

UNIVERSIDADE DE PERNAMBUCO
Faculdade de Ciência e Tecnologia de Caruaru
Sistemas de Informação

**ANÁLISE COMPARATIVA ENTRE BANCO DE DADOS RELACIONAIS E NÃO
RELACIONAIS COM ÊNFASE NO DESEMPENHO DE BASE DE DADOS NO
CONTEXTO DE INTERNET DAS COISAS**

Trabalho de Graduação
Sistemas de Informação

Ana Katarina Cavalcanti Medeiros
Orientador: Prof. Msc. Rômulo César Dias de Andrade



ANA KATARINA CAVALCANTI MEDEIROS

**ANÁLISE COMPARATIVA ENTRE BANCO DE DADOS RELACIONAIS E NÃO
RELACIONAIS COM ÊNFASE NO DESEMPENHO DE BASE DE DADOS NO
CONTEXTO DE INTERNET DAS COISAS**

Monografia apresentada como requisito parcial para obtenção do diploma de Bacharel em Sistemas de Informação pela Faculdade de Ciência e Tecnologia de Caruaru Universidade de Pernambuco.

Caruaru, dezembro, 2015

Dedico este trabalho a minha família em especial a minha mãe por toda dedicação, apoio, incentivo e carinho em todos os momentos.

AGRADECIMENTOS

Agradeço à minha mãe pela educação que ela me deu e que possibilitou essa minha conquista, sinto-me mais que agradecida, essa vitória também é sua, mãe.

A minha família, por todo amor, incentivo e apoio, em especial à meu tio João Paulo que sempre me mostra que as coisas boas sempre vem, e que me enche de orgulho e admiração, e a minha tia Verônica que está sempre me aconselhando e me dizendo que vai dar certo, só basta acreditar.

A Paloma, por ser a pessoa besta que me faz rir até quando eu não quero e com a qual eu aprendi a ter um pouquinho mais de paciência, por me dá incentivo de sempre continuar, me mostrando que sempre há um jeito, posso dizer que sou grata em ter você por perto e por estar sempre comigo em todos os momentos, sendo eles profissionais ou não.

A Janael por todo incentivo, por toda bronca, paciência, dedicação, e também pelos seus conselhos e ensinamentos mais valorosos, sem esquecer de toda ajuda na elaboração deste trabalho.

A Paulinha (LittlePaula) pelo incentivo desde o começo da faculdade, com uma tal de linguagem Pascal e por sua amizade carinhosa que sei que sempre posso contar.

Aos amigos que fiz durante esse tempo na universidade, Allan Souza, Eric Moraes, Jhonatha Donato, José Augusto, Lucas Neves, Marcelo Aparecido, por serem o melhor grupo de todos. Só foi possível aguentar todo esse tempo porque vocês estavam lá. Podem ter certeza, que levarei para sempre a amizade de vocês.

Aos professores da Universidade de Pernambuco, por contribuírem para a minha educação e desenvolvimento profissional, especialmente ao meu orientador Rômulo César, que me estimulou e me ajudou na conclusão deste trabalho.

E a todos que contribuíram de alguma forma com a elaboração deste trabalho.

RESUMO

No contexto de Internet das Coisas a manipulação das informações é uma importante direção de pesquisa na área de gerenciamento de dados. Dados massivos e heterogêneos trazem grandes desafios de armazenamento. Tecnologias existentes baseadas em banco de dados relacionais são inadequadas para lidar com tais volumes de dados devido à velocidade de processamento limitada e os custos de expansão significativos. As tecnologias que são normalmente baseadas em sistemas distribuídos de arquivos, gerenciamento de banco de dados distribuído, e tecnologias de processamento paralelo, como é o caso da tecnologia NoSQL, surgiram da necessidade de aprimorar a performance de aplicações e sistemas ao lidar com grande quantidade de dados. Com base na análise das características de dados da Internet das coisas, este trabalho propõe uma análise comparativa dos banco de dados relacionais e não relacionais como solução de armazenamento e disponibilização de dados nesse contexto.

Palavras-chaves: Banco de Dados; IoT; Benchmarking; Desempenho.

ABSTRACT

On the Internet of Things context, the manipulation of information is a very important direction on the data management research field. Massive and miscellaneous data brings up big storage challenges. The existing technologies based on relational database are inadequate to deal with such amount of data, due to limited processing speed and high expansion costs. Technologies that are usually based on distributed file systems, databases distributed management, and parallel processing technologies, such as NoSQL, have emerged the need to improve the performance of applications and systems to handle large amounts of data.. Based on the features of Internet of Things data analysis, this paper proposes a comparative analysis of relational and non-relational databases as a solution to storage and availability of data in this context.

Keywords: Database; IoT; Benchmarking.

ÍNDICE DE FIGURAS

Figura 1 - Modelo de Banco de Dados Relacional.....	18
Figura 2 - Exemplo de tabela no Modelo Relacional	19
Figura 3 - Exemplo de JSON	19
Figura 4 - Modelo de Banco de Dados de Documentos	22
Figura 5- Representação de JSON de um Banco de Dados de Documentos.....	23
Figura 6 - Representação Simplificada de um SGBD.....	25
Figura 7 - Comando para Instalação do Servidor do MongoDB.....	32
Figura 8 - Diagrama de Classe representando a estrutura de dados utilizada nos testes.....	33
Figura 9 - POST de Leituras de Sensores.....	34
Figura 10 – Sensor de Temperatura LM35.....	34
Figura 11 - Sensor de Luminosidade LDR 5mm.....	35
Figura 12 - Sensor de Umidade DHT11.....	36
Figura 13 - Query de Consulta Simples realizada no MySQL.....	36
Figura 14 - Query de Consulta Avançada realizada no MySQL.....	37
Figura 15 - Query de Consulta Simples realizada no MongoDB.....	37
Figura 16 - Query de Consulta avançada realizada no MongoDB.....	37
Figura 17 - Query de Exclusão realizada no MySQL.....	38
Figura 18 - Query de Exclusão realizada no MongoDB.....	38
Figura 19 - Comparação da Inserção: MySQL e MongoDB.....	40
Figura 20 - Comparação da Consulta Simples (a) e Consulta Avançada (b)	41
Figura 21 - Comparação da Exclusão.....	41

ÍNDICE DE QUADROS

Quadro 1. Comparação Benchmarking em Relação à Inserção.	39
Quadro 2. Comparação Benchmarking em Relação à Consulta Simples.....	39
Quadro 3. Comparação Benchmarking em Relação à Consulta Avançada.....	40
Quadro 4. Comparação Benchmarking em Relação à Exclusão	40

LISTA DE SIGLAS

IoT	<i>Internet of Things</i>
TCU	Tribunal de Contas da União
CNC	Confederação Nacional do Comércio de Bens, Serviços e Turismo
FNQ	Fundação Nacional de Qualidade
CPqD	Centro de Pesquisa e Desenvolvimento em Telecomunicações
SGBD	Sistema de Gerenciamento de Dados
IDC	<i>International Data Corporation</i>
BD	Banco de Dados
NoSQL	<i>Not Only SQL</i>
REST	<i>Representational State Transfer</i>
API	<i>Application Programming Interface</i>
CERP	<i>Cluster of European Research Projects</i>

SUMÁRIO

1	INTRODUÇÃO	12
1.1	Caracterização do Problema	13
1.2	Objetivos.....	14
1.3	Justificativa.....	14
1.4	Metodologia da Pesquisa	15
2	REFERENCIAL TEÓRICO	16
2.1	Banco de Dados	16
2.1.1	Banco de Dados SQL	16
2.1.2	Banco de Dados de Documentos	17
2.2	Escalabilidade	19
2.3	Modelo de Dados.....	20
2.3.1	Modelo Relacional.....	21
2.3.2	Modelo Não Relacional	23
2.4	Sistema de Gerenciamento de Banco de Dados	24
2.5	Internet das Coisas.....	25
2.5	Considerações Finais	27
3	FERRAMENTAS UTILIZADAS	28
3.1	MySQL.....	28
3.2	MongoDB	29
3.4	Benchmarking.....	30
3.4.1	Benchmarking Competitivo.....	30
3.4.2	Indicador Utilizado com o uso do Benchmarking	30
3.5	Considerações Finais	31
4	AMBIENTE E CENÁRIO DE TESTES	32
4.1	Configuração do Ambiente.....	32
4.2	Rede de Sensores	33
4.3	Teste de Desempenho.....	34
4.3.1	Inserção de Dados.....	34
4.3.2	Consulta de Dados	36
4.3.3	Exclusão de Dados.....	38

4.4 Resultado do Benchmarking.....	38
4.5 Considerações Finais	42
5 CONCLUSÕES	43
5.1 Trabalhos Futuros	43
REFERÊNCIAS BIBLIOGRÁFICAS	44

1 INTRODUÇÃO

No decorrer da história, houveram mudanças em plataformas, linguagens, arquiteturas e processos, onde o modelo de dados relacional sempre se manteve como principal escolha quando da definição de qual método utilizar para o armazenamento e recuperação de informações. De acordo com Date (2003) o modelo relacional é constantemente descrito tendo três aspectos: aspecto estrutural, aspecto de integridade e aspecto manipulador. No primeiro os dados são percebidos pelo usuário como tabelas, e nada além disso. No segundo essas tabelas satisfazem restrições de integridade. E no aspecto manipulador os operadores disponíveis para que o usuário possa manipular tabelas são operadores que derivam tabelas a partir de outras tabelas.

Diante do crescimento de aplicações que manipulam um volume considerável de dados e que, segundo Vieira et al. (2012), essa vasta quantidade de dados gerados traz novos e grandes desafios na forma de manipulação, armazenamento e processamento em várias áreas da computação, em especial nas áreas de bases de dados, mineração de dados e recuperação de informação.

Para Cunha (2011), essas novas aplicações fizeram com que a indústria de software apresentasse novas propostas às limitações dos sistemas de armazenamento relacional. Para Sousa et al. (2010), uma opção para solucionar os desafios inerentes a este problema é o movimento denominado NoSQL, que consiste em um Sistema de Gerenciamento de dados não relacional projetado para gerenciar grandes volumes de dados e que disponibilizam estruturas e interfaces de acesso simples.

