 set. 2016.
MORAIS, Cristina Alves de Sousa. **Protótipo de sistemas de informação aplicado a administração de materiais utilizando Data Warehouse e conceitos de Data Mart**. 2000. Disponível em: <<http://dsc.inf.furb.br/arquivos/tccs/monografias/2000-1cristinaalvesdesouzamoraivf.pdf>> Acesso em: 4 set. 2016.

NIEDERAUER, Juliano. **Web Interativa com Ajax e PHP**. São Paulo: Novatec, 2007.

PRADO, Edmir P. V et. al. **Critérios para Seleção de SGBD NoSQL**: o ponto de vista de especialistas com base na literatura. 2014. Disponível em: <<http://www.lbd.dcc.ufmg.br/colecoes/sbsi/2014/0012.pdf>> Acesso em: 4 nov. 2016.

RIBEIRO, Claudio José Silva. **Big Data: os novos desafios para o profissional da informação**. 2014. Disponível em:
<<http://www.ies.ufpb.br/ojs2/index.php/itec/article/view/19380/11156>>. Acesso em 09 de junho de 2016.

RUSCHEL, Henrique; ZANOTTO, Mariana Susan; MOTA, Wélton Costa da. **Computação em Nuvem**. 2010. Disponível em:
<<http://www.ppgia.pucpr.br/~jamhour/RSS/TCCRSS08B/Welton%20Costa%20da%20Mota%20-%20Artigo.pdf>> Acesso em: 25/08/2016.

SCHROEDER, Ricardo; SANTOS, Fernando dos. **Arquitetura e testes de serviços web de alto desempenho com NODE.JS e MONGODB**. Disponível em:
<http://www.ceavi.udesc.br/arquivos/id_submenu/787/ricardo_schroeder_versao_final_.pdf> Acesso em: 2 set. 2016.

SILVA, Gustavo Henrique M. A. da. **O big data como suporte às atividades forenses da perícia criminal**. 2015. Disponível em:
<<http://siaiap34.univali.br/sbseg2015/anais/WFC/artigoWFC04.pdf>>. Acesso em: 5 nov. 2016.

SILVA, Maurício Samy. **Ajax com jQuery: requisições ajax com a simplicidade de jQuery**. São Paulo: Novatec, 2009.

SILVA, Maurício Samy. **JavaScript: guia do programador**. São Paulo: Novatec, 2010.

SISLAGHI, Renato et al. **SACAD: construção de um Data Mart para apoio aos coordenadores e acadêmicos de cursos de graduação**. 2006. Disponível em:
<<https://repositorio.ufsc.br/bitstream/handle/123456789/66632/t0037.pdf?sequence=1&isAllowed=y>>. Acesso em: 2 set. 2016.

SOUSA, Flávio R. C.; MOREIRA, Leonardo O.; MACHADO, Javam C. **Computação em nuvem: conceitos, tecnologias, aplicações e desafios**. 2009. Disponível em:
<https://www.researchgate.net/profile/Javam_Machado/publication/237644729_Computacao_em_Nuvem_Conceitos_Tecnologias_Aplicacoes_e_Desafios/links/56044f4308aea25f3121f3.pdf> Acesso em: 26/08/2016.

SOUZA, Vanessa Cristina Oliveira de; SANTOS, Marcus Vinícius Carli dos. Maturing, Consolidation and Performance of NoSQL Databases-Comparative Study. In: _____. Proceedings of the annual conference on Brazilian Symposium on Information Systems: information systems: a computer socio-technical perspective, v. 1. **Brazilian Computer Society**, 2015. Disponível em: <http://plutao.sid.inpe.br/col/sid.inpe.br/plutao/2015/06.01.12.27/doc/souza_amadurecimento.pdf> Acesso em: 2 nov. 2016.

TAYLOR, ALLEN G. **SQL for Dummies**. 7. ed. Indianapolis: Wiley Publishing, 2010.

TIWARI, SHASANK. **Professional NoSQL**. [S.l.]: Wrox, 2011.

TONINI, Gustavo A., FRANK, Siqueira. **PGGRID: uma implementação de fragmentação de dados para o PostgreSQL**. 2010. Disponível em: <<http://www.inf.ufsc.br/~frank.siqueira/papers/WSL2010-Tonini.pdf>> Acesso em: 26 ago. 2016.

VERDI, Fábio Luciano et al. **Novas arquiteturas de Data Center para Cloud Computing**. 2010. Disponível em: <<http://www.ppgccs.net/verdi/mcsbrc2010.pdf>> Acesso em: 7 set. 2016.

VIEIRA, Marcos Rodrigues et al. **Bancos de dados NOSQL: conceitos, ferramentas, linguagens e estudos de casos no contexto de Big Data**. 2012. Disponível em: <http://data.ime.usp.br/sbbd2012/artigos/pdfs/sbbd_min_01.pdf> Acesso em: 3 nov. 2016.

VOLPATO, Tiago; RUFINO, Ricardo; DIAS, Jaime. Big Data: transformando dados em decisões. 2014. Disponível em: <http://web.unipar.br/~seinpar/2014/artigos/graduacao/Tiago_Volpato.pdf>. Acesso em: 9 jun. 2016.