Um caso de utilização intensiva de dados é o que hoje definimos como Internet das coisas, que de acordo com o *website* oficial da Cisco System (líder nessa transição), reúne pessoas (humanos), processos (gere a forma como as pessoas, dados e as coisas funcionam em conjunto), os dados (informações), as coisas (objetos inanimados) e dispositivos para fazer conexões de rede mais relevantes e valiosas do que nunca – transformando informações em ações que criam novas capacidades, experiências mais ricas e oportunidade econômica sem precedentes para as empresas, indivíduos e países.

Com esse número exacerbado de informações faz-se necessária a análise, mineração e manipulação dos dados de uma forma consistente, confiável e dando apoio na obtenção rápida e eficiente de informações relevantes. Diante disso, o fato de se ter um Sistema de

Gerenciamento de Banco de Dados se torna imprescindível, e quanto mais escalável, ágil e eficiente for o sistema, melhor será a sua produção de dados relevantes.

1.1 Caracterização do Problema

Segundo Nogueira et al. (2011), a Internet das coisas ou IoT (*Internet of Things*) é um dos pilares da internet do futuro. Sendo muitas vezes citada como nova revolução industrial ou ainda como *Web 3.0*. Para Atzori et al. (2010) IoT é a interconexão entre objetos físicos e computadores com a internet e tem como principal conceito a presença pervasiva da computação em nosso cotidiano, transformando tudo que nos cerca em objetos inteligentes (smart objects) virtuais.

Uma projeção da Cisco System (2015) indica que, até 2019, mais da metade do planeta (cerca de 3,9 bilhões de pessoas) terão acesso a Internet e o número de dispositivos capazes de acessar a web será três vezes maior do que a população global. O número faz parte de um levantamento divulgado no dia 27 de maio de 2015. Onde a Cisco trabalha com projeções da Organização das Nações Unidas e essas projeções indicam que, dentro de quatro anos, a população mundial será de cerca de 7,6 bilhões de habitantes. Pelos cálculos da Cisco, a internet terá cerca de 4 bilhões de usuários até lá. Isso representa um aumento de 1,4 bilhão na quantidade de conexões fixas e móveis em relação aos números atuais.

Segundo o International Data Corporation, o número de dispositivos ou coisas que podem ser conectados à IoT está se aproximando de 200 bilhões, com 7% (ou 14 bilhões) já conectados à Internet e se comunicando por meio dela. Hoje, os dados desses dispositivos conectados representam 2% dos dados do mundo inteiro. A IDC agora prevê que, até 2020, o número de dispositivos conectados chegará a 32 bilhões, representando 10% dos dados mundiais.

Até 2020, mais de 35% de todos os dados poderão ser considerados úteis, graças ao crescimento dos dados da IoT, mas dependerá das empresas colocá-los em uso.

Por outro lado, esse fenômeno apresenta desafios significativos, já que as empresas precisam gerenciar, armazenar e proteger todo o volume e toda a diversidade de dados. Por exemplo, a IDC estima que 40% dos dados no Universo Digital exigem algum nível de proteção, desde rigorosas medidas de privacidade até dados totalmente criptografados. Dito isso, só a metade desses dados (apenas 20%) está protegida atualmente.

Nesse contexto, este trabalho caracteriza-se pela problemática de que há a necessidade de um melhor entendimento sobre a escolha adequada de um banco de dados para que haja a

capacidade de avaliar qual tipo utilizar de acordo com esse ambiente de Internet das Coisas, exemplificando através de indicadores de comparação, para que consequentemente obtenha sucesso na implementação.

1.2 Objetivos

Este trabalho tem como objetivo geral comparar banco de dados relacionais e não relacionais no contexto de Internet das coisas.

Para se alcançar o objetivo geral, os seguintes objetivos específicos foram delineados:

- Estudar sobre o conceito de IoT;
- Estudar as características dos Bancos de Dados Relacionais;
- Estudar as características dos Bancos de Dados Não Relacionais;
- Implementar um Banco de Dados Relacional no contexto de IoT;
- Implementar um Banco de Dados Não Relacional no contexto de IoT;
- Comparar o desempenho entre os Bancos de Dados;
- Analisar os resultados comparativos.

1.3 Justificativa

Armazenar informações é, sem dúvida, o fundamento principal e primordial de uma base de dados, ou, comumente chamada de banco de dados. Segundo Date (2003) entende-se por banco de dados “uma coleção de dados persistentes, usada pelos sistemas de aplicação de uma determinada empresa”. Portanto um banco de dados deve além de armazenar coleções de dados importantes, ser capaz de manter a integridade e segurança dos mesmos e também oferecer facilidade no gerenciamento dos seus registros.

No ambiente de IoT não é diferente, pois segundo ATZORI, IERA e MORABITO (2010), a ideia consiste em poder conectar uma variedade de objetos (sensores, tags RFID, smartphones, computadores, e até objetos de uso mais comum) entre si. Estes objetos geram um fluxo de dados e a conexão permite que eles transmitam estes dados para outros objetos no meio, formando assim uma internet de coisas. Portanto há uma necessidade de se ter um banco de dados de forma a garantir confiabilidade, consistência e disponibilidade desses

dados, a partir disso conclui-se que é imprescindível a escolha de maneira adequada de uma ferramenta para atender a este requisito.

Esta pesquisa abordará uma comparação entre os modelos de dados relacional e não relacional aplicado ao ambiente de IoT, visando obter um melhor embasamento sobre qual modelo apresenta melhor desempenho nesse caso e, tendo em mente que o conceito de banco de dados não relacional é relativamente novo, e, portanto, pouco explorado teoricamente, o presente trabalho também contribuirá na produção de conhecimento sobre o tema, além de explorar o ambiente de IoT que de fato é muito promissor.

1.4 Metodologia da Pesquisa

Com o intuito de atingir os objetivos propostos nesse trabalho, a metodologia da pesquisa foi dividida em duas fases.

Na primeira fase foi realizada a revisão da literatura com objetivo de entender os conceitos que serão trabalhados nessa pesquisa.

Já na segunda fase foi utilizada a metodologia de benchmarking desempenho, que tem como propósito efetuar levantamentos de dados e análises, gerando informações para que seja possível conhecer outras formas de lidar com situações e problemas semelhantes. As informações geradas nos processos de benchmarking favorecem a transferência de conhecimento, além de garantir uma escolha adequada e pautada nos dados e análises obtidos no decorrer do processo.

Segundo a CNC (2015), serão consideradas algumas fases na metodologia benchmarking que são elas: planejar, coletar, analisar, adaptar e melhorar, enfatizando o principal objetivo que é a melhoria.

Considerando que o principal objetivo de uma análise comparativa é que as informações geradas pelo processo de comparação através do benchmarking são de suma importância e a forma de abordagem do benchmarking, pode ser quantitativa, representada por um resultado (indicador de desempenho), ou qualitativa, como um processo ou uma prática, o presente trabalho dessa maneira se classifica em quantitativo, por apresentar e indicar de forma objetiva um resultado de desempenho e qualitativa pois demonstrará através de um cenário qual escolha deverá ser adotada quando da sua conclusão.

2 REFERENCIAL TEÓRICO

Neste capítulo será apresentado o referencial teórico do presente trabalho, exemplificando contextos importantes para o entendimento do tema proposto.

2.1 Banco de Dados

Segundo Geremia (2010), um banco de dados é uma coleção de dados ou registros relacionados. Esses registros ou dados podem ser gravados e possuem um significado implícito representando aspectos do mundo real. Ou seja, é projetado, construído e povoado por dados a partir de uma proposta específica, segundo a necessidade do usuário.

Um banco de dados é criado e mantido por um SGBD, que é uma coleção de programas que permite aos usuários criar e manter um banco de dados. Para TANENBAUM (2000), “Um sistema de banco de dados é uma forma de armazenar dados e gerir informações para a posterior recuperação ou atualização dessas informações por um usuário. Deve evitar perdas de dados por falhas no sistema, acessos não Autorizados e anomalias de dados.” O SGBD é, portanto, um sistema de software de propósito geral que facilita os processos de definição, construção, manipulação e compartilhamento de banco de dados entre vários usuários e aplicações (ELMASRI E NAVATHE, 2011).

Date (2003) explica que um banco de dados representa algum aspecto do mundo real, algumas vezes chamado de “mini-mundo” e mudanças no mini-mundo provocam transformações no banco de dados.

Para GONÇALVES (2004) a definição de banco de dados se define como:

- Coleção de dados relacionados;
- Coleção logicamente coerente de dados com algum significado inerente;
- Um BD está sempre associado a aplicações e a usuários que têm interesse nele.

2.1.1 Banco de Dados SQL

São banco de dados que utilizam a linguagem SQL, que de acordo com Elmasri et al. (2011):

O nome SQL é derivado de Structure Query Language (Linguagem Estruturada de Consulta), e foi chamada inicialmente SEQUEL (Structured English QUERY Language — Linguagem de Pesquisa em Inglês Estruturado), sendo projetada e implementada na IBM Research como uma interface para um sistema experimental de um banco de dados relacional chamado SISTEMA R.

Elmasri et al. (2011) também explica que, a linguagem SQL pode ser considerada uma das maiores razões para o sucesso dos bancos de dados relacionais no mundo comercial. Como se tornou padrão para os bancos relacionais, os usuários têm pouca preocupação ao migrar suas aplicações de banco de dados, originadas por outros tipos de sistemas — por exemplo, em rede/e hierárquico —, para o sistema relacional. A razão disso é que, mesmo que se sintam insatisfeitos com o banco relacional que decidiram usar, qualquer que seja, não se espera que a conversão para outro banco de dados relacional gere custo e tempo excessivos, uma vez que ambos os sistemas devem seguir os mesmos padrões de linguagem. A vantagem de possuir esse padrão é que os usuários podem escrever declarações em um programa aplicativo e podem acessar dados armazenados em dois ou mais banco de dados relacionais, sem ter de alterar a sublinguagem de banco de dados (SQL) em ambos.

Para um melhor entendimento, a linguagem SQL é utilizada como linguagem-padrão para os bancos de dados relacionais. Segundo Elmasri et al. (2011), um esforço conjunto da ANSI (American National Standards Institute — Instituto Nacional Americano de Padrões) e da ISO (International Standards Organization — Organização Internacional de Padrões) chegou à versão-padrão da SQL (ANSI, 1986), chamada SQL-86 ou SQL1. Uma versão revisada e expandida chamada SQL2 (também conhecida como SQL-92) foi desenvolvida em seguida. A próxima versão do padrão foi originalmente chamada SQL3, mas atualmente é conhecida como SQL-99.

2.1.2 Banco de Dados de Documentos

Como já diz o nome, este modelo armazena coleções de documentos. LÓSCIO et al. (2013) declara que um documento, em geral, é um objeto com um identificador único e um conjunto de campos, que podem ser strings, listas ou documentos aninhados.

Segundo Sadalage et al. (2015), o banco de dados armazena e recupera documentos, os quais podem ser XML, JSON, BSON, entre outros. Esses documentos são estruturas de

dados na forma de árvores hierárquicas e autodescritivas, construídas de mapas, coleções e valores escalares.

Pelo fato de ser um banco NoSQL não possui esquemas, isso significa que um banco de dados de documentos não está associado a nenhuma coluna ou tipo de dado predefinidos como ocorre no modelo relacional. HOWS et al. (2015) declara essa a principal vantagem dessa implementação é a extrema flexibilidade para trabalhar com os dados, pois não há nenhuma estrutura predefinida necessária em seus documentos.

Temos uma representação de um modelo típico de um banco de dados de documentos, na Figura 1 apresentada abaixo:

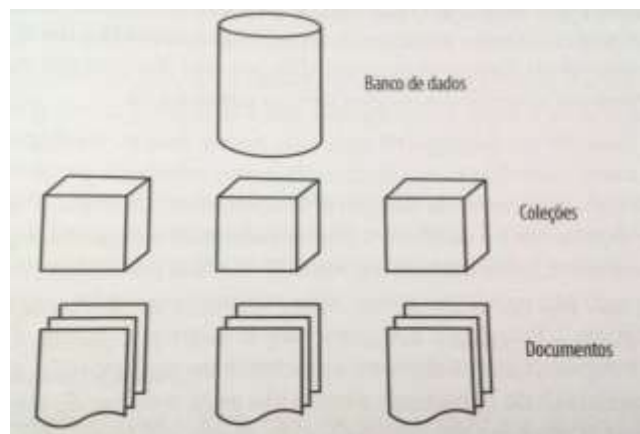


Figura 1 - Modelo de Banco de Dados de Documentos

Fonte: HOWS et al. (2015)

O modelo de estruturação de dados utilizado no banco de dados de documentos é o JSON, que segundo o *website* oficial o JSON (JavaScript Object Notation - Notação de Objetos JavaScript) é uma formatação leve de troca de dados. Para seres humanos, é fácil de ler e escrever. Para máquinas, é fácil de interpretar e gerar. Está baseado em um subconjunto da linguagem de programação JavaScript, Standard ECMA-262 3a Edição -Dezembro - 1999. JSON é em formato texto e completamente independente de linguagem, pois usa convenções que são familiares às linguagens C e familiares, incluindo C++, C#, Java, JavaScript, Perl, Python e muitas outras. Estas propriedades fazem com que JSON seja um formato ideal de troca de dados.

JSON é estruturado da seguinte maneira:

- Uma coleção de pares nome/valor. Em várias linguagens, isto é caracterizado como um *object*, record, struct, dicionário, hash table, keyed list, ou arrays associativas.

- Uma lista ordenada de valores. Na maioria das linguagens, isto é caracterizado como uma *array*, vetor, lista ou sequência.

Na Figura 2 está representada um exemplo de estrutura JSON normal:

```
{“nome”: “joao”, “idade” : “20”};
```

Figura 2 - Exemplo de JSON

Fonte: O Autor.

Temos uma representação do JSON de um banco de dados de documentos, na Figura 3 apresentada abaixo:

```
{
  "Type": "CD",
  "Artist": "Nirvana",
  "Title": "Nevermind",
  "Genre": "Grunge",
  "Releasedate": "1991.09.24",
  "Tracklist": [
    {
      "Track": "1",
      "Title": "Smells Like Teen Spirit",
      "Length": "3:02"
    },
    {
      "Track": "2",
      "Title": "In Bloom",
      "Length": "4:15"
    }
  ]
}
```

Figura 3 - Representação de JSON de um Banco de Dados de Documentos

Fonte: HOWS et al. (2015)

Alguns dos bancos mais populares que utilizam armazenamento de documentos são MongoDB, CouchDB, Terrastore, OrientDB, RavenDB e Lotus Notes.

2.2 Escalabilidade

A escalabilidade de um banco de dados é um ponto importante na etapa de projeção de qualquer banco de dados, pois o mesmo pode obter uma taxa de crescimento muito rápida se comparado a outros tipos de bancos.

“Uma série de fatores pode induzir o crescimento, incluindo o mercado, o setor e os fatores da economia. Se uma organização cresce mais rápido do que o previsto, experimenta todos os tipos de degradações de

desempenho, que vão desde falta de espaço em disco até uma desaceleração na velocidade de operação. Antever o crescimento esperado - e inesperado- é fundamental para a construção de sistemas escaláveis que possam suportá-lo.” (BALTZAN, 2012)

A escalabilidade é importante para a web, pois neste ambiente é comum haver a utilização de servidores adicionais para se manter a performance (conhecido como escalabilidade horizontal).

De acordo com Fowler (2003), “um sistema escalável é aquele que lhe permite adicionar hardware e obter uma melhora de desempenho proporcional.”

Um fato importante é que a arquitetura de um sistema pode influenciar em sua escalabilidade, impedindo por exemplo que novos recursos sejam adicionados, ou até mesmo que estes recursos não tenham impacto na performance do sistema.

Duas são as linhas mestras para atingir maior escalabilidade: o incremento com novos hardwares ou a substituição por novos hardwares de maior desempenho. O primeiro é denominado escalabilidade horizontal. Um exemplo típico de banco de dados que utilizam esse tipo de escalabilidade é o banco de dados não relacional, onde podemos incorporar novas máquinas para dar conta do aumento da demanda pelos usuários finais. O segundo é a escalabilidade vertical, onde uma nova máquina mais potente substitui a antiga.

2.3 Modelo de Dados

No contexto de base de dados uma das características fundamentais é que ela forneça algum nível de abstração de dados, para KOTARO (2005) um exemplo disso é a omissão de detalhes de armazenamento de dados que não são necessários para a maioria dos usuários. O modelo de dados é a principal ferramenta que fornece esta abstração.

Date (2003) detalha modelo de dados como:

Um conjunto de conceitos que podem ser usados para descrever a estrutura de uma base de dados. Por estrutura de uma base de dados entende-se os tipos de dados, relacionamentos e restrições pertinentes aos dados. Muitos modelos de dados também definem um conjunto de operações para especificar como recuperar e modificar a base de dados.

Existem vários modelos de dados, alguns já ultrapassados como o modelo de rede e o hierárquico, e outros que são utilizados atualmente como é o caso dos modelos relacional e não relacional, onde o primeiro modelo de acordo com Sadalage et al. (2013) se manteve constante há aproximadamente vinte anos, e o segundo que é um modelo relativamente novo referente ao anterior que surgiu da necessidade de manipulação de volumes maiores de dados.

2.3.1 Modelo Relacional

Banco de dados que utilizam o modelo relacional são bancos de dados que tomam como princípio a afirmação de que todas as informações estão armazenadas em tabelas (ou assim dizendo, relações).

Segundo Codd (1985), existem 12 regras juntamente com uma regra abrangente fundamental que ele chamou de Regra Zero para que um banco de dados seja considerado relacional:

0. Regra Fundamental: um Sistema de Gerenciamento de Banco de Dados relacional deve gerir os seus dados usando apenas suas capacidades relacionais. Este é o princípio fundamental em que as restantes 12 regras são baseadas.
1. Regra da informação: todas as informações devem ser representadas, ao nível lógico, apenas como dados em uma tabela.
2. Regra de acesso garantido: todo o dado pode ser acedido logicamente usando o nome da tabela, o nome da coluna e o valor da chave primária da linha.
3. Representação de valores nulos: o sistema deve ser capaz de representar valores nulos de um modo sistemático, independentemente do tipo do item de dados. Os valores nulos devem ser distintos de zero ou qualquer outro número, e de cadeias vazias.
4. Catálogo relacional: o catálogo do sistema que contém a descrição lógica do banco de dados deve ser representada da mesma forma que os dados comuns.
5. Sub linguagem abrangente: Independentemente do número de outras línguas que ele suporta, o sistema relacional deve incluir uma linguagem que permite declarações expressas como cadeias de caracteres para apoiar definição de dados, definição de pontos de vista, manipulação de dados, as regras de integridade, autorização do usuário, e um método de identificação de unidades para recuperação.

6. Regra da atualização de visões: qualquer visão que é teoricamente atualizável poderá ser também atualizada pelo sistema.
7. Inserção, atualização, e exclusão de alto nível: qualquer relação que pode ser tratada como um único operando para recuperação. Também pode executar de uma única vez operações de inserção, exclusão e alteração.
8. Independência física de dados: aplicações são imunes a alterações feitas em representações de armazenamento ou métodos de acesso.
9. Independência lógica de dados: as alterações feitas no nível lógico, tais como mesas de divisão ou tabelas combinadas, que não afetam o conteúdo da informação no nível lógico não exigem a modificação da aplicação.
10. Regra de integridade: As restrições de integridade tais como a integridade entidade e integridade referencial deve ser especificável na linguagem de manipulação de dados e armazenadas no catálogo (e não em programas).
11. Distribuição independente: a linguagem de manipulação de dados deve ser tal que, se o banco de dados é distribuído ou centralizado, os programas de aplicações e comandos dos usuários não precisam ser alterados.
12. Não subversão das regras: Se o sistema permite uma linguagem que suporta o acesso de um registro por vez, em qualquer programa usando este tipo de acesso não pode ignorar as restrições de integridade expressas em linguagem de alto nível.