WALTER, Roberto; DUARTE, Denio. **Implementação de Operadores OLAP Utilizando o Modelo de Programação Map Reduce no MongoDB**. 2016. Disponível em: <<http://www.lbd.dcc.ufmg.br/colecoes/erbd/2016/007.pdf>>. Acesso em: 3 nov. 2016.

ANEXO A – CÓDIGO PHP PARA INSERÇÃO DE DADOS NO COUCHDB

```

<?php
$couch_dsn = "http://localhost:5984/";
$couch_db = "notasfiscais";
require_once "couch.php";
require_once "couchClient.php";
require_once "couchDocument.php";

$client = new couchClient($couch_dsn,$couch_db);
$nota = new stdClass();

$cabecalho = new stdClass();
$rodape = new stdClass();
$empresaE = new stdClass();
$compradorPF = new stdClass();

//// cabecalho ////
$cabecalho->serie = "05336";
$cabecalho->numero = "456";

//// cabecalho - empresa emissora//
$empresaE->cgcEmissor = "12312.35.6456"; $empresaE->inscstadualEmissor =
"45456456556";
$empresaE->nomeEmissor = "Brasil LTDA"; $empresaE->telefoneEmissor = "98856-9632";
$empresaE->enderecoEmissor = "Rua Mar del Plata"; $empresaE->estadoEmissor = "RN";

$cabecalho->empresaEmissora = $empresaE;

// cabecalho – compradorPF //
$compradorPF->cic = "45456.4546.789798"; $compradorPF->nome = "Joao Dos Santos
Almeida";
$compradorPF->telefone = "98854-9658"; $compradorPF->endereco = "Av. das Alagoas, 523 -
Neopolis";
$cabecalho->comprador = $compradorPF;
// Item 1//
$item1 = new stdClass(); $item1->numlinha = "1"; $item1->idproduto = "158";
$item1->descproduto = "Notebook Lenovo"; $item1->qtd = "1"; $item1->precouni = "1264,50";
$item1->precototal = "1264,50";
// Item 2
$item2 = new stdClass(); $item2->numlinha = "2"; $item2->idproduto = "159";
$item2->descproduto = "Notebook LG"; $item2->qtd = "2"; $item2->precouni = "2251,60";
$item2->precototal = "4503,20";
// item 3
$item3 = new stdClass(); $item3->numlinha = "3"; $item3->idproduto = "160";
$item3->descproduto = "Impressora Laser"; $item3->qtd = "2"; $item3->precouni = "894,50";
$item3->precototal = "1789,00";
//
$rodape->totalimpostos = "256,68";
$rodape->valortotalnota = "4650,20";
//
// Criando a nota fiscal
$nota->cabecalho = $cabecalho;

// Itens da nota
$nota->itens[] = $item1; $nota->itens[] = $item2; $nota->itens[] = $item3;

// Rodape da nota

```

```
$nota->rodape = $rodape;

try {
    $response = $client->storeDoc($nota);
} catch (Exception $e) {
    echo "Error: ".$e->getMessage()." (errcode=".$e->getCode().")\n";
    exit(1);
}
print_r($response);
?>
```

ANEXO B – CÓDIGO PHP PARA INSERÇÃO DE DADOS NO MONGODB

```

<?php

$conexao = new MongoClient();
$db = $conexao->notafiscal;
//
$nota = array(
    "cabecalho" => array(
        "serie" => "05336",
        "numero" => "456",

        "empresaEmissora" => array(
            "cgcEmissor" => "12312.35.6456",
            "insc EstadualEmissor" => "45456456556",
            "nomeEmissor" => "Brasil LTDA",
            "telefoneEmissor" => "98856-9632",
            "enderecoEmissor" => "Rua Mar del Plata",
            "estadoEmissor" => "RN"
        ),
        "comprador" => array(
            "cic" => "45456.4546.789798",
            "nome" => "Joao Dos Santos Almeida",
            "telefone" => "98854-9658",
            "endereco" => "Av. das Alagoas, 523 - Neopolis"
        )
    ),
    "itens" => array(
        "item1" => array(
            "numlinha" => "1",
            "idproduto" => "158",
            "descproduto" => "Notebook Lenovo",
            "qtd" => "1",
            "precouni" => "1264,50",
            "precototal" => "1264,50"
        ),
        "item2" => array(
            "numlinha" => "2",
            "idproduto" => "159",
            "descproduto" => "Notebook LG",
            "qtd" => "2",
            "precouni" => "2251,60",
            "precototal" => "4503,20"
        ),
        "item3" => array(
            "numlinha" => "3",
            "idproduto" => "160",
            "descproduto" => "Impressora Laser",
            "qtd" => "2",
            "precouni" => "894,50",
            "precototal" => "1789,00"
        )
    ),
    "rodape" => array(
        "totalimpostos" => "256,68",
        "valortotalnota" => "4650,20"
    )
);

```

```
$colecacao = $db->notasfiscais;  
$res = $colecacao->insert($nota);  
print_r($res);
```

```
?>
```