Temos uma representação do modelo de um banco relacional, na Figura 1 apresentada abaixo:

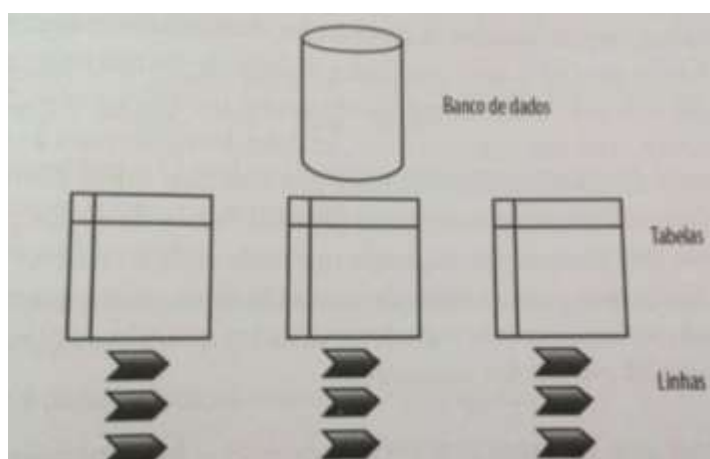


Figura 4 - Modelo de Banco de Dados Relacional

Fonte: HOWS et al. (2015)

E logo abaixo uma representação de tabelas de um banco de dados relacional, na Figura 2 apresentada abaixo:

Empregado

NumEmp	NomeEmp	Salário	Dept
032	J Silva	380	21
074	M Reis	400	25
089	C Melo	520	28
092	R Silva	480	25
112	R Pinto	390	21
121	V Simão	905	28
130	J Neves	640	28

Departamento

NumDept	NomeDept	Ramal
21	Pessoal	142
25	Financeiro	143
28	Técnico	144

Figura 5 - Exemplo de tabela no Modelo Relacional

Fonte: Neto (2008)

2.3.2 Modelo Não Relacional

A vasta quantidade de dados gerados por aplicações Web, juntamente com os requisitos diferenciados destas aplicações, como a escalabilidade sob demanda e o elevado grau de disponibilidade, têm contribuído para o surgimento de novos paradigmas e tecnologias (LÓSCIO et al. 2013).

De acordo com Sadalage et al. (2013), um termo usado como notação aos banco de dados não relacionais é o “NoSQL” em que é interpretado como “Not Only SQL” (não apenas SQL) e foi utilizado pela primeira vez em 1998, por Calor Strozzi. Ele surgiu da necessidade de aprimorar a performance de aplicações e sistemas ao lidar com grande quantidade de dados, que como explica LÓSCIO et al. (2011), essa nova categoria de banco de dados foi proposta com o objetivo de atender aos requisitos de gerenciamento de grandes volumes de dados, semi-estruturados ou não estruturados, que necessitam de alta disponibilidade e escalabilidade.

Para Sadalage et al. (2013):

É irônico que o termo “NoSQL” tenha feito sua primeira aparição no final da década de 1990 com o nome de um banco de dados relacional de código aberto (open source). O nome vem do fato de que o banco de dados não utiliza SQL como uma linguagem de consulta. Ele é manipulado por meio de shell scripts, que podem ser combinados em encadeamentos (pipelines) no Unix.

Diferente do modelo relacional que apresenta os dados em forma de tabelas e nada além disso, os bancos de dados não relacionais utilizam outras estruturas para a representação dos dados e também outros métodos para acesso dos mesmos.

Uma das características fortes dos bancos de dados NoSQL é que eles atuam sem um esquema, diferente do banco de dados relacional, e permite que sejam adicionados, livremente, campos aos registros do banco de dados sem ter que definir primeiro quaisquer mudanças de estrutura (Sadalage et al., 2013). Para uma melhor explicação, as informações são agrupadas em um único registro, sem a necessidade de relacionar várias tabelas para obter uma informação. Além disso, existem dentro do NoSQL subdivisões de acordo com a forma que será utilizada para o armazenamento de dados.

Sadalage et al. (2013) aborda, quatro categorias amplamente utilizadas no ecossistema do NoSQL: Chave-valor, Família de Colunas, Documento e Grafos. Dessas, as três primeiras compartilham uma característica comum em seus modelos de dados — que é chamado de “orientação agregada”, que nada mais é uma abordagem para se trabalhar com dados na forma de unidades que tenham uma estrutura mais complexa do que um conjunto de tuplas.

2.4 Sistema de Gerenciamento de Banco de Dados

De acordo com Date (2003), um Sistema Gerenciador de Banco de Dados é basicamente um sistema informatizado de manutenção de registros. Você pode considerar o próprio banco de dados como uma espécie de gabinete eletrônico para arquivamento; ou seja, ele é um tanque ou recipiente de uma coleção de arquivos de dados do computador. Onde os usuários do sistema podem executar uma variedade de operações sobre esses arquivos como por exemplo:

- Adicionar novos arquivos para o banco de dados;
- Inserir dados em arquivos existentes;

- Recuperar dados de arquivos existentes;
- Modificar os arquivos de dados existentes;
- Remover arquivos existentes;
- Remover arquivos existentes do banco de dados.

Na Figura 6 temos uma representação simples de um SGBD:

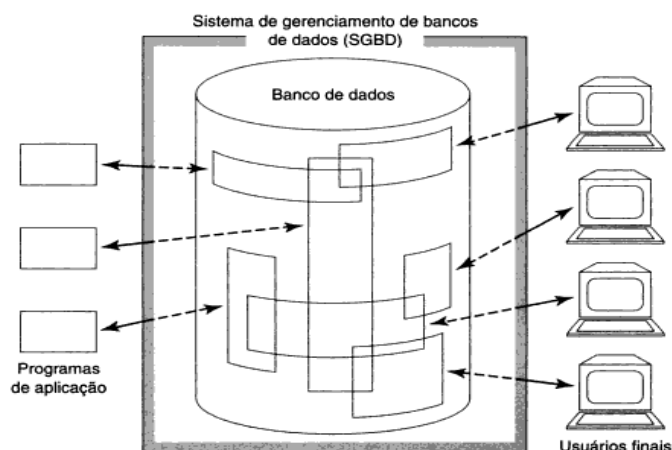


Figura 6 - Representação Simplificada de um SGBD

Fonte: Date (2003)

Pode-se afirmar então, que um Sistema Gerenciador de Banco de Dados é uma ferramenta para abstração de dados ou consulta e manipulação das informações contidas em um banco de dados, embora que nos dias atuais ainda seja confundido com o BD.

2.5 Internet das Coisas

Em consequência da popularização da internet, surgiram alguns conceitos e entre eles o de Internet das coisas (do inglês, Internet of Things), que para Melo (2014) é um conceito que já é discutido desde 1991. Mas foi apenas em 1999 que a definição “Internet das Coisas” foi cunhada, por proposição de Kevin Ashton – co-fundador do Massachusetts Institute of Technology (MIT).

De acordo com a Cluster of European Research Projects (2009), a Internet das Coisas é uma infraestrutura de rede global dinâmica, baseada em protocolos de comunicação onde “coisas” físicas e virtuais têm identidades, atributos físicos e personalidades virtuais, utilizando interfaces inteligentes e integradas às redes telemáticas.

O pensamento da CISCO (2011) é de que:

A Internet das coisas, algumas vezes referida como a Internet dos objetos, mudará tudo, inclusive nós mesmos. Isso pode parecer uma declaração arrojada, mas considere o impacto que a Internet já teve na educação, na comunicação, nos negócios, na ciência, no governo e na humanidade. Claramente, a Internet é uma das criações mais importantes e poderosas de toda a história humana.

Segundo HUI et al. (2008), a Aliança IPSO (do inglês, IP for Smart Objects) acredita em um futuro próximo que um grande número de dispositivos embarcados irá suportar o protocolo IP. Sendo assim, muitos objetos do cotidiano (como geladeiras, ar-condicionado, dentre outros) estarão em breve conectados diretamente a internet.

Na opinião de Martin Gren (2015) (co-fundador da Axis Communications):

A Internet das Coisas certamente cresceu no mundo do consumidor. Termostatos inteligentes enviam alertas aos proprietários das casas enquanto eles estão fora ou de férias. Babás eletrônicas permitem que os pais acompanhem seus filhos enquanto ainda estão no trabalho. Há até geladeiras inteligentes que encomendam automaticamente os mantimentos para a família quando eles acabam.

Alguns trabalhos focados nas “coisas” buscam apresentar propostas que visam garantir o melhor aproveitamento dos recursos dos dispositivos e sua comunicação. Por outro lado, ATZORI et al. (2010) define que os trabalhos que apresentam propostas focadas na semântica dos objetos da Internet das Coisas são importantes devido a grande quantidade de itens que estarão conectados a internet em um futuro próximo. Tais trabalhos apresentam propostas que estão focadas na representação, armazenamento, interconexão, pesquisa e organização da informação gerada na Internet das coisas, buscando soluções para a modelagem das descrições que permitam um tratamento adequado para os dados gerados pelos objetos.

Segundo o Centro de Pesquisa e Desenvolvimento em Telecomunicações há algumas premissas utilizadas em IoT, que são:

- Recursos não confinados.
- Liberdade na aplicação de dados.
- Cooperação entre todas as coisas.
- Invisibilidade da tecnologia.

Recursos não confinados e liberdade na aplicação de dados caminham juntos, porém um trabalhando com o lado mais físico e o outro com o mais lógico focando nos dados do equipamento, e como esses dados devem ser explorados não apenas com um objetivo, mas de forma genérica capazes de suportar aplicações de terceiros. Já cooperação entre todas as coisas implica como as coisas que segundo BANDYOPADHYAY et al. (2011), “coisas são dispositivos (físicos ou virtuais) que possuem identidades, atributos e personalidades virtuais e são capazes de utilizar interfaces inteligentes”, devem comunicar-se umas com as outras. E não menos importante, a invisibilidade da tecnologia que tem como conceito tornar as coisas inteligentes e cada vez mais invisível, ou seja, a tecnologia pode estar presente sem que as pessoas percebam.

2.5 Considerações Finais

Este capítulo teve como objetivo fazer uma revisão sobre os conceitos presentes em SGBD, modelo de dados, explanando dois exemplos, o modelo relacional e o não relacional, destacando os bancos de dados SQL e o orientado a documentos como exemplo de um banco relacional e não relacional, respectivamente, as características de IoT com ênfase na massa de dados gerada por ela.

No próximo capítulo serão descritos os sistemas utilizados nos experimentos, além da ferramenta de análise de desempenho. Além disso, serão introduzidos o conjunto de dados e o seu tratamento para a elaboração da modelagem e o plano de experimento do trabalho.

3 FERRAMENTAS UTILIZADAS

Com o intuito de demonstrar todos os passos adotados na implementação, estão descritas abaixo as características das ferramentas utilizadas.

3.1 MySQL

O MySQL é um servidor e gerenciador de banco de dados (SGBD) relacional, projetado inicialmente para trabalhar com aplicações de pequeno e médio portes, mas hoje atendendo a diversas aplicações de grande porte. Possui todas as características que um banco de dados de grande porte precisa, sendo reconhecido por algumas entidades como Facebook, Google, Twitter, entre outras.

Segundo *website* oficial da oracle, o MySQL é o banco de dados de código aberto mais popular do mundo e possibilita a entrega econômica de aplicativos de banco de dados confiáveis, de alto desempenho e redimensionáveis, com base na Web e incorporados. Esse banco teve origem quando os desenvolvedores David Axmark, Allan Larsson e Michael “Monty” Widenius, na década de 90, precisaram de uma interface SQL compatível com as rotinas ISAM que utilizavam em suas aplicações e tabelas. Em um primeiro momento, tentaram utilizar a API mSQL, contudo a API não era tão rápida quanto eles precisavam, pois utilizavam rotinas de baixo nível. Utilizando a API do mSQL, escreveram em C e C++ uma nova API que deu origem ao MySQL. Com o ótimo resultado gerado por essa nova API, o MySQL começou a ser difundido e seus criadores fundaram a empresa responsável por sua manutenção, denominada MySQL AB, o MySQL ainda possui: suporte à integridade referencial, transações e views, ferramentas gráficas de administração e extração de dados, atualizações constantes através de equipe própria, suporte a cluster, stored, procedure e subselects (MySQL, 2005). Outro ponto forte são seus diversos tipos de tabelas, que tornam o banco de dados bastante seguro e estável contando com integridade referencial, backup e restore, controle de usuários e acessos, e ainda, se necessário, verificação e correção de corrompimento de tabelas.

A partir dessa fase, o MySQL tornou-se mais conhecido por suas características de rápido acesso e cada vez mais utilizado. Novas versões foram lançadas, contemplando novas necessidades e firmando, assim, sua posição no mercado. Sua mais recente versão é a 5.7, a qual conta com novos recursos, estabelecendo sua capacidade para competir com os bancos de dados privados de maior popularidade (SQL Server e Oracle).

3.2 MongoDB

MongoDB é “um banco de dados orientado a documentos que provê alta performance, alta disponibilidade e fácil escalabilidade” (MongoDB Manual, 2015). Esse banco tem como uma das características não exigir uma estrutura fixa, como normalmente um modelo relacional utiliza. Para HOWS et al. (2015), o MongoDB (derivado da palavra humongous em inglês, que quer dizer gigantesco) é um tipo relativamente novo de banco de dados que não tem conceitos de tabelas, esquemas, SQL ou linhas. Não há transações, conformidade com ACID¹, joins (junções), chaves estrangeiras, nem vários outros recursos.

É composto por dois serviços: mongod e mongos. Em que o mongos de acordo com BELLIO (2014) é um cliente shell padrão usado para autosharding, onde o Sharding é a distribuição física de dados em nós distintos, sendo utilizado apenas quando se é necessário o balanceamento de carga. Já o mongod é o servidor do banco de dados, em que é altamente recomendado que ele seja executado em um sistema operacional 64 bits, uma vez que o tamanho máximo para um banco de dados em uma arquitetura de 32 bits é de 2 gigabytes.

Ainda segundo BELLIO (2014), o servidor de MongoDB pode armazenar um ou mais banco de dados distintos, onde cada banco de dados é constituído de uma ou mais coleções e cada coleção pode ter um ou mais documentos. Estes documentos são estruturados como documentos JSON e armazenados no formato BSON (Binary JSON). Para um melhor entendimento, HOWS et al. (2015) cita o JSON como uma excelente maneira de trocar dados, pois é desprovido de esquemas, e os documentos podem ser atualizados individualmente ou alterados de modo independente de qualquer outro documento, e o BSON é uma ótima maneira de armazená-los.

LÓSCIO et al. (2013) cita que o modelo de dados do MongoDB é bastante simples de compreender e descreve-se como:

- Um banco de dados armazena um conjunto de coleções;
- Uma coleção armazena um conjunto de documentos;
- Um documento é um conjunto de campos;
- Um campo é um par chave-valor;
- Uma chave é um nome (string);
- Um valor é um(a):
 - caracter, inteiro, ponto flutuante, timestamp ou binário;

¹ N.T.: *Atomicity, Consistency, Isolation and Durability*, ou Atomocidade, Consistência, Isolamento e Durabilidade.

- o um documento;
- o um "array" de valores;

Como solução NoSQL a escolha pelo banco de dados MongoDB se deve ao grande grau de maturidade que o mesmo já possui, bem como por uma maior adoção pela comunidade.

3.4 Benchmarking

Com o objetivo de se conhecer processos e métodos utilizados pelos bancos de dados analisados na pesquisa foi necessário o uso da ferramenta benchmarking que de acordo com BOGAN (1996) benchmarking é um padrão de excelência ou realização contra o qual outras coisas similares devem ser medidas ou julgadas, além de ser considerado um padrão utilizado em comparações de sistemas computacionais, que se baseia em medidas e uso de indicadores. A aplicação de testes seguindo tal padrão gera medidas quantitativas de desempenho capazes de serem comparadas com as de outros sistemas.

3.4.1 Benchmarking Competitivo

O benchmarking competitivo ou de desempenho como o termo é mais utilizado, pode ser classificado como um tipo de comparação realizado com base na utilização dos mais variados padrões e indicadores de desempenho, geralmente esses indicadores estão relacionados a aspectos como: produtividade, utilização de recursos, custos unitários e quando possível, da qualidade do serviço (TCU, 2000).

Esse tipo de benchmarking é especialmente útil para se identificar possíveis diferenças de desempenho, podendo estas serem significativas ou não, as quais podem sugerir possibilidades de executar melhor um tipo de atividade, melhorar a eficiência ou reduzir custos.

3.4.2 Indicador Utilizado com o uso do Benchmarking

Sendo o Benchmarking de Desempenho uma ferramenta que serve como auxílio quando deseja-se efetuar uma comparação, explicitando seus resultados de forma clara e

objetiva, foi estabelecido o indicador de **desempenho** para apoiar a escolha do banco de dados que melhor se adequa ao ambiente de IoT.

Alguns autores demonstram a importância e justificam esse indicador:

De acordo com SCHROEDER (2013), quando se fala em desempenho de sistemas, diversos fatores devem ser levados em consideração, pois desempenho não se refere somente a velocidade do sistema, e sim a outros fatores como disponibilidade e confiabilidade.

O teste de desempenho é fundamental para definir a capacidade do servidor e aplicação, bem como definir os limites do sistema de acordo com o hardware disponível. “O objetivo é duplo: entender como o sistema responde a carregamento (isto é, número de usuários, número de transações ou volume global de dados), e coletar métricas que vão levar a modificações de projeto para melhorar o desempenho”. (PRESSMAN, 2006).

3.5 Considerações Finais

Neste capítulo foram apresentados o MySQL e o MongoDB, que são os BD's utilizados no experimento. Também foi apresentado o Benchmarking Competitivo como ferramenta de análise de comparação nos testes de desempenho.

O capítulo seguinte mostra o ambiente de testes e os resultados que foram obtidos para os dois sistemas para o plano descrito acima.

4 AMBIENTE E CENÁRIO DE TESTES

Neste capítulo serão apresentados o ambiente e o cenário utilizado nos testes dos bancos de dados. Visando uma maior clareza as etapas da comparação a serem efetuadas, bem como uma maior fidelidade ao processo como um todo.

4.1 Configuração do Ambiente

O ambiente de análise dos bancos de dados presente nesse trabalho foi realizado a partir de um cenário de implementação do conceito de IoT, buscando exemplificar uma situação real onde uma rede de sensores diversos enviariam em tempo real informações dos mais variados tipos a um banco de dados que seria responsável pelo armazenamento e disponibilização de tais informações. Essas informações seriam enviadas ao banco via recurso de uma API REST, a qual receberia os dados e faria as operações objetos da análise como em uma aplicação real de IoT. Devido a complexidade do cenário algumas tecnologias foram adotadas visando proporcionar de uma forma rápida e confiável o melhor caminho para simulação de tal cenário conforme segue.

A instalação, configuração e execução dos testes de desempenho foram realizados em um computador do tipo PC, com processador AMD E-450, CPU de 1.6 GHZ com 2 GB de RAM e HD de 500 GB com sistema operacional GNU/Linux Debian 8 Jessie de 64bits e kernel 3.16.0.

A versão do MongoDB utilizada foi a 3.0 a qual foi instalada através do aptitude que é uma interface em modo texto para o gerenciamento de pacotes existente no Debian GNU/Linux, que permite instalar a maioria dos softwares bem como suas dependências, conforme Figura 7.

```
aptitude install mongod
```

Figura 7 - Comando para Instalação do Servidor do MongoDB

Fonte: O Autor.

Já no caso do mysql foi utilizado o pacote LAMP (Linux Apache, MySQL, PHP), que trata-se de um grupo de softwares open source que é tipicamente instalado em conjunto para permitir a um servidor hospedar websites dinâmicos e aplicações web. Este termo é atualmente um acrônimo que representa o sistema operacional Linux, com o servidor web Apache possuindo suporte a linguagem de scripts PHP e com os respectivos drivers

necessários para conexão com o também instalado servidor de banco de dados mysql, onde sua versão utilizada quando da instalação do pacote LAMP foi a 5.5.46. Lembrando ainda que no caso do mongo faz-se necessária a instalação manual do driver de conexão PHP, que foi realizada conforme o procedimento que consta no *website* oficial do MongoDB.

4.2 Rede de Sensores

O cenário foi composto por alguns sensores, esses sendo incluídos como forma de se obter uma base de dados consistente e com quantidade que se possa mensurar e analisar o desempenho dos bancos, já que cada um deles será lido e armazenado em tempo real, seguindo este pensamento foi utilizada uma rede de sensores que geraram uma massa de dados, abaixo estão descritos os sensores que foram utilizados.

- **Sensor de Temperatura:** A Figura 8 representa o sensor de temperatura LM35 que conforme *datasheet* (especificação) do fabricante, é um sensor de precisão que apresenta uma saída de tensão linear relativa à temperatura em que ele se encontra. Tornando possível fazer leituras de temperaturas do ambiente ou em geral.

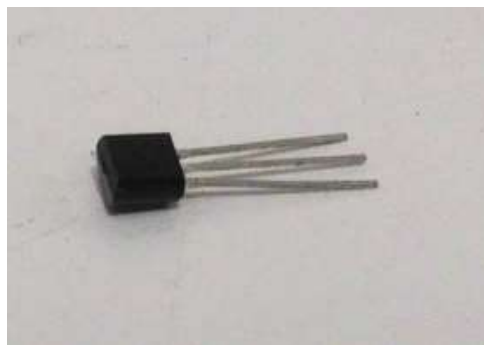


Figura 8 - Sensor de Temperatura LM35

Fonte: O Autor.

- **Sensor de Luminosidade:** A Figura 9 representa o sensor de luminosidade LDR 5mm que segundo *datasheet* (especificação) do fabricante, é um componente cuja resistência varia de acordo com a intensidade da luz. Quanto mais luz incidir sobre o componente, menor a resistência. A partir da utilização desse sensor, é possível medir a intensidade da luz no ambiente baseado em uma escala resistiva.



Figura 9 - Sensor de Luminosidade LDR 5mm

Fonte: O Autor.

- **Sensor de Umidade:** A Figura 10 representa o sensor de umidade DHT11 que de acordo com *datasheet* (especificação) do fabricante, este sensor é baseado em uma célula única capacitivo, com características de alta confiabilidade e estabilidade a longo prazo, estrutura de polímero sólido patenteado, adequado para tensão linear ou circuitos de saída de frequência, tempo de resposta rápido, entre outras.



Figura 10 - Sensor de Umidade DHT11

Fonte: O Autor.

4.3 Teste de Desempenho

De acordo com o cenário foram efetuados testes de desempenho nos quesitos: **inserção, consulta simples, consulta avançada e exclusão.**

4.3.1 Inserção de Dados

Para a inserção dos dados foi utilizada a estrutura que está representada através do diagrama de classe da Figura 11. Para testar o desempenho a inserção foi realizada em ciclos de um segundo, baseado nas quantidades 100, 1000 e 10000 registros/documentos,

respectivamente. Após isso serão coletados o tempo médio de inserção para cada quantidade em ambos os bancos de dados.

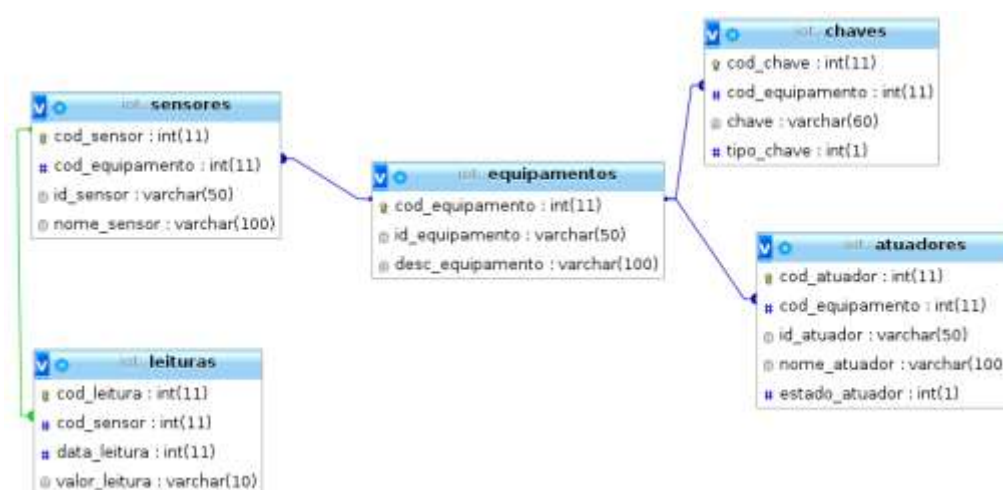


Figura 11 - Diagrama de Classe representando a estrutura de dados utilizada nos testes
Fonte: O Autor.

Para o envio dos dados dos sensores foi utilizada uma REST API escrita na linguagem PHP por conta da facilidade de uso e implementação.

Para um melhor entendimento REST é um estilo de arquitetura com foco em recursos, muito utilizado em aplicações web dada sua facilidade de uso. Esse termo foi descrito por Fielding (2000) em sua tese de doutorado, que define um conjunto de princípios que podem ser aplicados na construção de sistemas com uma arquitetura orientada a recurso. Um recurso pode ser qualquer componente de uma aplicação que seja endereçado na Web através de uma URI.

Sendo uma maneira fácil de enviar dados, e faz uso de recursos que podem ser representados na forma de JSON, e disponibilizados em uma Interface de Programação de Aplicativo (API).

A Figura 12 reproduz um exemplo de estrutura de envio de dados diretamente ao recurso /sensores/leituras da API, o qual utilizou o método POST do protocolo HTTP.

```

{
  "cod_sensor": "1",
  "data_leitura": "1449017717",
  "valor_leitura": "24.5"
},
{
  "cod_sensor": "2",
  "data_leitura": "1449017717",
  "valor_leitura": "55"
},
{
  "cod_sensor": "3",
  "data_leitura": "1449017717",
  "valor_leitura": "70"
}

```

Figura 12 - POST de Leituras de Sensores

Fonte: O Autor.

O conjunto de dados final contido em ambos os bancos foi de 386,700 (trezentos e oitenta e seis mil e setecentos), equivalentes a quantidade de documentos no MongoDB e registros na tabela leituras no MySQL.

4.3.2 Consulta de Dados

Para análise do desempenho das consultas de dados foram realizadas queries, respectivamente para MySQL e MongoDB, que retornam um intervalo de tempo de todos os registros/documentos referentes as leituras dos sensores que foram inseridos segundo o procedimento descrito no item 4.2.1. Este parâmetro de tempo serviu de fator de desempenho.

A consulta simples consiste na busca de todas as leituras com a data da leitura '1446113600' (que está representada em formato timestamp, que equivale a data 2015-11-10). Já na consulta avançada o resultado deve trazer todas as leituras entre as datas '1446336000' (2015-11-01) e '14447545600' (2015-11-15) em que o nome do sensor seja 'temperatura' e que possua 1 ou mais leituras.

A query equivalente ao banco MySQL a consulta simples está representada na Figura 13, onde só utiliza-se uma cláusula WHERE para localizar o data_leitura '1447113600' (2015-11-10) na tabela de leituras através do LIKE.

```

SELECT * FROM testeiot.leituras l WHERE l.data_leitura LIKE '1447113600'

```

Figura 13 - Query de Consulta Simples realizada no MySQL

Fonte: O Autor.

A query que apresenta a consulta avançada está representada na Figura 14, diferente da consulta simples nota-se além do tamanho, a utilização de várias junções entre tabelas e inúmeras funções AND, além de uma sub-consulta para retornar o número de leituras. Foi utilizado a interpolação de strings do PHP (%s) para passar os valores.

```
SELECT * FROM testeiot.leituras l, testeiot.sensores s,
testeiot.equipamentos e, testeiot.chaves c WHERE l.cod_sensor =
s.cod_sensor AND s.cod Equipamento = e.cod Equipamento AND
c.cod Equipamento = e.cod Equipamento AND l.data_leitura > %s AND
l.data_leitura < %s AND s.nome_sensor LIKE %s AND (SELECT count (*)
FROM testeiot.leituras l, testeiot.sensores s WHERE l.cod_sensor =
s.cod_sensor) >= 1 ORDER BY l.data_leitura, ("1446336000",
"1447545600", "1", "temperatura")
```

Figura 14 - Query de Consulta Avançada realizada no MySQL
Fonte: O Autor.

No caso do MongoDB a consulta simples foi efetuada através da função find() da coleção, passando a um dicionário com as respectivas chaves e valores. Esse dicionário de dados também deve ser um JSON válido. A Figura 15 demonstra o exemplo descrito abaixo.

```
db.leituras.find({"leituras.data_leitura":"1447113600"})
```

Figura 15 - Query de Consulta Simples realizada no MongoDB
Fonte: O Autor.

Para a consulta avançada foram utilizados vários operadores de comparação da linguagem de consulta do MongoDB, como por exemplo: \$gt, \$lt e \$size. Sendo o ultimo citado utilizado para o cálculo do numero de leituras. A querie equivalente a essa consulta está representada na Figura 16.

```
db.leituras.find({"leituras.data_leitura": {"$gt": 1446336000},
"leituras.data_leitura": {"$lt": 1447545600},
"leituras.sensor.nome_sensor":"temperatura", "leituras.sensor":
{"$size": 1 }}).sort("leituras.data_leitura")
```

Figura 16 - Query de Consulta avançada realizada no MongoDB
Fonte: O Autor.

4.3.3 Exclusão de Dados

Quando da análise do desempenho no que tange a exclusão dos dados, o teste seguiu a mesma lógica da metodologia adotada no item 4.2.2, onde todos os dados inseridos na tabela leituras e no caso do MongoDB coleção leituras serviram de parâmetros visando se obter o tempo para a conclusão da operação.

Para o MySQL a query equivalente a exclusão de um registro está representada na Figura 17, onde utiliza-se uma cláusula WHERE para localizar o sensor específico para poder efetuar a exclusão na tabela de sensores, utilizando também o LIKE.

```
DELETE FROM testeiot.sensores s WHERE s.id_sensor LIKE 1
```

Figura 17 - Query de Exclusão realizada no MySQL

Fonte: O Autor.

No caso do MongoDB a exclusão foi realizada utilizando-se a função remove() da coleção, passando mais uma vez um dicionário com as respectivas chaves e valores para especificar o documento que deseja excluir. Na Figura 18 está exemplificada a query que realiza a exclusão.

```
db.leituras.remove({"leituras.sensor.cod_sensor": "1"})
```

Figura 18 - Query de Exclusão realizada no MongoDB

Fonte: O Autor.

4.4 Resultado do Benchmarking

O Quadro 1 apresenta os resultados dos testes de inserção efetuados nos bancos MySQL e MongoDB, em que estão descritos os intervalos de tempo obtidos para se atingir as quantidades pré-definidas, no qual esse tempo está representado em segundos. A tarefa que mais depende esforço, é a operação de inserção. Isto acontece por ser uma ação bloqueadora no MySQL, pelo fato de possuir relacionamentos entre tabelas, diferente das buscas. Demonstrando nos resultados obtidos o motivo do banco de dados MongoDB ser bem aceito na comunidade².

² Quinto lugar no ranking do site DB-Engines (2014).

Quadro 1. Comparação benchmarking em relação à inserção em ambos os bancos.

	Inserção		
	100	1000	10000
MySQL	0,0781	0,8091	8,3877
MongoDB	0,0298	0,3763	4,2635
Resultado	Analisando os dois bancos foi observado que o MongoDB se apresentou com melhor performance em todos os testes. Nota-se a discrepância de valores, pois no teste de 100 registros/documentos o tempo gasto no MySQL é cerca de aproximadamente três vezes maior e o crescimento obtido em percentual é de 162% em relação ao o MongoDB. No segundo teste a diferença de tempo foi de duas vezes maior e em percentual 115%. Já no terceiro observa-se a diferença de tempo de aproximadamente três vezes e em percentual 96,7%, demonstrando o MongoDB teve um melhor desempenho.		

Fonte: O Autor.

No Quadro 1, 2 e 3 estão representados os resultados dos testes de consulta simples, consulta avançada e exclusão efetuados em ambos os bancos, e também estão descritos os intervalos de tempo de execução obtidos em segundos.

Quadro 2. Comparação benchmarking em relação à consulta simples.

	MySQL	MongoDB
Consulta Simples	0,0033	0,0009
Resultado	Nessa análise pode-se perceber que o MongoDB apresentou um melhor tempo de execução. Nota-se a diferença de valores, pois no teste o tempo gasto no MySQL é cerca de aproximadamente quatro vezes maior e o valor em percentual é de 266,6% em relação ao o MongoDB.	

Fonte: O Autor.

Quadro 3. Comparação benchmarking em relação à consulta avançada.

	MySQL	MongoDB
Consulta Avançada	1,2689	0,2151
Resultado	Na operação de consulta avançada e por haver o uso de queries mais elaboradas, o MongoDB demonstrou melhor desempenho, percebendo um melhor tempo de execução. No teste o tempo gasto no MySQL é cerca de aproximadamente seis vezes maior e o valor em percentual é de 489,9% em relação ao o MongoDB.	

Fonte: O Autor.

Quadro 4. Comparação benchmarking em relação à exclusão.

	MySQL	MongoDB
Exclusão	0,0015	0,0008
Resultado	Na exclusão a diferença de valores foi um pouco menor, no entanto o MongoDB teve o melhor tempo de execução. O tempo obtido no MySQL é cerca de aproximadamente duas vezes maior e o valor em percentual é de 87,5% em relação ao o MongoDB.	

Fonte: O Autor.

Para melhor visualização da comparação dos resultados, gráficos foram criados e para os resultados dos testes foram separados por operação. Na Figura 19, os resultados da operação de inserção são comparados. Considerando que os resultados de inserção para o banco MongoDB foram melhores em relação a mesma operação no banco MySQL.

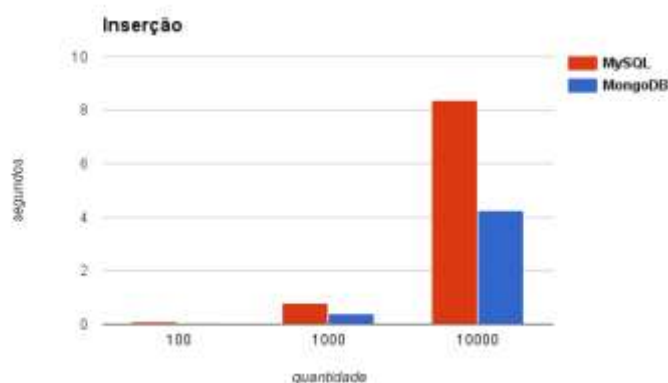


Figura 19 - Comparação da Inserção: MySQL e MongoDB

Fonte: O Autor.

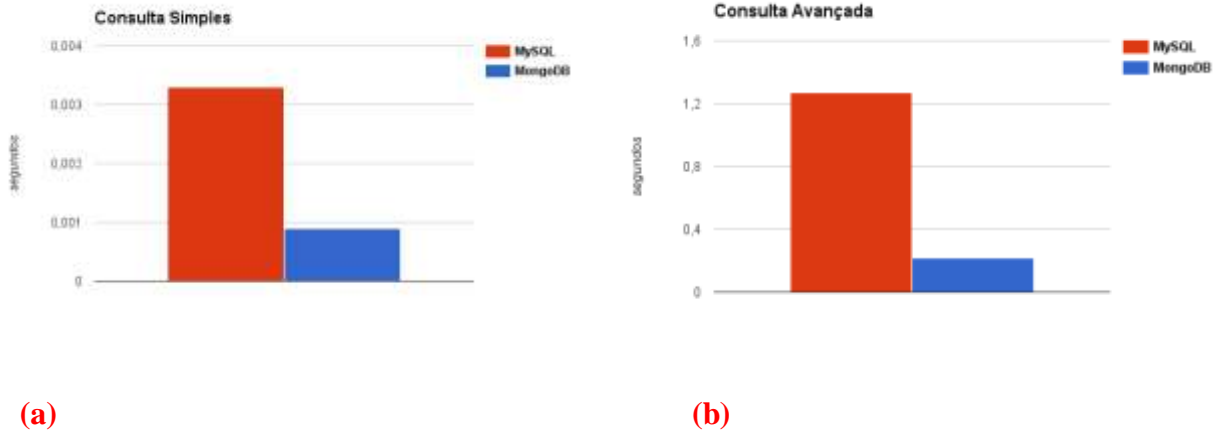


Figura 20 - Comparação da Consulta Simples (a) e Consulta Avançada (b)
Fonte: O Autor.

Na Figura 20 (consulta simples e avançada) o banco MongoDB mais uma vez mostrou melhores resultados na velocidade das consultas de recuperação de documentos. Já o MySQL, apesar de ser possível otimizar as consultas utilizando um código SQL diferente, demonstrou resultados piores.

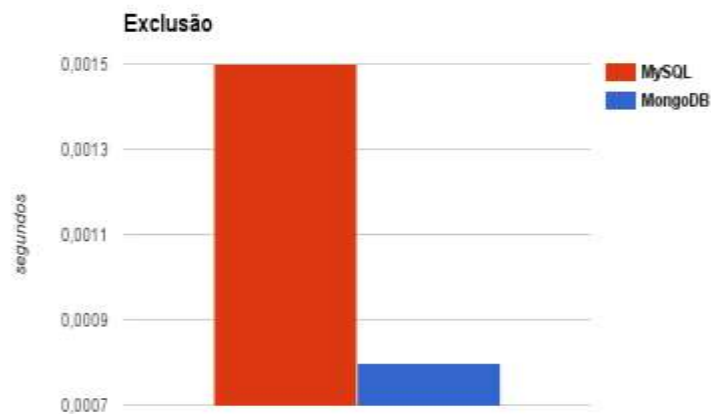


Figura 21 - Comparação da Exclusão
Fonte: O Autor.

Na Figura 21 o banco MongoDB obteve o melhor resultado na velocidade de exclusão de documentos, embora que com escala menor o MySQL teve o pior resultado.

4.5 Considerações Finais

Este capítulo apresentou os resultados obtidos para o MySQL e o MongoDB, mostrando o tempo de resposta e em quanto tempo as tarefas foram finalizadas. Foram testadas quatro situações, sendo elas de inserção, consulta simples, consulta avançada e exclusão, onde foi possível notar que, em todas, o MongoDB teve um desempenho melhor do que o MySQL. Visto também que a base de dados utilizada foi disponibilizada através uma rede de sensores e logo após como os dados coletados foram tratados.

5 CONCLUSÕES

O objetivo do trabalho foi apresentar um estudo comparativo de desempenho entre os Banco de Dados MySQL e MongoDB, na plataforma GNU/Linux, utilizando benchmarking competitivo (ou desempenho). O banco MongoDB apresentou melhores resultados na maioria dos testes, obtendo tempo de execução máximo em 489,9% mais rápido em relação ao MySQL e o mínimo de 87,5%. Os resultados apresentados indicam que o MongoDB está mais apto ao ambiente de IoT, pois demonstra características de boa performance em oportunidades de melhoria no ajuste de desempenho e configuração dos Banco de Dados para tal ambiente, visto que há necessidade de rapidez de inserção de dados, seleção e exclusão dos mesmos.

O resultado apresentado se deu também pelo fato de que os bancos NoSQL terem nascido para suprir a demanda por performance, deixando outros detalhes, como atomicidade, por exemplo, em segundo plano.

Vale ressaltar que nenhum tipo de otimização foi feita para a realização dos testes, o que pode ter influenciado no desempenho dos bancos de dados testados.

Bancos de dados Relacionais e Não Relacionais utilizam paradigmas diferentes e, por sua vez, possuem finalidades diferentes, mas com o mesmo propósito: persistir dados. Segundo os testes de performance, para uma aplicação com alta carga de consultas à base de dados, como o de Internet das Coisas, por exemplo, o banco MongoDB é uma ótima alternativa. Embora que os resultados finais não sejam levados em consideração como um padrão para qualquer outro tipo de ambiente ou aplicação, pois considera-se suas especificações.

Além de salientar a originalidade do estudo ora apresentado, pois foi utilizado um ambiente real de testes, deixa-se recomendações a futuros trabalhos.

5.1 Trabalhos Futuros

O uso do MongoDB foi bastante motivador. Portanto, como trabalhos futuros pretende-se replicar os testes de desempenho descritos neste documento, com adição de testes de manutenção e escalabilidade, envolvendo outros tipos de BD não relacionais, como o orientado a grafo, família de colunas e chave/valor.

REFERÊNCIAS BIBLIOGRÁFICAS

ATZORI, L., IERA, A., MORABITO, G. **The Internet of Things: A survey**. Computer Networks vol. 54 (2010).

ATZORI, Luigi. et al. **The Social internet of things (SIoT) when social networks meet the internet of things: concept, architecture and network characterization**. The International Journal of Computer and Telecommunications Networking, Amsterdam. v. 54, Issue 15, 14 nov. 2012. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1389128612002654>>. Acesso em 26 set. 2015.

BALTZAN, Paige; PHILLIPS, Amy. **Sistemas de Informação**. Porto Alegre: Bookman, 2012.

BANDYOPADHYAY, Soma. et al. Role of middleware for internet of things: a study. International Journal of Computer Science and Engineering Survey, Bhopal. v. 2, n.3, Ago 2011. Disponível em: <<http://airccse.org/journal/ijcses/papers/0811cses07.pdf>>. Acesso em 05 out. 2015 (Publicado por AIRCC).

BELLIO, Matheus Nascimento. **Mongo DB: Um Estudo Teórico-Prático do Conceito de Banco de Dados NoSQL**. Disponível em: <http://www.researchgate.net/publication/272887058_MongoDB_Um_Estudo_Terico-Prtico_do_Conceito_de_Banco_de_Dados_NoSQL_-_Trabalho_de_Diplomao> Acesso em 16 set. 2015.

BOGAN, Christopher E.; ENGLISH, Michael J. **Benchmarking, aplicações práticas e melhoria contínua**. São Paulo: Makron Books, 1996

CERP; **Cluster of European Research Projects on the Internet of Things** (2009) Disponível em: <http://www.internet-of-things-research.eu/> Acesso em 2 out. 2015

CISCO; **Internet of Things**. Disponível em: < <http://www.cisco.com/web/BR/index.html> >
Acesso em 16 set. 2015.

CISCO; **Cisco Visual Networking Index: Forecast and Methodology, 2014-2019 White Paper**. Disponível em: < http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white_paper_c11-481360.html > Acesso em 20 set. 2015

CISCO; **A Internet das Coisas Como a próxima evolução da Internet está mudando tudo**. (2011) Disponível em: < http://www.cisco.com/web/BR/assets/executives/pdf/internet_of_things_iot_ibsg_0411final.pdf>
Acesso em 1 out. 2015

CNC; **Orientador para a prática de Benchmarking**. Disponível em: < <http://www.cnc.org.br/central-do-conhecimento/apostilas/orientador-para-benchmarking> >
Acesso em 3 set. 2015.

CPQD; **Webinar IoT: Do M2M a uma rede global de objetos inteligentes**. Disponível em:
< http://materiais.cpqd.com.br/webinar_iot_m2m > Acesso em 20 set. 2015.

CUNHA, T. M. de A. **Escalabilidade de Sistemas com Banco de Dados NoSQL: Um Estudo de Caso Comparativo com MongoDB e MySQL**. (2011) - Trabalho de Conclusão de Curso (Ciência da Computação) – Centro Universitário da Bahia – Estácio, Salvador.

DATE, C. J. (2003). **Introdução A Sistemas de Banco de Dados**. 8ª Ed. Rio de Janeiro: Elsevier.

DB-Engines (2014). Db-engines web site. Disponível em: <<http://db-engines.com/en/ranking>> Acesso em 14 nov. 2015.

ELMASRI, Ramez; NAVATHE, Shamkant B. (2011) **Sistemas de banco de dados**. 6ª Ed. Traduzido. São Paulo: Pearson Addison Wesley, 2011

FIELDING, R. **Architectural Styles and the Design of Network-based Software Architectures**. 100p. Tese (Doutorado) - University of California, 2000.

FOWLER, Martin. **Padrões de Arquitetura de Aplicações Corporativas**. Porto Alegre: Bookman, 2003.

FUNDAÇÃO NACIONAL DE QUALIDADE; **Informação e Conhecimento**. Pag. 6. São Paulo: FNQ, 2008.

GEREMIA, Juliana. (2010) **Tutorial de Introdução a Banco de Dados**. Disponível em: <http://www.telecom.uff.br/pet/petws/downloads/tutoriais/db/Tut_DB.pdf> Acesso em 10 out. 2015.

GONÇALVES, ROGÉRIO B. (2004) **Aspectos Básicos de Banco de Dados**. Disponível em: <<https://www.marilia.unesp.br/Home/Instituicao/Docentes/EdbertoFerneda/BD%20-%20Aspectos%20Basicos.pdf>> Acesso em 13 set. 2015

GREN, Martin A **Internet das Coisas e o futuro da tecnologia em segurança IP**. Disponível em:<<http://computerworld.com.br/internet-das-coisas-e-o-futuro-da-tecnologia-em-seguranca-ip>> Acesso em 20 set. 2015

HOWS, D.; MEMBREY, P.; PLUGGE, E. **Introdução ao MongoDB**. 1. Ed. São Paulo: Novatec, 2015.

HUI, J. W., CULLER, D. E. **Extending IP to Low-Power, Wireless Personal Area Networks**. Internet Computing, IEEE 12, no. 4 (2008).

JSON; **Introdução ao JSON**. Disponível em: <<http://www.json.org/json-pt.html>> Acesso em 20 nov. 2015.

KOTARO, O. T.; CRISTINA, I. I.; EDUARDO, J. F. (2005) **Introdução a Banco de Dados**. Disponível em: <<https://www.ime.usp.br/~jef/apostila.pdf>> Acesso em 13 set. 2015

LÓSCIO, B. F., OLIVEIRA, H. R. e PONTES, J. C. S. (2011) “**NoSQL no desenvolvimento de aplicações Web colaborativas**”, VIII Simpósio Brasileiro de Sistemas Colaborativos.

MELO, Sandro (2014). **Como a internet das coisas influencia a gestão de TI**. Disponível em: < <http://tecnologia.bandtec.com.br/como-a-internet-das-coisas-influencia-a-gestao-de-ti> > Acesso em 21 set. 2015

MongoDB. Disponível em:< <http://www.mongodb.org>> Acesso em 18 out. de 2015

MySQL. Anúncio publicitário. Revista Linux Magazine. Ano II, no6, jan. 2005, p. 2

NOGUEIRA, M.; SANTOS, A.; TORRES, J.; ZANELLA, A.; DANIELEWICZ, Y. **Gerência de Identidade na Internet do Futuro**. Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos-SBRC (2011) Disponível em < <http://sbrc2011.facom.ufms.br/files/mc/mc4.pdf> > Acesso em 20 set. 2015

NETO, F. G. P. (2008). **Banco de Dados Relacional - Exemplo** [on-line]. Disponível em: <<http://e-reality-database.blogspot.com.br/2008/09/banco-de-dados-relacional-exemplo.html>>. Acesso em 19 set. 2015

Oracle. Disponível em: <<http://www.oracle.com/br/products/mysql/overview/index.html>>

PRESSMAN, Roger S. **Engenharia de Software**. São Paulo: Makron Books, 6. Ed. 2006: McGraw-Hill 2006.

SADALAGE, PRAMOD J.; FOWLER, M. **NoSQL – Um Guia Conciso para o Mundo Emergente da Persistência Poliglota**. 1. Ed. São Paulo: Novatec, 2013.

SCHROEDER, Ricardo; SANTOS, Fernando dos; **Arquitetura e Testes de Serviços Web de Alto Desempenho com Node.js e MongoDB**. (2010). Disponível em: <http://www.ceavi.udesc.br/arquivos/id_submenu/787/ricardo_schroeder_versao_final_.pdf> Acesso em 3 set. 2015

SOUSA, F. R. C., MOREIRA, L. O., MACÊDO, J. A., MACHADO, J. C. **Gerenciamento de Dados em Nuvem: Conceitos, Sistemas e Desafios**. SBBD 2010. p. 101–130.

Disponível em: <<http://www.ufpi.br/subsiteFiles/ercemapi/arquivos/files/minicurso/mc7.pdf>>

Acesso em 9 set. 2015

TRIBUNAL DE CONTAS DA UNIÃO: SECRETARIA-GERAL DE CONTROLE EXTERNO


COORDENADORIA DE FISCALIZAÇÃO E CONTROLE, 2000: **Técnica de Auditoria: Benchmarking.** Disponível em:

<http://portal3.tcu.gov.br/portal/page/portal/TCU/comunidades/programas_governo/tecnicas_anop/BENCHMARKING.pdf> Acesso em 22 set. 2015

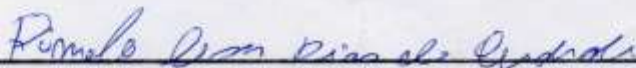
VIEIRA, M. R., FIGUEIREDO, J. M., LIBERATTI, G., VIEBRANTZ, A. F. M. **Bancos de Dados NoSQL: Conceitos, Ferramentas, Linguagens e Estudos de Casos no Contexto de Big Data.** Minicurso SBBD 2012, São Paulo, 2012. Disponível em: <

http://data.ime.usp.br/sbbd2012/artigos/sbbd_min_01.html. > Acesso em 3 set. 2015.

Monografia de Graduação apresentada por **Ana Katarina Cavalcanti Medeiros** do Curso de Graduação em Sistemas de Informação da Faculdade de Ciência e Tecnologia de Caruaru – Universidade de Pernambuco, sob o título “ANÁLISE COMPARATIVA ENTRE BANCOS DE DADOS RELACIONAIS E NÃO RELACIONAIS COM ÊNFASE NO DESEMPENHO DE BASE DE DADOS NO CONTEXTO DE INTERNET DAS COISAS”, orientada pelo Prof. Msc. **Rômulo César Dias de Andrade** e aprovada pela Banca Examinadora formada pelos professores:



Prof. Msc. Elyda Laisa Soares Xavier Freitas
Departamento de Sistemas de Informação / UPE



Prof. Msc. Rômulo César Dias de Andrade
Departamento de Sistemas de Informação / UPE

Visto e permitida a impressão.
Caruaru, 16 de dezembro de 2015.

Prof. Dra. Patricia Takako Endo
Coordenadora do Curso de Bacharelado em Sistemas de Informação da
Faculdade de Ciência e Tecnologia de Caruaru – Universidade de Pernambuco.